System Level Modelling and Design of Hypergraph Based Wireless System Area Networks for Multi-Computer Systems

Nuredin Ali Salem Ahmed

A thesis submitted in fulfilment for
the degree of Doctor of Philosophy

to the College of Science and Engineering, School of Engineering
Department of Electronics and Electrical Engineering

University *of* Glasgow

Declaration

The work described in this Thesis was carried at the University of Glasgow under the supervision of Dr Khaled Elgaid and Dr Fernando Rodríguez-Salazar, Department of Electronics and Electrical Engineering, in the period March 2007 to January 2010.

The author hereby declares that the work described in this Thesis is her own, except where specific references are made. It has not been submitted in part or in whole to any other university for a degree.

Nuredin Ahmed

Glasgow, January 2011

Abstract

This thesis deals with issues pertaining the wireless multicomputer interconnection networks namely topology and Medium Access Control (MAC). It argues that new channel assignment technique based on regular low-dimensional hypergraph networks, the dual radio wireless hypermesh, represents a promising alternative high-performance wireless interconnection network for the future multicomputers to shared communication medium networks and/or ordinary wireless mesh networks, which have been widely used in current wireless networks.

The focus of this work is on improving the network throughput while maintaining a relatively low latency of a wireless network system. By means of a Carrier Sense Multiple Access (CSMA) based design of the MAC protocol and based on the desirable features of hypermesh network topology a relatively high performance network has been introduced. Compared to the CSMA shared communication channel model, which is currently the de facto MAC protocol for most of wireless networks, our design is shown to achieve a significant increase in network throughput with less average network latency for large number of communication nodes.

SystemC model of the proposed wireless hypermesh, validated through mathematical models, are then introduced. The analysis has been incorporated in the proper SystemC design methodology which facilitates the integration of communication modelling into the design modelling at the early stages of the system development. Another important application of SystemC modelling techniques is to perform mean-

ingful comparative studies of different protocols, or new implementations to determine which communication scenario performs better and the ability to modify models to test system sensitivity and tune performance. Effects of different design parameters (e.g., packet sizes, number of nodes) has been carried out throughout this work.

The results shows that the proposed structure has out perform the existing shared medium network structure and it can support relatively high number of wireless connected computers than conventional networks.

Acknowledgements

I would like to thank my supervisors, Dr. Khaled Elgaid and Dr Fernando Rodríguez-Salazar, for their precious guidance and continuous support throughout this work. Thanks goes to all of the support received from my family; sometimes emotional other times intellectual. In particular a big thanks go to my beloved children for being here and providing joy to my life. Finally, to my wife who helped me to organise my ideas and life. It is to her that I dedicate this work.

Contents

List of Publications

Aspects of the work described in this Thesis has been published in following papers:

1. I. A. Aref, **N. A. Ahmed**, F. Rodriguez-Salazar, and K. Elgaid, RTL-level modelling of an 8b/10b encoder-decoder using systemc, *5th IFIP International Conference on Wireless and Optical Communications Networks*, WOCN '08, pp. 1–4, May 2008.

2. **Ahmed, N.** A.; Aref, I. A.; Rodriguez-Salazar, F. & Elgaid, K., Wireless Channel Model Based on SoC Design Methodology, *4th International Conference on Systems and Networks Communications* (ICSNC 09), IEEE Computer Society, 2009, 72-75

3. Aref, I.; **Ahmed, N.**; Rodriguez-Salazar, F. & Elgaid, K. Wireless extension into existing SystemC design methodology, *2nd International Conference on Computer Engineering and Technology (ICCET)*, Computer Engineering and Technology (ICCET), 2010, 3, V3-374 -V3-379

4. Aref, I.; **Ahmed, N.**; Rodriguez-Salazar, F. & Elgaid, K. Measuring and Optimising Convergence and Stability in Terms of System Construction in SystemC, *17th IEEE International Conference and Workshops on the Engineering of Computer-Based Systems* (ECBS), IEEE Computer Society, 2010, 0, 263-267

5. Aref, I.; **Ahmed, N.**; Rodriguez-Salazar, F. & Elgaid, K. Modelling of Flocking Behaviour System in SystemC, *6th Advanced International Conference on on Telecommunications* (AICT), 2010, 358 -363

6. **Nuredin Ahmed**; Ibrahim Aref; Fernando Rodriguez-Salazar & Khaled Elgaid, Network Performance Evaluation Based on SoC design Methodology, *7th IEEE, IET International Symposium on Communication Systems, Networks and Digital Signal Processing* (CSNDSP), 2010, 256 - 261

7. **Nuredin Ahmed**; Ibrahim Aref; Fernando Rodriguez-Salazar & Khaled Elgaid, Network performance Evaluation using Realistic Design Process, *5th Libyan Arab International Conference on Electrical and Electronic Engineering* (LAICEEE), 2010, 1, 63-75

8. Aref, I.; **Ahmed, N.**; Rodriguez-Salazar, F. & Elgaid, K. A SystemC-Based Design Methodology for Modelling Complex Wireless Communication Systems, *5th Libyan Arab International Conference on Electrical and Electronic Engineering* (LAICEEE), 2010, 1, 51-61

9. **Nuredin Ahmed**; Ibrahim Aref; Fernando Rodriguez-Salazar & Khaled Elgaid, Single and Multi-Channel Networks: Performance Comparison at System Level, *submitted to IET Communications Journal.*

List of Figures

List of Abbreviations

8B/10B	8Bit/10Bit encoder/decoder
ABM	Asynchronous Balanced Mode
ACK	Acknowledgement
ASK	Amplitude Shift Keying
BEP	Bit Error Probability
CDMA	Code Division Multiple Access
CRC	Cyclic Redundancy Check
CSMA	Carrier Sense Multiple Access
dB	decibel
DLC	Data Link Control
DRWHN	Dual-radio Wireless Hypermesh Network
DUT	Device Under Test
EIRP	Effective Isotropic Radiated Power
ESL	Electronic System Level Design
FDMA	Frequency Division Multiple Access
FIFO	First In First Out
FPGA	Field-Programmable Gate Array
FSM	Finite State Machine
GBN	Go Back N
HDL	Hardware Description Languages
HW/SW	Hardware/Software
I/O	Input/Output
IFS	Inter Frame Spacing
IP	Intellectual Property

ISO	International Organisation for Standardisation
LAN	Local Area Network
MAC	Medium Access Control
MP2MP	Multipoint to Multipoint
OFDM	Orthogonal Frequency Division Multiplexing
OOK	On-Off Keying
OSI	Open Systems Interconnection
P2M	point to multipoint
P2P	Point to Point
PDF	Probability Density Function
PHY	Physical Layer
PLL	Phase-Locked Loop
PSK	Phase Shift Keying
QAM	Quadrature Amplitude Modulation
RF	Radio Frequency
RTL	Register Transfer Level
RTS/CTS	Request to Send/Clear to Send
SANs	System Area Networks
SEP	Symbol Error Probability
SINR	Signal-to-Noise-and-Interference Ratio
SNR	Signal to Noise Ratio
SoC	System on Chip
TCP	Transmission Control Protocol
TDMA	Time Division Multiple Access
WLAN	Wireless Local Area Network

1 Introduction

1.1 Thesis Motivation

The design of modern computing systems is evolving into a more and more complex task. Moreover the increased demand for high performance computing from the military and research centres in industry and academia for the simulation of complex problems makes System Area Networks (SANs) a very interesting research area. System area networks are designed to interconnect high performance computing resources that are located over a short distance, typically in a building with a range of a few meters. Each computer is formed by a processing element, where information is processed, and a switching or communication element, where packets of data are sent or received from other computers.

Scientific computing research applications such as fluid dynamics or finite element methods, modelling of nuclear explosions, climate modelling, and others, need a very high level of integration between computation and communication, because they have very rich communication patterns and they produce a huge amount of data, that need to be exchanged between the communication nodes. Others like Google for instance have a very large bandwidth and need to have very high speed access to its databases. So the solution for all these needs is to communicate faster with low latency, and that is the fundamental problem, high bandwidth with low latency. It is impossible or extremely difficult and costly to achieve.

The central problem in achieving faster and cheaper communications in large multicomputer networks has proven to be the way in which the processors are connected together (their topology) [1, 2, 3]. There are several network architectures and topologies for implementing a networked computer system. Some of the most important networks are: Fully connected or all-to-all, a Circular Ring, a Star, a Binary tree, Mesh (Torus), Hypermeshes, or even Random networks. Most of the above mentioned interconnections are designed to implement wired networks, which can have more applications in a wireless sensors network.

Current SANs interconnection requires wire connections by means of fibre optics like Myrinet, Gigabit Ethernet, Quadrics, Infiniband and ServerNet. The cost of the hardware rises as the number of computers/workstations increases. As the number of communication nodes increase, so does the switching delay, becoming the bottleneck of the interconnection. This degrades the overall performance.

Connecting a cluster of computers wirelessly by replacing the standard fixed cabling, which creates more compact configurations is an important issue. Cabling technology has reached fundamental limitations in the current technology. The maximum wiring density and the maximum number of pins per chip or per board limits the network scalability [2]. For instance Point to Point (P2P) requires switching elements and that will introduce scaling problems of the networks. The cables are expensive because they need to be controlled and constructed very well.

However, a communication node in a wireless environment has the ability to communicate with a large number of other nodes. So, the physical constraints associated with wired connectivity or expensive attenuation equipment (devices that maintain signal strength during transmission) becomes less of an issue in a wireless environment.

Disadvantages of wireless network systems are that they are slower than the existing wireless technology, which does not offer the performance of wired network systems.

However, the rapid development of wireless technology and the explosive growth in wireless communication, where fixed cabling is undesirable, has found many applications. An alternative solution is proposed in this work. Namely, the use of wireless technology in the realisation of high performance SANs. To achieve high data rate wirelessly, next generation wireless networks will employ various physical layer techniques, e.g., multiple input multiple output (MIMO) beam forming, directional antenna, etc., to improve the link capacity and reliability. In addition, advanced modulation techniques such as Orthogonal Frequency Division Multiplexing (OFDM) and/or On-Off Keying (OOK) has emerged as a good choice for wide bandwidth to share the electromagnetic spectrum with already deployed systems. These solutions are the most investigated for wide bandwidth communication systems to make maximum use of available bandwidth. It seems reasonable that an attempt to utilise this technology in parallel computing is advised.

Naively some researchers like [4] advocated the use of wireless channels to substitute wire channels in graph based interconnections networks. Published work [4] proposed the use of wireless technology in the realisation of inter-processor communication in parallel processing. The network topology, proposed in [4] is the fully connected network and they argued that this topology may be the best possible solution for parallel programming tasks, because it has a simple scheduling technique. Hypermesh is the most interesting topology in this work. Hypermesh has been implemented in different ways such as in references [5, 6, 7, 8, 9]. However, utilizing the hypermeshes to construct wireless interconnected multiprocessors is still of interest for future work. According to [5, 6, 7, 8, 9] hypermeshes have desirable features over other interconnection networks such as a low diameter, high bandwidth, low latency network. Those advantages make the hypermeshes embed naturally in a wide range of communication patterns [9]. A number of different hypermesh implementations have been suggested in the literature including shared buses, crossbar switches, distributed crossbar switch with different implementation technologies, costs and constraints [10, 5, 6, 8, 9].

In wireless communication, a channel is shared by all nodes in that when a sender transmits a packet, all nodes within the sender's transmission range can receive this transmission. The particular advantage is that in wireless you don't need to go for P2P you can have broadcast communication. Point to multipoint (P2M) broadcast or Multipoint to Multipoint (MP2MP) in wireless is very efficient. These wireless features imply bus structures, which imply hypergraphs or hyperchannels. The fundamental issue here relate to physical limitations of the wireless channel such as noise, fading and multipath interference. One way to circumvent such limitations is to use more than one independent wireless channel. Instead of using only one channel, in this work, I consider a two dimensional dual-radio wireless hypermesh network, where each router node is equipped with two radio interfaces and two non-overlapping channels are available for communication for each node. I address the problem of assigning channels to communication links in the network with the objective of keeping overall network latency low and provide a relatively high throughput. This approach differs from common ad-hoc networking proposals, which use only a single channel. Given that a radio channel is a shared resource, the use of more channels would reduce contention if I distribute nodes on different channels. To deploy this method, the system owner will intelligently place the nodes on a way to meet the hypermesh topology to provide adequate coverage and sufficient capacity. I believe that my proposal of using multiple radio interfaces on a node can address important weaknesses of ad-hoc networks. At present, very little is known about performance improvements that can be achieved using a multi-interface multi-channel network system. It is thus necessary to evaluate the behaviour of such systems through analysis and simulation before an eventual implementation.

With the advent of digital communication systems, more and more components to support new functionalities are being integrated in a single package. This trend is likely to continue, as devices continue to incorporate an ever-growing number of components to provide inter-operability with the large plethora of standards and protocols from previous and the present state of the art systems. The traditional ways of network designing mainly depend on experience. However, the ways that

simply depend on experience to design networks can not keep up with the development step of new systems. Moreover traditional design methodologies increasingly fail to handle such reasoning in a cost- and time-effective manner, when product time-to-market is the key to success [11, 12, 13].

Electronic System Level Design (ESL) in particular SystemC design methodology lines up to tackle this issue. In the past decade it has been observed a transition from gate-level design to Register Transfer Level (RTL)-level design. SystemC is a candidate for the language that will be used at all levels of system and chip design. Starting to use SystemC in current RTL/Behavioural design can accelerate the transition. Despite being a relatively new emerging design methodology, SystemC has not, to the best of my knowledge, been extended to incorporate, within the same framework, the design of a wireless communication systems (as found in digital radio communications, for instance). SystemC design and verification methodology simulate the interaction and communication of different system parts at a different levels of abstraction. This leads to more effective functional verification of all the components working together in early development phases.

The development of most wireless communication systems implies the use of block coding. However as it is not an absolute requirement of the construction of wireless hypermesh, the modelling of an RTL-level model of an 8Bit/10Bit encoder/decoder (8B/10B) block in SystemC has been developed in this work. The use of 8B/10B coding is an important technique in the construction of high performance serial interfaces. These are particularly suitable for alleviating the Input/Output (I/O) bottleneck of state of the art systems (which are pinout, rather than bandwidth limited).

1.2 Aims and Objectives

The aim of this PhD is to model a Dual-radio Wireless Hypermesh Network (DRWHN) Based-on Carrier Sense Multiple Access protocol for a multi-computer system. Thus, the development of DRWHN that deploys all the communication system components in different levels of abstraction is defined as the main task. The design will consider providing low-latency and high bandwidth, based on a classical hypermesh topology for expandability which intern advocates scalability. In order to accomplish this aim, several key research objectives have been identified:

1. Designing, implementing and exploring different sub-system blocks for the construction of the whole communication system.

2. *Digital wireless communication channels:* which represent the communication link or medium between communicating nodes. This can be sub-classified into the following scenarios:

    a) *Point-to-point* communication channel to model the first prototype communication system and design a noisy digital wireless channel.

    b) *Multipoint communication channel:* to address the broadcast and/or multicast scenarios:

        i. Utilising the communication channel for broadcasting scenario in a multichannel model with the assumption of no correlation between channels.

        ii. Shared communication channel model between nodes, where multiple nodes are connected through it with assurance that the behaviour of the channel will be modelled correctly as a wireless communication

medium for multi-point communication scenario.

iii. Refining the channel module to support contention and noncontention based wireless communication. And also, it will support different noise and corrupting methods.

3. *Communication nodes:* That might represent the computers or any communication device, which includes Network interface, switching methods, routing algorithms and data flow control, etc.. I first introduced the layering principle that is commonly used in the design of communication nodes. This challenging task is usually accomplished with a layered architecture such as the open systems interconnection (OSI) model proposed by the international standards organization (ISO).

4. After examining a series of components as separate entities, I focused on the integration of these components to form a DRWHN to construct the whole targeted system.

5. Development of a reusable system intellectual property (IP) cores, which are reusable hardware blocks offering flexible interoperability, which can be used to allow efficient prototyping of communication systems.

6. Performance analysis issues related to the communication refinement and performance evaluation steps and their impact to the overall system performance, which will lead to improved design and will provide a foundation to improve minimisation of communication latency.

1.3 Approach

This thesis evaluates the potential benefits of using hypermesh network topology to design a multi-interface multi-channel wireless SAN. I first carried out the development of a wireless communication link in order to add a missing element of modelling off-chip communications such as wireless links to SystemC. Since, SystemC design methodology has been developed for design and implementation of complex systems, in particular for System on Chip (SoC), it has not, to the best of my knowledge, been extended to incorporate, within the same framework, the design of wireless communication system. Wireless communication network system design begins with detailing the channel model, then developing the transmitter and receiver that best compensate for the channel's corrupting behaviour. Then I proceeded to the modelling of a single-channel and multiple-channel network in order to confirm the application of using SystemC design methodology in developing wireless network systems. Based on this work, I designed a mechanism to coordinate multiple radio-interfaces on one node based on the hypermesh channel assignment mechanism described in chapter 8. I then evaluated the network performance for the different network topologies.

1.4 Thesis outline

The modelling of a Dual-Radio Wireless Hypermesh Network Based-on CSMA protocol is organised into several chapters, formulated in a general structure of introduction main body and conclusion.

In Chapter 2 a review wireless data networks with an emphasis on the network topologies, routing and switching techniques, and multi-channel transmission protocols. I also include a brief description of the exiting ad-hoc wireless networks. It

discusses the limitations and the previously proposed implementation schemes for ad-hoc/mesh wireless networks.

Chapter 3 will review the existing system level network modelling techniques in particular SystemC design methodology. Furthermore, it will highlight and describes the advantages of using SystemC methodology and compare it to other existing techniques. For instance it will discuss the use of Co-simulation techniques using Matlab, Simulink and SPICE for analogue parts of the system and other Hardware Description Languages (HDL) for digital parts. And provide a clear insight of using this relatively new design methodology. In addition, it will introduce briefly the simulation measurement technique used throughout the development of the simulation work. Moreover, it will highlight the importance of this part in the development process, which is all too often overlooked or skipped.

Then I proceed to the modelling of a noisy digital communication channel, because I have found in my review of SystemC methodology that it is missing the elements of modelling off-chip communications such as wireless links. Chapter 4 dealt with the modelling of wireless digital communication channel at the system level. And it presents the integration of basic wireless characteristics of the communication medium which is a packet corrupting channel at the bit or digital level of communication. Then using it in SystemC to design the digital wireless network system.

The modelling of a digital communication channel is representing just one component of any communication system. other components are essential in implementing digital communication systems such as 8B/10B encoder which can increase the transmission performance. In Chapter 5, the modelling of an RTL level of an 8B/10B encoder in SystemC has been presented. As it has been stressed earlier in this chapter that, the use of 8B/10B coding is an important technique in the construction of high performance serial interfaces. In addition, to optimise the use of the transmission medium encoding may be chosen to conserve bandwidth or to minimise errors. Furthermore, 8b/10b has been widely adopted by a variety of high speed

data communication standards used today such as PCI Express prior to 3.0 [14], InfiniBand [15], HyperTransport [16] and Common Public Radio Interface (CPRI)[17] and should prove ever more useful for FPGA-based designs as clock speeds and I/O capabilities increase. It was a challenging task because, I wanted to produce an RTL level model of the encoder. And because, it allowed me for better examination of the SystemC methodology in modelling at different levels of abstraction (which encompass all levels from system specification to implementation).

Then I moved to Chapter 6, the modelling of the wireless networking system based on CSMA protocol. In this chapter I modelled at the system level all protocols that needed for the communication over wireless channels. The communication node includes most of the network layers of the network are modelled as modules with different methods, which will reflect the Application, Transport and Data Link Control (DLC) and Physical layers (PHY) of the standard reference model of an OSI/ISO. In addition to test my designs of individual components, a shared communication medium network topology has been modelled. Performance results for the simulations as well as development effort are presented thus showing how this methodology is well suited to the modelling of wireless network systems.

In chapter 7, a channel module refinement process has been in detail presented. And also presents the use of SystemC to compare two different networks at the system level. Single cluster of a single channel network based on CSMA and multi-channel network cluster based on non-overlapping channels available for each node, has been developed at the system level. Moreover this chapter has incorporated analytical approximations to validate the results obtained in chapter 6. The results have revealed that the multi-channel network has superior performance characteristics over the shared communication network. Moreover the experiences of using SystemC design methodology to analyse the performance properties of wireless network has been presented.

Chapter 8 presents the implementation proposed for low dimensional DRWHN which

allows relaxation of the wireless bandwidth constraints, and provide an outperform channel assignment configuration to the existing WMN. The focus of this chapter is on improving the network throughput while maintaining a relatively low latency of the network by means of a CSMA-based design of the MAC protocol, and based on the desirable features of hypermesh network topology. Compared to the CSMA shared communication channel model, which is currently the de facto MAC protocol for most of wireless networks, my design is shown to achieve a significant increase in network throughput with less average network latency for a large number of communication nodes. Moreover, the model has been validated by means of analytical approximations.

Finally, Chapter 9 is the last in this thesis and provides conclusions and recommendations for future work.

2 Wireless Network Background

In this chapter, I will introduce the background under which the work of this thesis is done. A review of wireless data networks will be introduced. I will consider the physical arrangement which is used to interconnect nodes, that is known as the network topology and the process of determining a path between any two nodes over which traffic can pass which is called routing. Next is the switching techniques used in this work, which refers to the transfer method of how data is forwarded from the source to the destination in a network. In addition I will address medium access control protocols for wireless network system. And finally channel assignment strategies and wireless channel models will be reviewed.

2.1 Wireless Networks

Wireless networks, also called ad-hoc networks, formed by collections of wireless nodes communicating with one another with no pre-existing infrastructure in place; therefore, they are also called infrastructureless networks [18]. A wireless network is ad-hoc if each node forwards data from other nodes and produces and consumes data of its own. Wireless ad-hoc networks have been the focus of much recent research, and include Mobile Ad-hoc networks (MANETs), Wireless Sensor Networks (WSNs), Wireless Mesh Networks (WMNs), and Vehicular ad-hoc Networks (VANETs).

An infrastructureless network can be either a single hop or a multi-hop network which autonomously operates in an ad-hoc mode without a central controller. The term multi-hop refers to the fact that data from the source needs to travel through several other intermediate nodes before it reaches the destination. Ad-hoc networks based on wireless technologies, such as IEEE 802.11 standard, which covers the physical and data link layers and mostly utilize a single radio and a single shared channel. As such, the bandwidth is divided between the nodes trying to communicate. One common problem with such protocols is that the network performance will degrade quickly as the number of nodes increases, due to higher contention/collision [19]. On the other hand, the wireless standards IEEE 802.11a/b/g and IEEE 802.15.4, offer up to 16 non-overlapping frequency channels for simultaneous communication. These multiple channels have been utilized in infrastructure-based networks by assigning different channels to adjacent access points, thereby minimizing interference between access points. However, multi-hop wireless networks have typically used a single channel to avoid the need for co-ordination between adjacent pair of nodes, which is necessary in a multi-channel network.

Multiple channels, however, partition the network based on the channel used. This may result in a disconnected network if the nodes communicate only in their assigned channels. To resolve this problem, several multi-channel ad-hoc/mesh network approaches have been proposed in the literature [20, 21]. Furthermore, Some research [22, 23] has been done on routing schemes in multichannel networks where the topology discovery and routing are performed with a channel assignment. In addition, they are considered these issues as separate problems thus reducing complexity of the schemes. So et al. [22] have proposed a routing protocol for multi-channel networks that uses a single interface at each node, while our proposed solution works with multiple radio interfaces per node. Raniwala et al. [23] propose routing and interface assignment algorithms for static networks. Similar to our proposal, they also consider the scenario wherein the number of available interfaces is less than the number of available channels. However, their solution is designed specifically for use in those mesh networks where all traffic is directed toward specific gateway nodes.

In contrast, our proposal is designed for more wireless hypermesh networks, where potentially any node may communicate with any other node.

Previous research has highlighted that, using multiple channels in wireless ad-hoc networks, can enhance the network capacity and throughput, as concurrent communications can go on simultaneously over different frequency channels without interfering [24]. Moreover for exploiting the advantages given by multiple channels, advanced MAC protocols are required to properly assign available channels to the radio interfaces mounted on each node [24]. Nasipuri et al. [25] show that, in the extreme case when channel assignment is perfect and each pair of nodes has a dedicated channel, contention and collision disappear. In this situation, network capacity can be fully used. It is worth noticing that the best channel assignment can be provided if each network node has a number of interfaces, namely NICs (Network Interface Cards), equal to the number of channels.

Given that this work is based on the modelling of digital wireless network for SANs, I first briefly describe topologies relevant for wireless networking. I then discuss routing protocols and switching techniques, which are essential for multi-hop, multiple channels wireless networks. Finally channel assignment strategies and wireless channel models will be addressed.

2.2 Topologies Relevant for Wireless Networking

One of the main design choices for any interconnection network is the topology, which affects directly or indirectly other design considerations such as routing, switching and flow control. Topology refers to the configuration of the network nodes and how data is transmitted through that configuration. In addition, it includes characteristics such as the degree and diameter of the network. The degree is the maximum number of neighbours connected to a node. The diameter is the maximum shortest

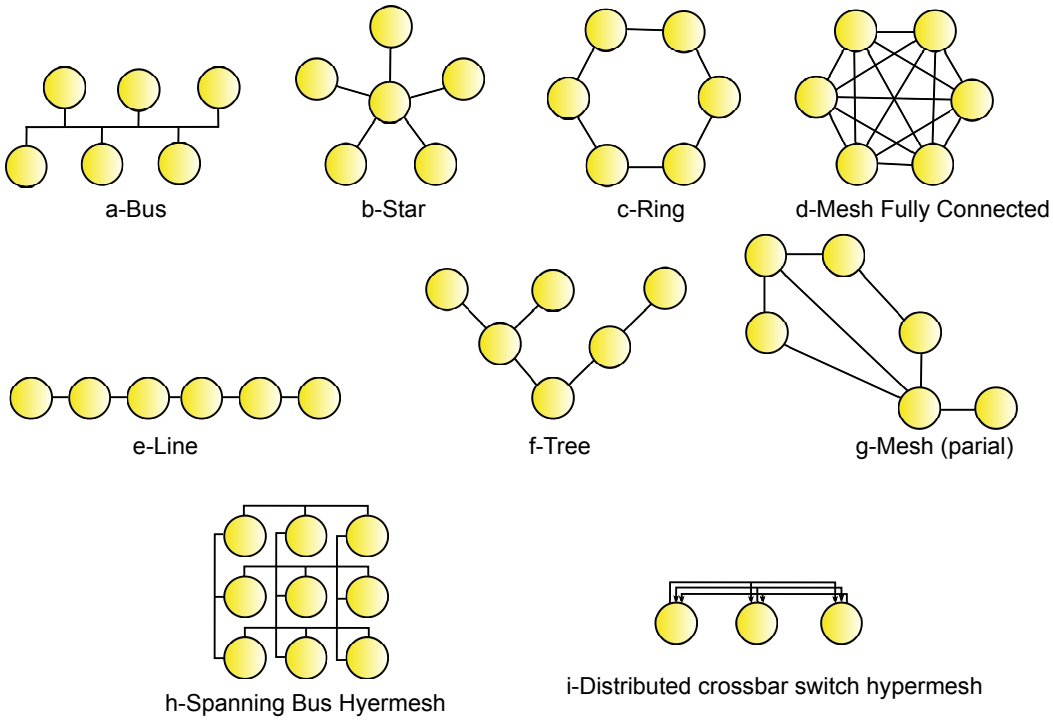

Figure 2.1: Network Topologies Relevant for Wireless Networking

Topology	Relevance
Bus	Yes, shared medium, CSMA based
Star	Yes, standard wireless topology (P2M)
Ring	Possible, but rarely found
Fully Connected Mesh	Yes, but rarely found
Line	Yes, with two or more elements (P2P)
Tree	Yes (a combination of star and line)
Mesh	Yes, mainly partial mesh

Table 2.1: Topologies Relevant for Wireless Networking

distance between any pair of nodes. Researchers have proposed various topologies [10, 26, 2]. Various topologies are shown in Figure (2.1) like Bus, Fully connected or all-to-all, a Circular Ring, a Star, a line, a Binary tree, Mesh (Torus), Hypermeshes, or even Random networks. In this section, I briefly discuss popular topologies that are relevant for wireless networking. Table (2.1) briefly provide a quick overview of that topologies relevance to wireless networks.

1. *Bus Topology:* The bus topology of Figure (2.1-*a*) has been used extensively by LANs. Bus topology is the most common type of interconnection networks

since it can be implemented easily with a cheap hardware cost. A unique characteristic of a shared medium is its ability to support broadcast, in which all nodes on the medium can monitor network activities and receive the information transmitted on the shared medium [26]. Although, this topology allows only one pair of nodes to communicate at any given time instance. This deadly bottleneck makes the bus topology saturate quickly for a large number of nodes.

2. *Star Topology*: star topology is the most common infrastructure in wireless networking. It is a single-hop interconnect in which all nodes are within direct communication range — usually 30 to 100 meters [27] for small networks — to the central communication unit. It is well suited for Point to Multipoint communication. Figure (2.1-*b*) shows a typical star topology network. Star topology has also more application in cellular systems, WLAN, and satellite systems in which one satellite station communicates to multiple ground stations [26, 27]. Disadvantage, if the central unit fails then everything connected to it is down.

3. *Line or Chain Topology*: In Chain, all communication nodes reside on a single path line topology to form a point-to-point network topology. Each network node directly communicates to only one other node. Figure (2.1-*e*) shows a typical topology of a point-to-point network. Wireless point-to-point systems are often used in wireless "backbone" systems such as microwave relay communications. The biggest disadvantage of a point-to-point wireless system is, that it is strictly a one-to-one connection. This means that there is no redundancy in such a network at all. If the RF link between two point-to-point radios is not robust, the communicated data can be lost [26, 28]. In a line network with N nodes, the diameter is (*N-1*), average distance is $\frac{N-1}{2}$, and bisection width is 1.

4. *Ring Topology*: ring topology is also a P2P network topology. In a ring, each

node is connected in the form of a closed loop of the communication medium. Signals travel in one direction from one node to all other nodes around the loop and all nodes are working as repeaters. Figure (2.1-*c*) shows a ring topology. A ring makes a poor interconnection network due to its large diameter and poor fault tolerance since it takes more radio hops to reach distant node [26, 28].

5. *Tree Topology*: The tree topology is essentially a hybrid of the bus and star layouts. This topology has a root node connected to a certain number of descendant nodes. Each of these nodes is in turn connected to a disjoint set of descendants. A node with no descendant is a leaf node. Figure (2.1-f) shows a tree topology. The biggest drawback of the tree topology as a general purpose interconnection network is that the root and the nodes close to it become a bottleneck. Additionally, there are no alternative paths between any pair of nodes.

6. *Fully Connected Mesh Topology*: Such a mesh might seem an obvious first approach to interconnecting nodes. A mesh topology shown in Figure (2.1-*d*) provides each device with a P2P connection to every other device in the network. These are most commonly used in WAN's, which connect networks over telecommunication links. Mesh networks provide redundancy, in the event of a link failure. Meshed networks enable data to be routed through any other site connected to the network. Because each device has a P2P connection to every other device, mesh topologies are the most expensive and difficult to maintain. I will investigate this topology under the assumption that in a wireless network, each node needs one communication channel to communicate with other nodes. Using this assumption the number of switches that need to have the same topology in wired networks will be reduced to N instead of $N(N-1)$ in wired networks to switch from channel to the other. This assumption will lead to the configuration present in Figure (2.1-*i*) and I will discuss it later.

7. *Spanning Bus Hypermesh*: The hypermesh network consists of communication nodes, which are constructed from routers and switches. Therefore, any node in the network can receive and forward data packets on behalf of other nodes that may not be within direct transmission range of their destination. A typical example of a 2-D hypermesh implementation is illustrated in Figure (2.1-*h*), which is the spanning bus hypercube (SBH) proposed by [10]. In addition it has been further studied by [6, 8]. The topology has very low diameter, and the average distance between nodes scales very well with network size.

8. *Distributed Crossbar Switch Hypermesh*: Another alternative way of connecting multiple computers is to simply connect every node to every other node by means of multiple channels. Such configuration can be achieved with a topology of distributed crossbar switch hypermesh cluster proposed by [29] and subsequently expanded by Old-Khaua in [30], it is depicted in Figure (2.1-*i*). This topology gives the best possibilities for parallel programming tasks, because it does not require complicated node scheduling techniques. It has the node degree equal to one and the delay of internode messages is equal for every node pair. The number of channels for an interconnection of N nodes is equal to N, which makes it unsuitable for a large number of nodes. Since the number of channels becomes very large, the bandwidth will degrade substantially.

2.2.1 Metrics for Network Topologies

Diameter: The distance between the farthest two nodes in the network. Metric for worst-case latency.

Node Degree: Number of channels connecting that node to its neighbours.

Bisection Width: The bisection width of a network is the minimum number of

channels cut when the network is divided into two equal halves.

Pin-Out: Is the number of pins per node or the number of I/Os available per router.

Cost: The number of links or switches (whichever is asymptotically higher) is an important contributor to cost. However, a number of other factors, such as the ability to layout the network, the length of channels, fanout, etc., also factor in to the cost.

Regularity: A network is regular when all nodes have the same degree.

2.3 Routing

The address header of a message carries the information needed by routing hardware inside a switch to determine the right outgoing channel, which brings the data nearer to its destination. The objective of a routing algorithm is to discover efficient paths to obtain high system throughput.

Many deterministic and adaptive routing algorithms have been proposed in the literature. Deterministic routing algorithms always supply the same path between a given source/destination pair. Adaptive routing schemes try to find dynamically alternative paths through the network in the case of overloaded network paths or even broken links. Nevertheless, adaptive routing has not found its way into real hardware yet [26]. Adaptive routing is out of the scope of this work. Since I know the network topology of the whole network, distributed routing algorithms are best fit to regular topologies since it does not relay on central authority. The same routing algorithm can be used in the communicating nodes. With distributed routing, the header of a packet is very compact. It only requires the destination address and a few implementation dependent control bits.

In this work I am going to use the hypermesh topology, which can be easily decomposed into orthogonal dimensions. It is possible to use a simple routing algorithm based on a finite-state machine like dimension order routing. This routing algorithm routes packets by crossing dimensions in increasing (or decreasing) order. The routing algorithm supplies an output channel crossing the lowest dimension for which the offset is not null. Dimension-order routing produces deadlock-free routing algorithms [1]. A detailed description of the algorithm can be found in the implementation section 8.4 in Chapter 8.

2.4 Switching

The term switching refers to the transfer method of how data is forwarded from the source to the destination in a network. Two main packet switching techniques, as depicted in Figure (2.2) and Figure (2.3), are used in today's networks, store & forward and cut-through switching respectively. The first technique transmits a packet completely across one channel before the transmission across the next channel started. Since the packet may be competing with other messages for access to a channel, a queuing delay may be incurred while waiting for the channel to become available. This mechanism needs an upper bound for the packet size and some buffer space to store one or several packets temporary [31, 26, 2]. This is the common switching technique found in LAN/WANs, because it is easier to implement and the recovery of transmission errors involves only the two participating network stages.

Newer SANs like ServerNet, Myrinet and QsNet use cut-through switching (also referred to as wormhole switching), where the data is immediately forwarded to the next stage as soon as the address header is decoded. In Figure (2.3), one sees packets transmission over their channel is pipelined, with each phit being transmitted across the next channel as soon as it arrives. A phit is the unit of information that can be

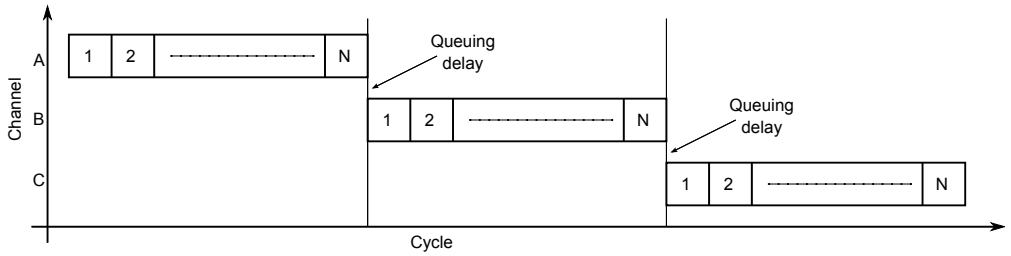

Figure 2.2: Time space diagram showing store and forward packet switching. The vertical axis shows space (channels) and the horizontal axis shows time (cycles).

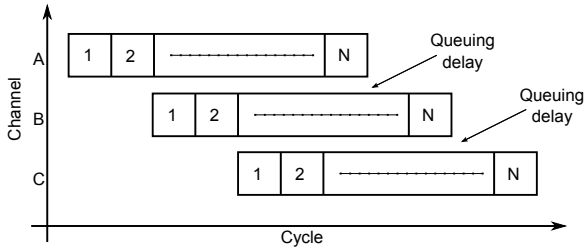

Figure 2.3: Time space diagram showing cut-through packet switching. The vertical axis shows space (channels) and the horizontal axis shows time (cycles).

transferred across a physical channel in a single clock cycle. Cut-through switching hubs exhibit slightly shorter latency than store-and-forward switches. In addition, it requires only a small amount of buffer space which is an advantage of wormhole switching. However for wireless environment, error handling is more complicated, since more network stages are involved due to packets or flits blocking as traffic increases [32]. Corrupted data might be forwarded towards the destination before it is recognized as erroneous.

2.5 Wireless MAC Protocols

A crucial part of a wireless communication system is the MAC protocol. The MAC protocol is responsible for regulating the usage of the communication medium, and this is done through a channel access mechanism. A channel access mechanism is a way to divide the main resource between nodes, the radio channel, by regulating the use of it [33]. MAC for wireless networks can be categorized into three groups [34,

35]. The fixed assignment set (Channel Partitioning set) divide channel into smaller "pieces" (time slots, frequency) and have schemes like Time Division Multiple Access (TDMA), Code division multiple access (CDMA) and Frequency Division Multiple Access (FDMA). These protocols lack the flexibility in allocating resources and thus have problems with configuration changes. This makes them unsuitable for dynamic and bursty wireless packet data networks.

The random assignment class (Contention based schemes) such as pure Aloha [36], slotted Aloha, carrier sense multiple access with collision avoidance (CSMA/CA), and non/p/1-persistent CSMA [37], etc., are very flexible instead and is what is predominantly used in wireless LAN protocols. The demand assignment (Taking turns) with schemes like Token Ring, attempt to combine the nice features of both the above and tightly coordinate shared access to avoid collisions. However, special effort is needed to implement them in the wireless case (E.g. Token Ring needs to know its neighbours).

As described in literature [37, 38, 39], a CSMA protocol works as follows. A station desiring to transmit senses the medium. If the medium is busy (i.e., some other station is transmitting), the station defers its transmission to a later time. If the medium is sensed as free, the station is allowed to transmit. These kinds of protocols are very effective when the medium is not heavily loaded, since it allows stations to transmit with minimum delay. Nevertheless, there is always a chance of stations simultaneously sensing the medium as free and transmitting at the same time, causing a collision. Subsequent variations like p-persistent and nonpersistent CSMA significantly improve the performance. In *p-persistent* CSMA, the station senses the broadcast medium and if it is idle, then it transmit a packet. If the medium is not idle, then it waits until it becomes idle. Once the medium is idle it sends a packet with probability p. Without a scheme like exponential backoff for collision resolution, p-persistent CSMA can be unstable when offered loads are high, as many stations begin transmission simultaneously when the current transmission ends. In *non-persistent CSMA,* a station will set a random time interval when it senses that

the channel is busy and tries to transmit again after that instead of continuously monitoring the channel. Packet transmission may be successful or not (collision). An acknowledgement approach or the timeout scheme is used to detect a collision.

The latter case will cause significant delay. In order to overcome the collision problem, two extensions to CSMA has been introduced, collision detection (CSMA/CD) and collision avoidance (CSMA/CA). In the former the node reads what it is transmitting, if there are differences, the node detects a collision (and thus immediately learns of transmission failure) and stops transmitting to reduce the overhead of a collision. In collision avoidance, the sender waits for an Inter Frame Spacing (IFS) before contending for the channel after the channel becomes idle [37, 38, 39].

2.6 Channel/Interface Assignment Strategies

This section will present a taxonomical classification of channel assignment strategies possible for wireless mesh networks. Channel assignment in wireless networks environment consists of assigning channels to the radio interfaces, which directly determines the efficiency of the frequency utilization. Channel or Interface assignment strategies can be classified into Fixed or static, dynamic, and hybrid strategies [40, 21, 41, 42, 43, 44].

2.6.1 Static Assignment

Static assignment strategies assign each interface to a channel either permanently, or for long time intervals with respect to the interface switching time. Fixed channel assignment can be further classified into two schemes common channel and varying channel assignment [45, 23].

1. *Common channel assignment mechanism:* In this mechanism, radio interfaces of all nodes are assigned to a common set of channels as in [45]. For instance, each communicating node has two radio interfaces, both interfaces have been assigned to the same two channels at every node. The benefit of this technique is that the connectivity of the network is the same as that of a single channel scheme. However, references [40, 21, 41] argue that the use of multiple channels increases network throughput. Moreover, they added the gain of using multiple interfaces per node might be limited in scenarios where the number of nonoverlapping channels is much greater than the number of network interfaces used per node. Note that the scenario where a single channel and a single interface is used, is a special case of the static common channel assignment strategy.

2. *Varying channel assignment mechanism:* In this mechanism, as described in [23, 46, 47] radio interfaces of different nodes may be assigned to a different set of channels. However, network partitions may arise and topology changes increase the length of the routes between nodes. Therefore, the channel assignment needs to be done carefully.

Static assignment strategies perform very well if the interface switching delay is large. In addition, when the number of available interfaces is equal to the number of available channels, interface assignment problem can be done easily [23, 40]. With static assignment, nodes that share a channel on one of their interfaces can directly communicate with each other, while others cannot. Thus, the effect of static channel assignment is to control the network topology by deciding which nodes can communicate with each other.

In this work, I am interested in this type of channel assignment, since it is the best fit to the hypermesh topology I have chosen it to implement in my proposed network. I will assign the number of channels for feasible conflict free channels in such a network. Multi-channel and multi-radio interfaces are used to improve

network capacity by operating radios on non-overlapping channels (i.e., channels that can be used simultaneously) to reduce or eliminate collisions and improve the throughput. Although the upper limit of the capacity is unaffected by the way the bandwidth is split among multiple interfaces, in practice, with realistic MAC and routing protocols, the throughput capacity can be significantly increased by the use of multiple interfaces and by the fine tuning of protocols [48, 21].

As it will be seen in Chapter 8, the hypermesh orthogonal channels are assumed to be present along with each cluster of the topology. This will allow us to carefully assign the channels to the radios in such a way as to accommodate the network traffic. In addition, by carefully balancing the assignment of fixed channels of different nodes over the available channels, all channels can be utilized, and the number of contending transmissions in a neighbourhood significantly reduces. Moreover, the protocol can easily scale if the number of available channels increases. Details of the implementation and extensive simulations are performed in Chapter 8 to illustrate the effectiveness of our proposed scheme.

2.6.2 Dynamic Channel Assignment

In this mechanism, any radio interface can be assigned any channel, and interfaces can frequently switch from one channel to another. In this setting, two nodes that need to communicate with each other need a coordination mechanism to ensure they are on a common channel at some point of time. There are many schemes using multiple channels to realize the MAC. Nasipuri's scheme [25] is one of the first multi-channel CSMA protocols, which uses soft channel reservation. If there are N channels, the protocol assumes that each host can monitor all N channels simultaneously with N transceivers. A host ready to transmit a packet searches for an idle channel and transmits on that idle channel. Among the idle channels, the one that was used for the last successful transmission is preferred. Others like [49]

showed that the coordination mechanism might require all nodes to visit a common channel periodically, or require other mechanisms such as the use of pseudo-random sequences [50, 41].

The benefit of dynamic assignment is the ability to switch an interface to any channel, thereby offering the potential to cover many channels with few interfaces. However, the key challenge with dynamic switching strategies involve channel switching delays typically on the order of milliseconds (as in commodity 802.11 wireless cards), and the need for coordination mechanisms for channel switching between nodes.

2.6.3 Hybrid Channel Assignment

Hybrid assignment strategies combine static and dynamic assignment strategies by applying a static assignment for some interfaces and a dynamic assignment for other interfaces. Hybrid strategies can be further classified based on whether the interfaces that apply static assignment use a common channel approach, or a varying channel approach [40, 51]. Wu et al. [19] proposed a protocol that assigns channels dynamically, in an on-demand style. Their approach is based on assigning one interface from each node permanently to a common control channel, the other interface can be dynamically switched among other channels. Hybrid assignment strategies are attractive as they allow simplified coordination algorithms supported by static assignment while retaining the flexibility of dynamic assignment. However it does not fit well with the targeted topology, so it is out of the scope of this work.

2.7 Channel Model

For the design of wireless systems, where the signal is distorted due to physical phenomena, it is necessary to characterize the channel, using a channel models. Wireless networks are inherently more difficult and computationally expensive to simulate than fixed wired networks. The notion of a fixed link is replaced with an error-prone broadcast channel. Bit errors in wireless networks are orders of magnitudes higher than fixed wired networks and vary with the received Signal-to-Noise-and-Interference Ratio (SINR) or Signal to Noise Ratio (SNR), usually measured in decibel (dB). SNR is actually the ratio of what is wanted (signal) to what is not wanted (noise).

In wireless channels, the state of the channel may change within a very short time span. This random and drastic behaviour of wireless channels turns communication over such channels into a difficult task. In addition, wireless channels may be further affected by the propagation environment encountered. Many different propagation environments have been identified, such as urban, suburban, indoor, underwater or orbital propagation environments, which differ in various ways.

A channel model represents signal inputs and outputs should be taken into account to speed up the process of estimating the performance of a communication system. The channel output signal *y(t)*, is different from its input signal *x(t)*, regardless of the nature of the channel. A mathematical model of the received signal *y(t)*, depending on the sent signal *x(t)* and all influencing factors has been presented in Figure (2.4). The difference might be deterministic or non-deterministic, but it is typically unknown to the receiver [52, 53].

It is not always clear what is referred to as a wireless channel in a communication system since there are multiple instances in the transmission and reception process of a signal. Figure (2.5) represents the most commonly referenced channels (as

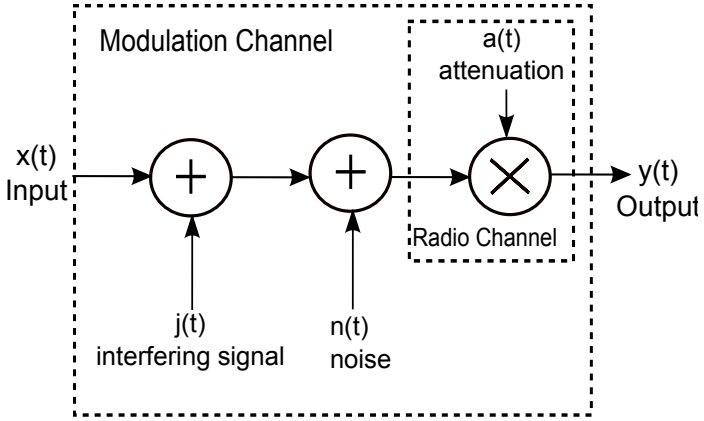

Figure 2.4: Mathematical model of the modulation channel

referred in [52, 53, 54]) to clarify different notions related to the concept of wireless channels in digital communication systems.

1. The **Transmission Channel**: The transmission channel is the medium between the transmit antenna, and the receive antenna. The signal transmitted consists of the information modulated on top of the carrier frequency [54, 55].

2. The **radio channel**: consists of the propagation channel and both the transmitter and receiver antennas. As described by [52, 56, 53, 55], the radio channel influences the received signal only by a multiplicative factor, the attenuation (loss of a signal's power) *a(t)*, as given in Figure (2.4). Analytically it is useful to distinguish between three different effects that result in an overall attenuation of the transmitted signal.

    a) The first effect is called *path loss*. It is a deterministic effect depending only on the distance between the transmitter and the receiver [57]. It is the reduction or loss in signal power as it propagates through space. It plays an important role on larger time scales like seconds or minutes, since the distance between transmitter and receiver in most situations does not change significantly on smaller time scales.

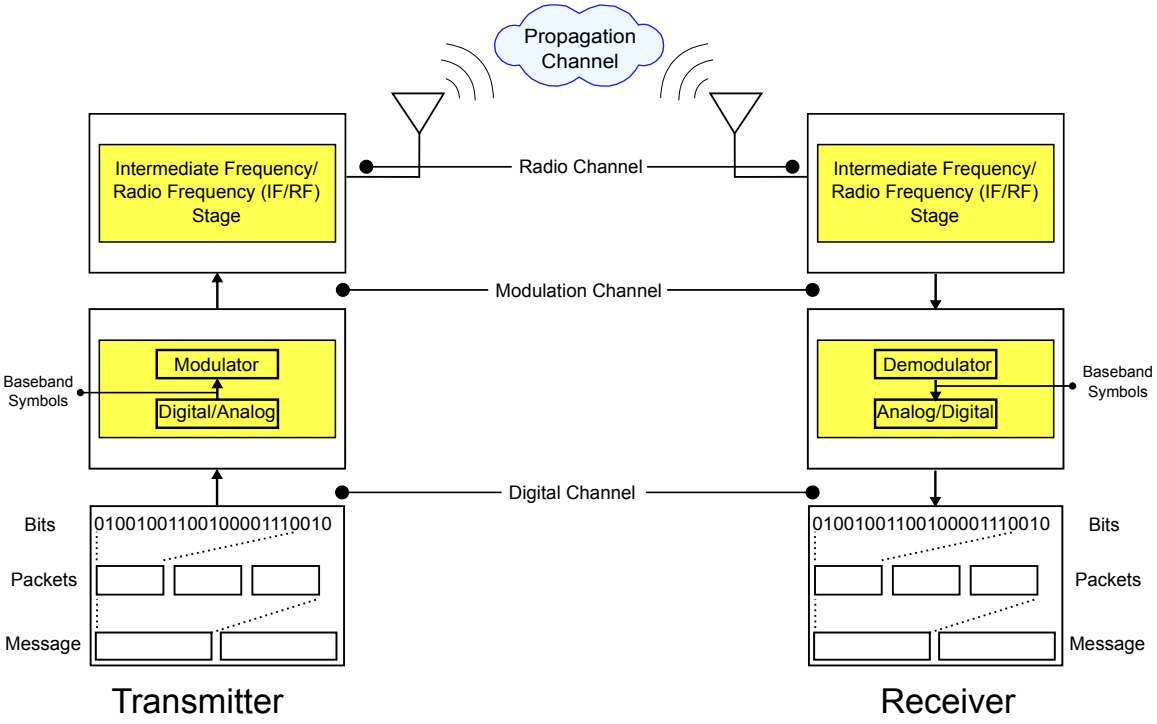

Figure 2.5: Channel model classification: propagation channel, radio channel, modulation channel, and digital channel

b) The second effect is called *shadowing*. Shadowing is not deterministic. It is due to obstacles affecting the signal propagation, some times called shadow fading. It varies on the same time scale as the path loss and causes fluctuations of the received signal strength at points with the same distance to the transmitter. However, the mean over all these points yields the signal strength given by path loss only.

c) The third effect is called *fading*. Fading is a phenomena occurring when the amplitude and phase of a radio signal change rapidly over a short period of time or travel distance. A multipath propagation environment always cause fading, by the environment reflecting the transmitted electromagnetic waves, such that multiple copies of this wave interfere at the receiving antenna [58]. Fading is generally divided into several categories. A flat fade is one where all frequency components of the signal are affected equally. A selective fade is one where only specific frequencies are affected, so the received spectrum appears to have notches. A fast

fade is one where the channel dynamics are faster than the information rate [54, 53].

3. The **modulation channel**: consists of the radio channel plus all system components (like amplifiers and different stages of radio frequency circuits) up to the output of the modulator on the transmitter side and the input of the demodulator on the receiver side. While the effects of the radio channel have an attenuating impact on the relationship between the transmitted and received signal, the modulation channel has an additive impact on this relationship. Two major sources of effects are modelled in general. The first one is noise such as thermal noise or impulse noise. Noise is always stochastic in nature and varies with time. It is denoted by *n(t)*. The second effect corrupting the received signal in an additive manner is interference. Other RF transmitting electronic devices cause interference. As it is with the noise, interference has a stochastic nature and varies with time [53, 54, 55]. It is denoted by *j(t)* in Figure (2.4).

4. The **digital channel**: consists of the modulation channel plus the modulator and demodulator. It relates the digital baseband signal at the transmitter to the digital signal at the receiver, and describes the bit error patterns. At the channel level no further effects come into play, instead, the corrupted signal is interpreted at this level as a bit sequence and if the signal has been corrupted too heavily, the interpreted bit sequence differs from the true bit sequence intended to convey. The input to this channel is bits, which might stem from packets. The bits are grouped then turned into analog representations, so called symbols. These symbols belong to the baseband. This analog signal is then passed to a modulator, which modulates these baseband signals on top of the carrier frequency [52, 56, 53, 59, 55].

For a digital communication system a couple of performance metrics are in common use, such as the Symbol Error Probability (SEP) or the Bit Error Probability (BEP).

Both performance metrics relate to the digital channel, the BEP relates to the interpreted bit stream, while the SEP relates to the stream of symbols formed from this bitstream. Both metrics depend on the average signal power ratio between the received signal power $y^2(t)$ and the noise and interference powers ($n^2(t)$ and $j^2(t)$). This average signal power ratio is given by the Signal-to-Noise-and-Interference Ratio (SNIR). Note that the attenuating influence of the radio channel is already included in the received signal power $y^2(t)$.

If the average SNIR of a link is available, also the average error rates like symbol error rate (SER) or bit error rate (BER) can be obtained. In general, the relationship between SNIR and error rates or error probabilities are not linear, instead it is highly complex and depends on many details. Therefore, a link budget analysis of all effects regarding the receiver SNIR is required for investigations of the performance of any wireless communication systems. The link budget is simply a balance equation of all the gains and losses on a transmission path [57]. The link budget usually includes a number of product gains/losses and "margins".

- Product Parameters in the Link Budget:

    1. Transmit Power: Simply the Effective Isotropic Radiated Power (EIRP) of the transmitter.

    2. Antenna Gain: A measure of the antenna's ability to increase the signal.

    3. Receive Sensitivity: The lowest signal a receiver can receive and still be able to demodulate with acceptable quality.

- Typical Margins in the Link Budget is Fade Margin, which accounts for multipath fading

While the link budget is not the primary quantity of interest in simulations, it does establish a range of values of S/N or Eb/N0 over which simulations for performance estimations have to be carried out. Since, developing a new channel model is out of the scope of this project. This work is more focused on what already exists in modelling a digital wireless communication channel. Moreover, we will use it to predict the performance of our communication system at the system level. For example, this knowledge is crucial in order to design and parametrise simulation models of wireless channels at the system level.

2.8 Interference in Wireless Communications

The wireless signal propagates in space, based on the laws of physics. An electromagnetic Radio Frequency (RF) signal which travels in a medium suffers an attenuation (path loss) based on the nature of the medium. In addition, the signal encounters objects and gets reflected, refracted, diffracted, and scattered. The cumulative effect results in the signal getting absorbed, traversing multiple paths and are having its frequency shifted due to the relative motion between the source and receiver (Doppler effect). Interference phenomena take place at the physical layer of the receiver node, as interfering (undesired) signals disturb the reception of a given desired signal. However, the characteristics of any interfering signal and its disturbance effects are determined by features of the interfering transmission at different layers or domains. Therefore, an interference model can be viewed as the combination of the following components.

2.8.1 Propagation Channels model

This describes the effects of radio propagation on the received signal, such as deterministic path loss, small-scale and large-scale fading. Path-loss is an attenuation of the signal strength with the distance between the transmitter and the receiver antenna. Shadowing is caused by obstacles between the transmitter and receiver that absorb power. Variation due to shadowing occurs over distances proportional to the length of the obstructing object (10-100 meters in outdoor environments and less in indoor environments). Multipath fading results in the constructive or destructive addition of arriving plane wave components, and manifests itself as large variations in amplitude and phase of the composite received signal in time [60, 61, 62]. Variation due to multipath fading occurs over very short distances, on the order of the signal wavelength.

2.8.2 Intersymbol Interference (ISI)

In radio channels for digital communication, ISI is due to multipath propagation when the delay spread of the channel is large compared to the duration of modulated symbol [63, 53]. The ISI results in a non-flat transfer function in the frequency domain such that all the frequency components in the transmitted signal may not experience similar amplitude and phase variations [63, 53, 64]. This is an unwanted phenomenon as the interfering symbols have a similar effect as noise, making the communication less reliable. ISI can often limit the effective data rate of wireless LAN transceivers.

2.8.3 Co-Channel and Adjacent-Channel Interference

Co-channel interference is caused by undesired transmissions carried out on the same frequency channel; and adjacent channel interference is produced by transmissions on adjacent or partially overlapped channels. The presence of Co-Channel and adjacent channel interference reduces the effective SNIR and therefore, the number of errors in reception is increased [65].

2.8.4 Affects of Interference and Methodology support

In the presence of multiple nodes and other interference sources, the probability of error will depend on the signal to interference ratio (SIR) instead of just the SNR. These errors may force data retransmissions which will add latency to the system and will degrade throughput. For instance in path loss models the attenuation suffered by a signal while travelling from the transmit antenna to the receive antenna. A commonly assumed model for the deterministic path loss is $^{P_r}/_{P_t} \propto d^{-n}$, where, P_t and P_r, the average transmit and receive power levels, respectively, d is the transmitter-receiver separation distance and n is the path loss exponent [53, 66]. Path loss exponents is typically ≥ 2 obtained based on measurements in free space is equal 2 and in buildings is equal (4-6) [67].

In addition, in an indoor environment as in our case, this factor is increased, because of the presence of objects such as furniture and also because of destructive interference of the transmitted signal caused by the reflected signals from these objects [53, 68]. Furthermore, multipath effect can vary the signal from 10-30dB over a short distance[64].

The components listed in the previous section appear in any interference model either as a deterministic process or as a random process, according to the scenario

considered. For instance, if fading is a relevant effect of the radio propagation environment, then the interference signal is a random process, regardless of the nature of all other components. Therefore, an appropriate fading distribution must be adopted in the interference model. Clearly, the nature of the components (either deterministic or random) collectively determines the nature of the interference modelled.

Our SystemC methodology allows for any interference model to be integrated in the communication channel. In wireless communications the channel is often modelled by a random attenuation (known as fading) of the transmitted signal, followed by additive noise [69, 61]. The key point to note in the methodology is the possibility of separation between communication and behaviour using interfaces. Interfaces may be accessed from outside a SystemC module using SystemC ports. This technique can be used to have different methods within the same interface. For instance the channel described to model noise has been built for a digital erasure channel using the channel model described in section 4.2. The channel model is sufficiently flexible to support various numbers of channel models with a configurable numbers of paths per channel. The implementation needs further extensions to include more realistic and accurate radio propagation channel models. A useful method of presenting these models is through further refinement that could enhance the physical layer from digital channel modelling level to the modulation channel and then to radio channel modelling. This should provide real implementation of different modulation techniques and can easily incorporate the interference models described above.

What is even more important, it allows for using a real trace as entry data. A trace is in fact a register of events, ordered in time, which is obtained from a real system. This approach has the advantage of providing high credibility as there is a great similarity between the model and the real system; which has a very large impact on the real performance obtained in the network.

2.8.5 Calculation of the Bit Error Probability as a function of SNIR

In wireless networks, it is an important consideration to determine whether a pair of nodes can communicate together. A link analysis has been carried out to describe the behaviour of the communication medium between the transmitter and the receiver terminals. A free space propagation representation of the communication link as in equation (2.1) has been taken into account, which represents the signal decay as a function of distance, when signals are propagated from transmitters to receivers. And it is the difference in dB between the transmitter and receiver power. The following calculation follows the one in [70, 53, 71]

$$L_{fs} = 10log_{10}\left(\frac{P_{tx}}{P_{rx}}\right) = -10log_{10}\left(\frac{G_t G_r \lambda^n}{(4\pi)^n d^n}\right) \tag{2.1}$$

Where the Free Space propagation Loss L_{fs}(line-of-sight) in dB, the transmitted power (P_{tx}) and received power (P_{rx}) in Watts. The transmitter and receiver's antenna gains G_t and G_r. The receiver sensitivity required is usually quoted in dBm, can be computed as in equation (2.2).

$$P_{rx} = P_{tx} - L_{fs} - Fade\,Margine \tag{2.2}$$

The general expression for propagation loss in dB with the assumption that the antenna gains is 0dB for a simple dipole antenna and in free space n is assumed to have the value of 2.

$$L_{fs} = P_{rx}\,[dB] - P_{tx}\,[dB] = 10 \times n \times log_{10}\,(4\pi^{d}/\lambda)\;\; dB \tag{2.3}$$

where:

d =Distance between Transmitter and Receiver
$\lambda = {}^{c}/_{f}$ the free space wavelength at the carrier frequency
c = speed of light (3×10^8 m/s)
f = frequency (Hz)

Fade Margin Multipath phenomenon tends to be dynamic and is mostly present in an indoor environment. Signal reduction due to multipath fading is reported in many literature such [53, 72] and is normally in the range of 20 to 30dB. Practically in designing a wireless system, a Fade Margin substitution of the power loss due to the multipath is required.

2.8.6 Signal to Noise Ratio and Bit error rate

The most important metrics of performance in digital communication systems is the plot of the bit-error probability P_B which represents the (BER) versus Bit energy per noise-density of the signal ${}^{E_b}/_{N_0}$. Figure (2.6) illustrates the waterfall like shape of most such curves. For the purpose of link budget analysis, the most important aspect of a given modulation technique is the signal to noise ratio necessary for the receiver to achieve a specified level of reliability in terms of P_B (BER).

The probability of a bit error, P_B, is defined as:

$$P_B = Q(z) = Q\left(\sqrt{\frac{E_b}{N_0}}\right) \tag{2.4}$$

Where $Q(z)$, is called the complementary error function or co-error function. This complementary error function is numerically equal to the area under the "tail of the

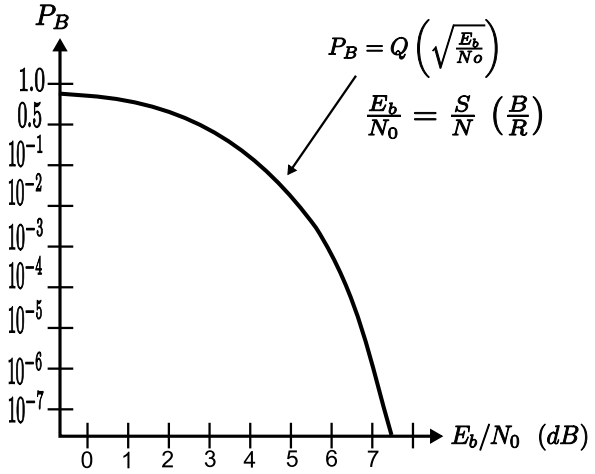

Figure 2.6: General shape of P_B versus$^{E_b}/_{N_0}$curve

Gaussian". It is closely related to the complementary error function $erfc(z)$ and error function $erf(z)$:

$$erf(z) \equiv \frac{2}{\sqrt{\pi}} \int_0^z e^{-x^2} dx \qquad z \geq 0 \tag{2.5}$$

$$erfc(z) \equiv \frac{2}{\sqrt{\pi}} \int_z^\infty e^{-x^2} dx = 1 - erf(z) \qquad z \geq 0 \tag{2.6}$$

The *Q-function* is related to these functions by

$$Q(z) = \frac{1}{2} erfc\left(\frac{z}{\sqrt{2}}\right) \qquad z \geq 0 \tag{2.7}$$

Using the previous analysis a sample *BER* model from literature [53] for a specific modulation scheme, Quadratic Phase-Shift Keying ($QPSK$) is given by Equation (2.8). The *BER* assuming white noise AWGN is given by

$$BER = \frac{1}{2} erfc\left(\sqrt{E_b/N_0}\right) \tag{2.8}$$

where the maximum thermal noise power within a given bandwidth B is given by

$$N_0 = kTB \tag{2.9}$$

In the radio and microwave bands, the spectral density is taken as N for one sided spectrum, and as $\frac{N_0}{2}$ for two side spectrum

where:

N_0 = Noise power (watts)
k = Boltzmann's's constant (1.38×10^{-23} Joules/Kelvin)
T = System temperature in Kelvins, usually assumed to be 290K
B = Channel bandwidth (Hz)
SNR gives the relation between the received signal power and the noise power as given by

$$SNR[dB] = 10 \times Log_{10}(SignalPower[W]/NoisePower[W]) \tag{2.10}$$

Where, E_b/N_0, in equation (2.8) is just a normalised version of SNR [53]. E_b is bit energy and is equal to signal power S times time T_b. The reciprocal of the bit rate R is $1/R$ which represents the bit time T_b, and then we can write:

$$\frac{E_b}{N_0} = \frac{S\,T_b}{N/B} = \frac{S/R}{N/B} \tag{2.11}$$

Equation (2.11) can be rewritten to emphasise that E_b/N_0 is just a version of S/N normalised by bandwidth and bit rate, as follows:

$$\frac{E_b}{N_0} = \frac{S}{N}\left(\frac{B}{R}\right) \tag{2.12}$$

The ratio E_b/N_0 is important because the bit error rate for digital data is a (decreasing) function of this ratio. Given a value of E_b/N_0 needed to achieve a desired error rate, the parameters in the preceding formula may be selected. Note that as the bit rate R increases, the transmitted signal power, relative to noise, must increase to maintain the required E_b/N_0 [73].

3 System Level Network Modelling Background

In this chapter, I will briefly describe network simulation methods and the proposed SystemC design methodology to meet my aim to develop digital wireless SAN. And then, I will address the simulation measurement setup of interconnection networks to avoid sources of errors, when estimating network performance. Moreover, the statistical approach for assessing the accuracy of the measurement is presented.

3.1 Network Modeling

Network developing environments, or commonly called network simulators, actually propose substantial support for modelling and simulating different protocols (i.e., TCP, routing, and multicast protocols) over wired and wireless networks. Network simulators reproduce the functional behaviour of protocols by managing time information about transmission and simulating packet losses due to congestion or link failure. There are many well accepted network simulators available both commercial OPNET [74]; QualNet [75] and as open source projects OmNet++ [76]; NS-2 [77] whereby designers can accurately implement protocols, manipulate bytes, packet headers, and implement algorithms that run over large data sets. Libraries with a wide range of communication protocols are generally available and constantly up-

dated and, thus, designers can efficiently design, modify, and test various network parameters or configurations, for quickly exploring different network scenarios. However, the main drawback is that network simulation tools describe functionalities without reproducing the interaction between different components within the single node, as in actual systems. This fact limits the reusability of the functional description for the design, validation, and synthesis of the actual system as traditionally happens with hardware description languages [78].

It is important to note that domain specific tools do not provide all the capabilities required for a comprehensive simulation. Any network simulator does not provide mechanisms for HW description. Co-simulation of analogue parts using Matlab, Simulink [79] and SPICE (Simulation Program with Integrated Circuit Emphasis) are common, Network protocols parts in Opnet, NS-2 and OmNet++ are also common and digital parts in HDL would be too simulation intensive for large and complex systems. On the other hand, even the most versatile tool for Transaction-Level-Modelling TLM Hardware/Software (HW/SW) simulation (i.e., SystemC) does not provide models for well-known communication protocols and, therefore, the designer has to completely rewrite them. However, SystemC incorporates the system development into a homogeneous design environment [13, 78, 80].

To overcoming their limitations in modelling and simulating TLM HW/SW systems, in this work, I propose the development of a network simulator based on SystemC [81] modelling and design methodology. SystemC [11, 81] is gaining increasing attention for its great flexibility in describing devices at different abstraction levels and for its interoperability with other HDL's, for example, System Verilog and VHDL. The advantage of the use of a single tool is the higher simulation speed, since synchronization between different tools is not required. The drawback is that models of well-known protocols have to be completely rewritten in the system description language together with models of nodes outside the design scope.

3.2 The SystemC design Methodology

SystemC, which is a design and modelling methodology, has been chosen to develop my models in this work. It represents an emerging standard modelling-platform based on C++ plus a simulation kernel. SystemC builds the bridge between hard- and software design. It extends C++ with macros, functions and other constructs to allow a hardware-oriented design. Strictly SystemC is not a language, its a class library that can be used to model systems in a homogeneous environment all the way from requirement capture to system partitioning, cycle accurate modelling and backend implementation.

Transaction-level modelling is an approach to modelling digital systems where details of communication among modules are separated from the details of the implementation of the functional units or of the communication architecture [82, 83, 84, 85, 81, 86, 80]. On the other hand, Behavioural- level modelling is a model of a system at any level of abstraction that includes timing information [12]. The inherent modularity of SystemC allows designers to model and refine the individual elements of a system with varying levels of detail and timing accuracy. It begins with a functional untimed implementation which can be utilized to identify the required elements verifying functional correctness (*behaviour executable model*). However it does not provide sufficient details regarding the final implementation or resulting system performance.

Through the continuation of the refinement process which is facilitated by the SystemC design methodology, a model can be extended to an approximate timed model by incorporating timing annotations to approximate the time required to perform specific operations. Moreover, an RTL model implementation defines the exact cycle-by-cycle timing where all operations are defined at the bit level, it can be carried out using the same methodology by using the synthesisable subset of the SystemC or by exporting the SystemC code into VHDL or Verilog for synthesis. Sys-

temC supports several techniques for addressing the complexity of modern designs. Today's system designer can use several approaches for attacking the complexity issues that come with complex system design, these approaches are abstraction, design reuse, team discipline, project reuse and automation [82, 84, 86, 80].

SystemC has been chosen because it provides a homogeneous design flow for complex designs (i.e. SoC and IP based design), where system modelling at the early stages of the design becomes increasingly important. It allows designers to reconfigure the platform architecture quickly to explore the use of different IP blocks and solution trade-offs among performance, power consumption, and area. Moreover, it has been used to model digital systems due to its versatility, high modelling fidelity, early verification of the entire system, support of homogeneous system models and ability to express system functionality at various levels of abstraction. Also, because SystemC can describe hardware at high levels of abstraction, it also provides a faster simulation time, when compared to VHDL [87], which is an important feature when a large design space needs to be explored.

It is important to recognise that the SystemC methodology does not impose a top-down or bottom-up or even middle-out design flow. With the SystemC design flow, design is not converted from a specification level description to a synthesisable level in one large effort. However it is recognised that most design flows are iterative, as it is rare that all modules within a system are modelled at the same level of abstraction. For example, parts of a system can be modelled at a low level of abstraction and combined with parts described at a higher level in the same design [11, 83, 84, 88]. The design is refined incrementally in small parts to add the necessary hardware constructs. Using such methodology a designer can more easily implement design changes and verify model.

Modules are the basic building blocks of SystemC. The behaviour of a module is specified by defining one or more processes. Processes are declared as special functions of modules and can be reactive to any input signal or to an event. A process

should be declared either as a method or as a thread. The difference is that blocking statements are not allowed in the method processes, making the whole body of the method to be executed entirely. Processes and modules can communicate via ports, signals, events and variables [11, 83, 84, 89].

The simulation kernel of SystemC invokes the concept of delta-time delays to model the concurrent execution of the hardware systems. So, if processes that are running concurrently change the values of some signals, the values of the signals shall be updated simultaneously after a delta-time delay [84].

A system-level design flow is presented in Figure (3.1). The design phases could be defined as follows:

1. *System Requirements and Specification Capture:* The system is modelled at a high level in order to check the pure functionality, disregarding details related to the target architecture. An abstract C/C++ conceptual model is the beginning of the SystemC design flow. Simulation of this executable functional specification allows a validation of the system functionality. Several design concepts might come up from developers for realising this functionality such as the specified requirements that are expected to be satisfied [11].

2. *Development of a behaviour Untimed Functional Model:* to investigate whether the effects of the proposed concepts are indeed as expected. This model involves making abstractions from implementation details since these details cannot be known in advance. At this level, a first functional partitioning between data and control is performed but it is not determined yet which modules will be implemented in Hardware (HW) and which will be implemented in Software (SW). Data transfers are described through abstract data types and point-to-point communications between modules. The functional model is purely representative of the functionality of the target system model, with no notion of timing or architecture. The behaviour is modelled as a set of un-

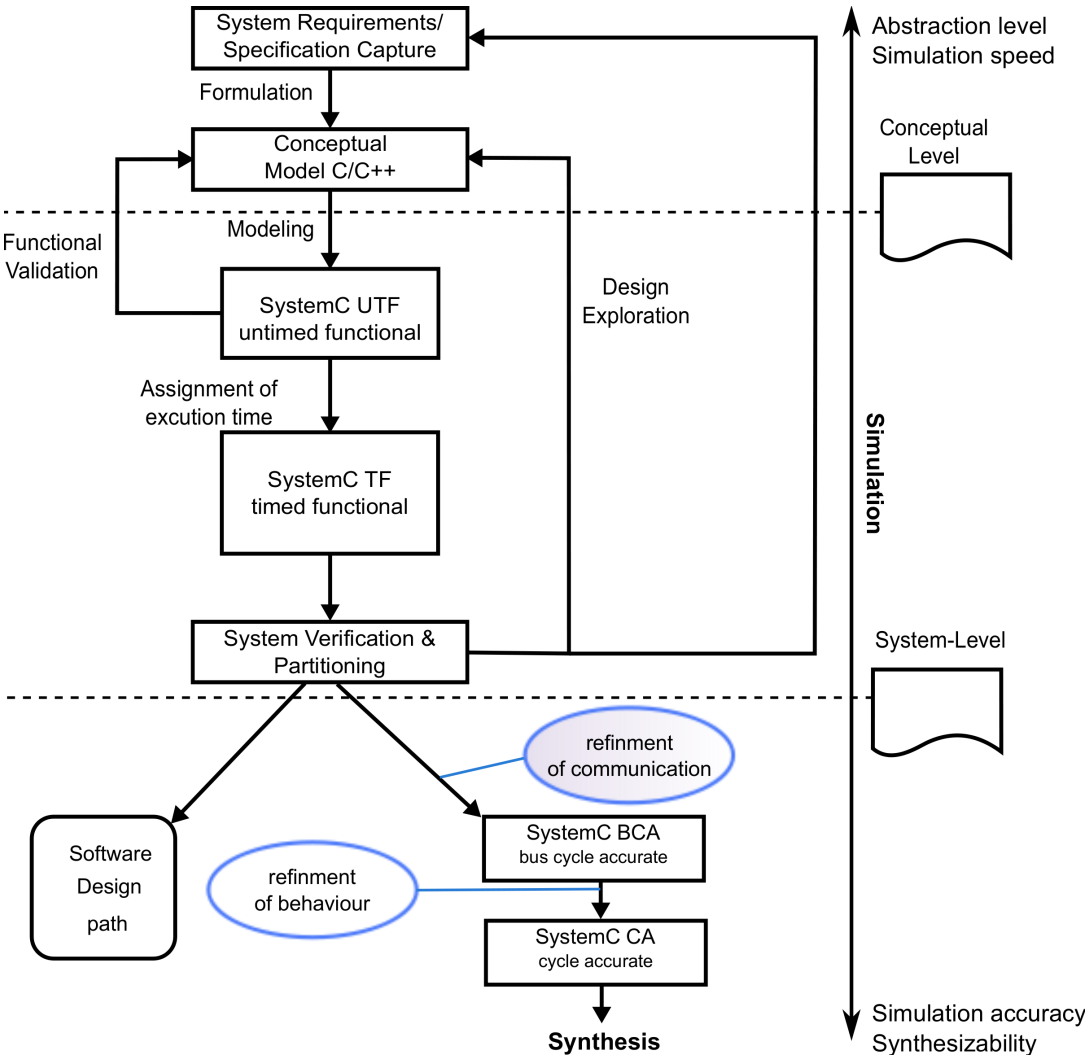

Figure 3.1: SystemC design Methodology

timed event-driven function calls, as in common application SW. The untimed model has absolutely no timing information related to the micro-architecture, i.e. there is no clock in an untimed model system [90, 78].

3. *SystemC Timed Functional Model:* The extended behavioural model allows us to evaluate whether the design solutions for realising the functionality will satisfy the performance requirements. Such high-level system specification is refined by replacing native C++ data types to bit accurate SystemC data types, introducing concurrency, timing mechanisms and processing delay figures. In addition, a first HW/SW partitioning is performed according to several constraints (e.g., performance, cost, and reuse of available components). Data transfers are described in terms of bit-width and message size in order to estimate bus bursts. SystemC facilitates the modelling of time using unsigned integers of length 64 bits or more, which is higher than VHDL and Verilog [11, 88, 78]

4. *System partitioning:* If all requirements of certain design alternatives are satisfied according to the simulation of the abstract model, these alternatives may serve as a basis of realising the system. After refinement of higher abstraction level model hardware/software partitioning and architectural exploration take place. The unified and common C++ based system description capabilities ease the architectural exploration, i.e. analysis and partitioning hardware or software components to allow a performance optimisation of the system. HW/SW partitioning gives a first direct impact on both throughput and latency parameters. The decision to implement a given functionality either by HW or SW components may affect the performance of the system and, therefore, the amount of data offered to the network in the time unit and the delay between consecutive transmissions. More HW components guarantee a higher parallelism degree in data processing and, consequently, a greater amount of data can be processed and ready to be sent through the network in shorter time.

5. *SystemC RTL Model:* At this level designers gradually refine and verify the system description more accurately, toward the final RTL implementation. At this transaction level, interfaces of HW blocks are defined even if pins are still hidden. This level captures the micro-architecture details and typically has bit-level interface. The model is clocked and all timing annotations are accurate in cycle. In addition, the final implementation can be easily synthesizable by commercial synthesis tools.

3.3 Simulation Measurement

Simulation measurement or in other words simulation setup is an important step that is all too often ignored or skipped. Determining the type, and stopping criteria are the first step toward simulation measurement. The simulation can be set to run for specified fixed period of time or may stop when reaching steady state. Researchers are often interested in the long term average; specifically the cumulative moving average. However, a common pitfall is to claim the simulation results have reached steady state without assuring the degree of convergence.

There are two main sources of errors, when estimating network performance: systematic error and sampling error [91, 92, 2]. Systematic errors are errors introduced by bias in the measurements or simulation itself and for network simulation is generally a result of the initialisation of the simulator. To minimise the effect of systematic errors a suitable warm-up period for a simulation must be chosen. This warm-up period will allow a network to reach its steady state. That is the statistics of the network are stationary and no longer change with time and an accurate estimate of a particular network parameter can be determined [2].

Simulation typically begins at an initial state where there are no packets in the network; these packets will cross the network quickly, since they have less contention.

However, over the course of the simulation, buffers begin to fill up and later packets see more contention, which increase their latencies. Consequently, the initial transient from the idle state to point that the long term average is achieved skews the results. This is referred to as the steady state. Over time, the influence of the initialisation becomes minimal, and at this point, the simulation is said to be warmed up. This warm-up period is used to alleviate this bias, or to minimise the impact of systematic errors, where the collection of data does not begin until steady state is reached.

As stated by [2], there is no common method for determining the length of the warm-up period, but most approaches follow the same basic procedure:

1. Set the initial warm-up period to T_{wu} based on a heuristic.

2. Collect statistics, ignoring samples during the estimated warm-up period.

3. Test the remaining samples to determine if they are stationary. If stationary use T_{wu} as the warm up period. Otherwise, increase T_{wu} and repeat steps 2 and 3.

3.4 Simulation Stopping Criteria

Once simulation begins, a natural question is: when to stop it? Typically, a simulation experiment is stopped as soon as the relative precision of point estimates, defined as the relative half-width of confidence intervals at a specified confidence level, reaches the required level. Alternatively, the simulation may be run for a preset period of time, but this is generally a a poor practise [91, 92]. A confidence interval is a range of values in which the true solution is believed to lie , with an associated confidence level. A reasonable stopping criterion for a simulation is to

iterate until the width of the confidence interval for a performance metric of interest is acceptably small, say ε [91].

3.4.1 Description of the Method

In a given simulation run of n independent sample values X_1, X_2, ..., X_n whose mean (denoted by $\bar{X}$) is given by:

$$\bar{X} = \frac{1}{n}\sum_{i=1}^{n}(X_i) \tag{3.1}$$

$\bar{X}$ is called the sample mean. Since it is a function of random variables, which is typically an un-weighted average of the sequence X.

Equation (3.1) can be written in the form below to utilise the recursive form of the mean estimator or cumulative average form,

$$\bar{X} = \frac{x_1 + \cdots + x_n}{n} \tag{3.2}$$

let

$$x_1 + \cdots + x_n = n\bar{X}$$

And then X_{n+1} is

$$x_{n+1} = (x_1 + \cdots + x_{n+1}) - (x_1 + \cdots + x_n) = (n+1)\bar{X}_{n+1} - n\bar{X} \tag{3.3}$$

Solving this equation for $\bar{X}_{n+1}$ results in:

$$\bar{X}_{n+1} = \frac{x_{n+1} + n\bar{X}}{n+1} = \bar{X} + \frac{x_{n+1} - \bar{X}}{n+1} \tag{3.4}$$

Where $\bar{X}_0$ can be taken to be equal to *zero*.

Thus, the current cumulative average for a new data point is equal to the previous cumulative average plus the difference between the latest data point and the previous average divided by the number of points received so far. Any statistic, X, may take on many values and its precise distribution is unknown. Unfortunately, detailed observations of specific system parameters collected during typical simulations are found to be correlated and non-normal. For instance, in a mesh connected network, local traffic density is correlated with node position, as edge nodes have differing connectivities compared with central nodes. However, in this work we assume that the Central Limit Theorem holds. We do this because of the homogeneity of the systems studied (each node and its connectivity is identical to all others), and because the injected traffic to the system is homogenous in nature. We also analyse the performance of the system as a whole, rather than of specific local nodes, Thus, we can assume that the system parameters are the result of many local processes in a large homogeneous system, and the Central Limit Theorem is valid, indicating that the resultant system parameters can be assumed to be Normal in nature to first order. It is also assumed that the underlying processes being measured are stationary. The warm-up procedure described in section (3.3) ensures that this is so.

Therefore, in accordance with the central limit theorem, regardless of X's actual distribution, as the number of samples grows large in equation (3.4) $\bar{X}$ converges to a limiting value $E[X]$, called the expectation of X. For conciseness $E[X]$ will be represent by a normal random variable with mean μ , the same mean as the random

variable X itself.

Similarly, we can find a recursive point estimator for the variance. Since the mean only does not tell us all we want to know; we would also like to know something about the variance of X, as a measure of the statistical variability of X. The variance is a measure of the dispersion of a distribution. First, define:

$$s^2 = \frac{1}{n}\sum_{i=1}^{n}\left(X_i - \bar{X}\right)^2 \tag{3.5}$$

s^2 is called the sample variance.

The way of computing variance goes back to a 1962 paper by B. P. Welford and is presented in [93]. The algorithm is as follows.

Initialise $M1 = x1$ and $S1 = 0$.

For subsequent x's, use the recurrence formulas

$M_k = M_{k-1} + (x_k - M_{k-1})/k.$
$S_k = S_{k-1} + (x_k - M_{k-1}) * (x_k - M_k)$
For $2 \leq k \leq n$, the k^{th} estimate of the variance is $s^2 = S_k/(k-1)$.

Also in accordance with the central limit theorem, as n approaches infinity, s^2 converges to a limiting value $E[s^2] = E\left[(X-\mu)^2\right]$, denoted by σ^2 and the variance for statistic X is $\frac{\sigma^2}{n}$. Thus, $\bar{X}$ and s^2 are the sample mean and variance, and μ and σ^2 are the distribution mean and variance [94, 57]. The square root of the variance is the standard deviation. This is how to estimate how close the sample mean obtained from a finite length simulation to the distribution mean μ or, equivalently, how long run lengths have to be to obtain a sample mean arbitrary close to μ.

Numerically stable algorithm, which described by [93] is given below. It also computes the mean.

```
1  def online_variance(data):
2      n = 0
3      mean = 0
4      M2 = 0
5      for x in data:
6          n = n + 1
7          delta = x - mean
8          mean = mean + delta/n
9          M2 = M2 + delta*(x - mean)  # This expression uses the
               new value of mean
10     variance_n = M2/n
11     variance = M2/(n - 1)
12     return variance
```

3.5 Confidence Interval

As described in [91] for a given system mean, we would like to collect as many simulation observations as needed to be approximately $100(1-\alpha)\%$ certain that its estimate is within H units in error. Define $(1-\alpha)$ as the confidence level or confidence coefficient, which is a range of values that contains the true mean of the process with a given level of confidence. In other words it is the probability that the absolute value of the difference between the sample mean and μ is equal or less than H:

$$P\left(\left|\bar{X}-\mu\right| \leq H\right) \approx 1-\alpha \tag{3.6}$$

Then a confidence interval for the mean is defined as

$$P\left(\bar{X} - H \leq \mu \leq \bar{X} + H\right) \approx 1 - \alpha \tag{3.7}$$

The interval $\bar{X} - H$ to $\bar{X} + H$ is called the confidence interval, H is called the confidence interval half-width. Typical values for α are 0.1, 0.05, and 0.01, which translate into a confidence level of 90%, 95%, and 99% respectively. A 95% confidence interval for μ means that there is a 95% certainty level that the true mean of X lies within the intervals bounds. The confidence level $(1 - \alpha)$ is specified by the system designer; H is determined by the sample values, number of samples, and the value of α as follows:

The calculation of the confidence interval relies on the fact that the distribution of the statistic X is normal. As described by [91], quantiles of the normal distribution and the sample mean and standard error of the mean can be used to calculate approximate confidence intervals for the mean. And in this case we have $\bar{X} \pm z_{\alpha/2}\sigma_{\bar{x}}$ contains μ with probability of approximately $1 - \alpha$ where:

- $z_{\alpha/2}$ is the $1 - \alpha/2$ point for the standard normal distribution,
- $\sigma_{\bar{x}}$ is the standard deviation of $\bar{X}$,
- and n is large enough to assure approximate normality of $\bar{X}$.

When the sample size is large, the standard deviation of the population may be approximated by the standerd deviation of the sample s. Then the interval given above will be approximated by the approximate $(1 - \alpha)$ confidence interval: $\bar{X} \pm z_{\alpha/2}s/\sqrt{n}$ where $\frac{s}{\sqrt{n}}$ is the Standard Error of the Mean (SEM) is usually estimated by the sample estimate of the population standard deviation (sample standard deviation)

divided by the square root of the sample size (assuming statistical independence of the values in the sample).

H is given by the term $H = z_{\alpha/2}{}^{s}/\sqrt{n}$. The values of $z_{.05}$ and $z_{.025}$ are respectively, 1.65 and 1.96. The following expressions can be used to calculate the upper and lower 95% confidence limits, where $\bar{x}$ is equal to the sample mean, SEM is equal to the standard error for the sample mean, and 1.96 is the .975 quintile of the normal distribution:

$$Upper\,95\,Limit = \bar{x} + (SEM \times 1.96)$$

$$Lower\,95\,Limit = \bar{x} - (SEM \times 1.96)$$

In particular, the standard error of a sample statistic (such as sample mean) is the estimated standard deviation of the error in the process by which it was generated. In other words, it is the standard deviation of the sampling distribution of the sample statistic.

This thesis presents the results gathered from simulations using 95% confidence intervals. Figure 3.2 shows an example time-evaluation of the packet latency using the SystemC network simulator presented in Chapter 6. The data obtained with the simulator has some mean and variance, resulting from the transient in the system state (in this case as the system fills to an equilibrium of packet origination, transmission and reception) and the limited number of samples making up the calculation of average packet latency. The variance of the data decreases with the increasing number of samples. As it can be seen in the plot, the simulation does not seem steady even after the first 1000 packet arrivals. Figure 3.2 shows that the system appears to stabilize after around 7000 sample points. At this point the simulation of this system was stopped as the steady-state results reached a relative precision of at least 0.05 at the 0.95 confidence level, where relative precision is defined as

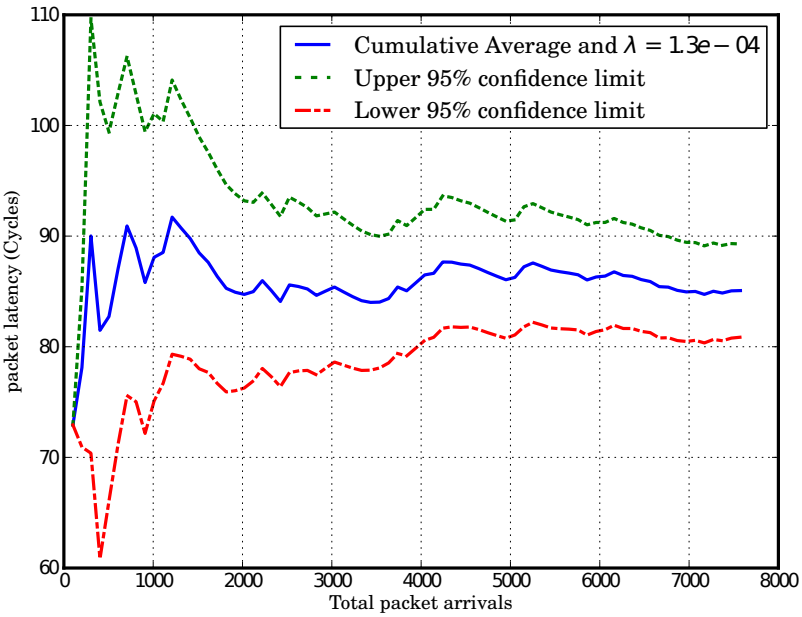

Figure 3.2: Average packet latency and 95% confidence intervals in 144 nodes bus topology near saturation vs. the number of packet arrivals. The upper and lower limits of the confidence intervals are indicated by the red and green colors.

the ratio of the current half-width of the confidence interval of mean to the current value of the estimated mean.

4 Wireless Channel Model Based on SoC Design Methodology

In this chapter, a new method to model and simulate a wireless communication system based on SoC design methodology will be presented. How well our method correctly exposes the underlying network performance of the system is directly related to the amount of detail in the simulation model. Hence there is a need to develop suitable abstractions that maintain the accuracy of the simulation while keeping the computational resource requirements low. The integration of communication modelling into the design modelling has been shown by modelling a noisy communication channel in SystemC. The channel supports different modulation techniques such as, Amplitude-shift keying, Phase-shift keying, Quadrature amplitude modulation. It supports the setting of different Signal to noise ratio and different types of interference for Point-to-Point and Point-to-Multipoint platforms based on SystemC design methodology.

4.1 Introduction

The design complexity of digital communication systems has risen significantly in recent years [11]. This trend is likely to continue, as devices continue to incorporate an ever-growing number of components in order to support new functionalities whilst

providing inter-operability with the large plethora of standards and protocols from previous and present state of the art systems. System designers, on the other hand, are facing increased pressures from:

1. A reduced time to market, brought by a reduced sale window.

2. An increase in design responsibilities and technical expertise, due to the introduction of SoC design methodologies, which in the future will include a wireless.

3. A lack of a unified design and test environment.

4. An increasing cost of failure (from lost opportunity and large re-engineering costs).

While new design methodologies have emerged for the design of complex systems, in particular for SoC. They have not, to the best of my knowledge, been extended to incorporate, within the same framework, the design of communication system (as found in digital radio communications, for instance). This is surprising, considering the fact that the system performance cannot be accurately determined otherwise, and that this performance is used to guide the designer through the architectural exploration phase and arrive at the final implementation. Leaving the development of the wireless system outside of this integrated methodology is likely to produce bad or poor results.

The modelling of noisy communication channels is not new. Analogue channel modelling using Matlab, Simulink and Opnet is common. Even behavioural modelling has been proposed in [95], but has not been incorporated into a homogeneous design environment. Paper [95] is one of the few papers so far about modelling using SystemC. The paper shows a systematic approach to modelling and simulating an

OFDM transceiver for wireless LAN using SystemC.

In this work I propose a methodology for the modelling of noisy communication channels so that they can be incorporated from the early phases of the design. My methodology is based on the popular SystemC design methodology, which provides a consistent framework for the design and modelling of complex systems at numerous levels of abstraction (which encompass all levels from system specification to implementation).

This part of the thesis will provide a first step towards this methodology by introducing a simple noisy digital channel in SystemC, that can be used to model the whole system interactions. It will show the construction of the wireless communication system such the one shown in Figure (4.1), which represents two communication nodes that exchange information through noisy communication channel. The channel model that I develop supports different modulation techniques such as, Amplitude Shift Keying (ASK), Phase Shift Keying (PSK), Quadrature Amplitude Modulation (QAM). It supports the setting of different SNR and different types of interference.

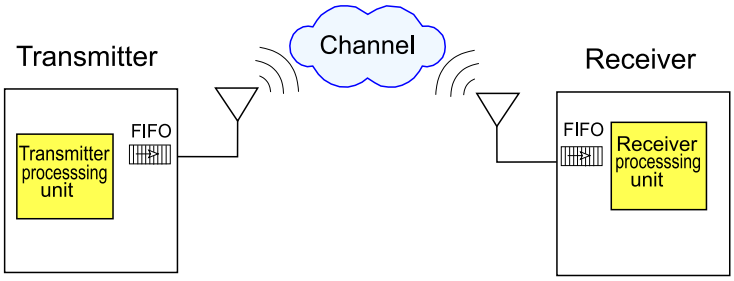

Figure 4.1: Block Diagram of Wireless communication system consists of two nodes

The main contribution of this part of the work is the integration of communication modelling into the design modelling at the early stages of the system development. To my knowledge, this is the first time that some one done this modelling. The remainder of this chapter continuous with description of how the noise are going to be generated using the channel model discussed in chapter 2. Then section 4.3 the

channel module implementation, which describes the implementation of the noisy digital communication channel by incorporating the noise generation process into it and highlights the details of using it to construct a P2P communication system. Section 4.4 describes the work undertaken on the modelling of P2M communication system. Section 4.5 describes the system demonstrator. And the final two sections will describe the simulation results with discussion and conclusion.

4.2 Channel Model

According to the mathematical model shown in Figure 2.4 in Chapter 2 the transmitted signal is influenced by three effects: interference signal, attenuation and noise, which have been discussed in Chapter 2. For a digital communication system the question arises, how these signal distortions translate into noticeable service degradations of running applications. Moreover, how to use the channel model presented in Chapter 2 to predict the performance of my communication system at the system level. For example this knowledge is crucial in order to design and parametrise simulation models of wireless channels at the system level. And also applying this knowledge in designing of communication protocols.

4.2.1 Noise Generation Process

The performance of any communication system is effected by noise and interference from other sources, which play a crucial role in communication systems. In practise noise and interference represent the number of errors occurring in the system. The simplest model of the digital noise is just to consider the impulsive noise, in which the individual bits or packets are modified with a given probability. The other type of error is lost bits, which is only happen, if the system is out of synchronisation.

However it is assumed that, there is no model of a Phase-Locked Loop (PLL) in the system, and it is assumed to be PLL locked. So as consequence, I will not going to model the second type of error at this stage, which will be left as an exercise for future work. Here the model is done in conventional way and, I assumed a memory-less system, which implies the Markov property. Therefore the exponential distribution is good fit for the model. Though other distributions such as Pareto distribution can be easily coded as well. Here for simplicity I used the exponential distribution, but it is not accurate to model real world systems. And it gives the inter-arrival times for errors. Then by solving the Probability Density Function (PDF) of the exponential distribution for t, which represents the inter-arrival time of errors. And therefore, a random experiment is performed to determine the position, where errors are injected in the bit stream.

For each modulation technique there is a waterfall curve, which relates a SNR to a specific BER as shown in Figure (4.2). The PDF of the exponential distribution is given by:

$$f(t) = \lambda e^{-\lambda t} \text{for } 0 < t < \infty \tag{4.1}$$

where $BER = \mu = \frac{1}{\lambda}$ and μ is the mean and variance for the exponential distribution. Therefore solving it for t to determine the position of errors and introducing them to the channel follows these steps:

1. Choosing system SNR which correspond to one of the modulation schemes.

2. Getting the corresponding BER from the waterfall curves relate BER to the SNR.

3. Calculate $\lambda = \frac{1}{BER}$.

4. Perform a random experiment according to the PDF of the exponential distribution.

5. Inject the error in the bit stream.

The process steps from (1 to 5) of solving the PDF can be easily traced on the graph shown in Figure (4.2).

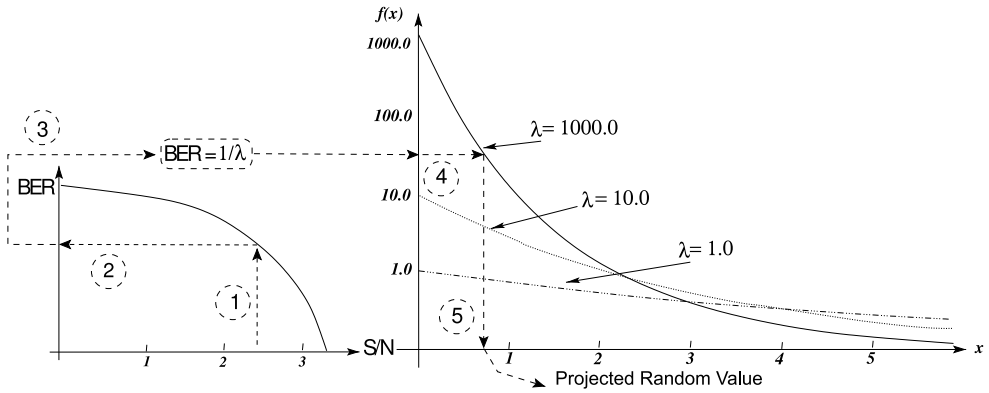

Figure 4.2: Stochastic projection of Noise, 1- Choosing S/N, 2- Getting corresponding BER, 3- Calculate $\lambda = \frac{1}{BER}$, 4- Generate random value, 5- Insert it in the simulation program.

4.3 Channel Module Implementation

This channel need to support P2P and P2M scenarios. It has been constructed using module class from SystemC library and connected together through ports as shown in Figure (4.3). These ports are instantiated from the base class `sc_port` and are bound to an interface type. Port interfaces employs First In First Out (FIFO) queues as interface type and are used to connect the system modules to each other. The FIFOs can be accessed using blocking read and write methods. So synchronisation is implicit, the methods being automatically suspended and resumed, depending on the status of the FIFO channels. This guarantees that no packets get lost [83].

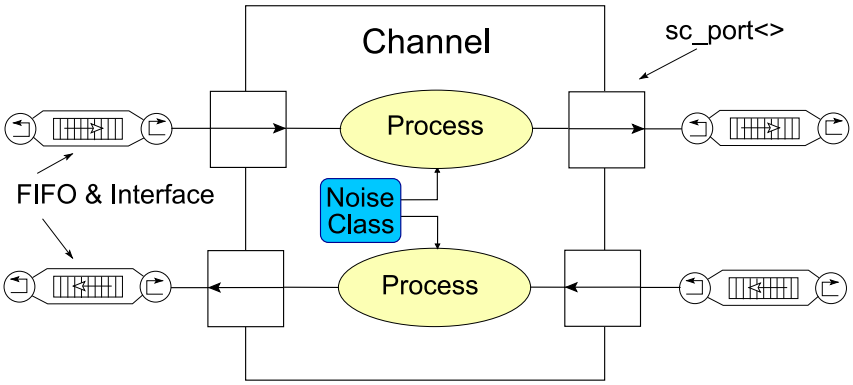

Figure 4.3: Channel Module Implementation

The basic design of P2P consists of the source module, the destination module and a channel module as shown in Figure (4.4). The channel has been implemented as the communication scheme between the source and destination nodes. The source module sends data packets that are created based on the HDLC format through the channel. After that the channel module processes and forwards them directly to the destination. The digital channel introduces errors to represent the actual error rate of the system as mentioned above. Future work will be devoted to an accurate wireless channel modeling.

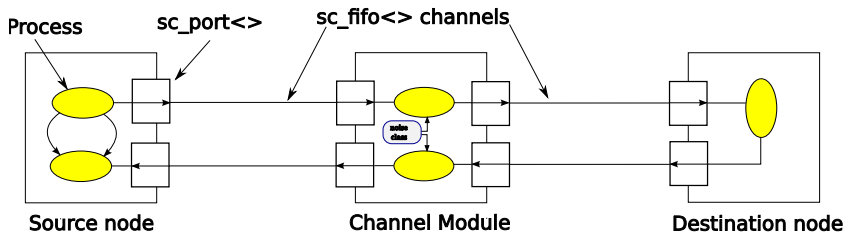

Figure 4.4: Simple point to point communication channel Structure

4.4 Point to Multipoint Construction

The point to multipoint is more complicated compared to P2P, because in this case, I need to define multiport and to set a different noise values and communication parameters to each channel module. Multiports are created by using the

port_array feature of SystemC and by providing second parameter N in the base class `sc_port<interface[,N]>port_name` and then bound to an interface type. I assumed that the source node is connected to N target nodes via N interface channels, therefore the receiver modules are assigned a position in the array to connect the channel modules on a first-come first-serve basis as shown in Figure (4.5). Actually, there is two way of connecting, the first one is known as a shared channel and the other one is multiple channels. In this chapter of the thesis, the later is selected to describe P2M, because it assumes there is no correlation between channels. Moreover, this structure fits well with multiport definition and module structure of SystemC.

The P2M System described in this chapter assumes a schematic of master slave synchronisation mechanism, such that only one slave is allowed to transmit in the feedback channel at given time. For the general case it is required to model the contention, whilst accessing two resource. This is left for further refinement and expansion of this work in the next stage. To create the P2M platform, firstly all modules are created during elaboration time; namely the source module and the N channels and destinations are dynamically instantiated. Secondly, the module interfaces are instantiated. Finally named port binding of modules and interfaces is applied between source module and channel modules, also between channel modules and destination modules.

4.5 Simulation Platform

In SystemC functional verification is done through simulation, applying stimulus to the Device Under Test (DUT) and verifying the response against an expected result as shown in Figure (4.6). In order to test and evaluate the noisy digital channel described in this chpter, a test platform consisting of the P2P design described earlier, packet format based on HDLC has been constructed as described in [35, 96].

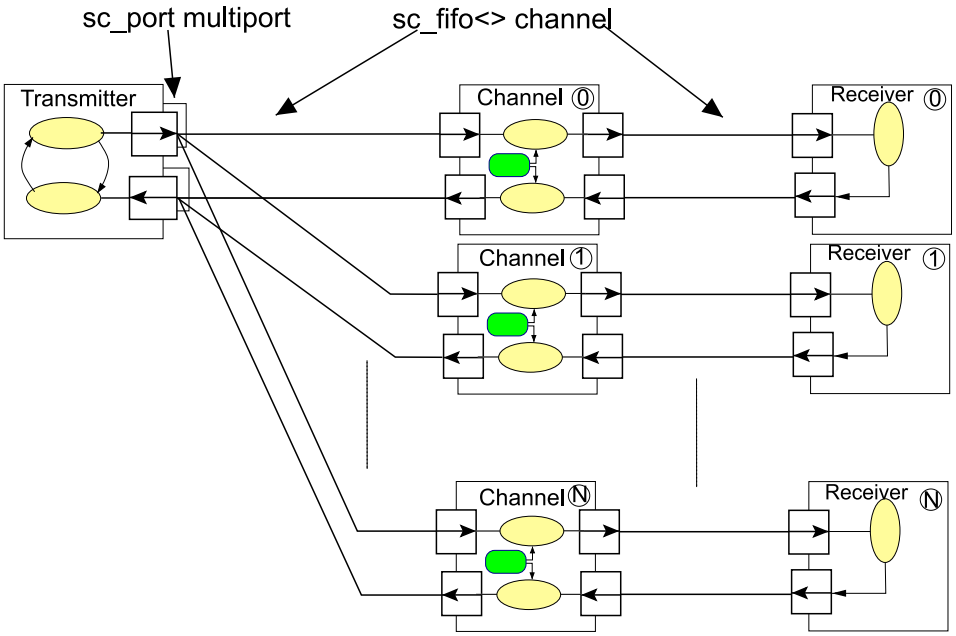

Figure 4.5: Point-to-Multipoint Communication Channel Structure

The stimulus sends bit stream to the source module. After that, the bits can be sent across the noisy digital channel to the destination module. Then the monitor module gets these bits from the destination module in order to test channel module, i.e. it receives the bits with some errors from the channel module and then checked, verified and analysed the bits in order to detect the errors. The modelled system is assumed to be in a clock locked state. So there is no need to model PLL for clock recovery.

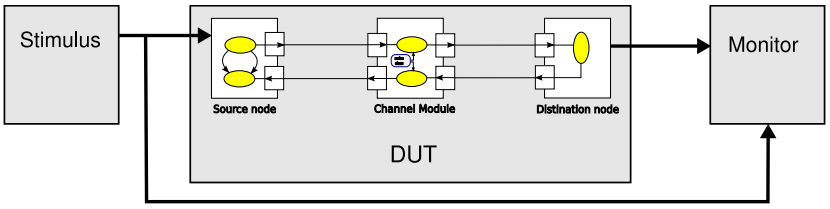

Figure 4.6: Simulation Platform

4.6 Results and Analysis

Throughout the course of developing of the wireless communication link, the model needs to be verified before conducting experiments. Verification is a measure of how accurate the problem or model has been transformed into software. In other words, verification deals with building the model correctly. Since a simulation can process only a finite number of bits or packets, the bit error rate that need to evaluate can only be determined and depends on the number of packets. In this work, BER is determined by passing a large number of packets through the system and count errors at the destination , i.e. packets with errors that observed at monitor module divide total number of packets that sent from the source. So I have set my simulation for running until the BER converge.

In Addition packet delivery ratio has been also choosen as a metric of measure in this work. It represents the ratio between the number of packets originated by the source node and the number of packets received by the destination nodes. Packet delivery ratio is important as it describes the loss rate that will be seen by the transport protocols, which in turn effects the maximum throughput that the network can support. This metric characterizes both the completeness and correctness of the transmission protocol.

In the simulation of P2P, I set BER to 0.01 and therefore the result of replicating the random experiment of passing a large number of packets or bits through the random channel is shown in Figure (4.7). The BER based on any number of transmissions gives a spread of results and it is evaluated when the error occur. This spread is related to the variance of the estimated value in general. In order for simulation results to be useful, the spread should be small and convergence to the given value of BER. Note that, for the results shown, the variance grows smaller as the number of packets injected grows larger. This is typical behaviour for a correctly developed estimator.

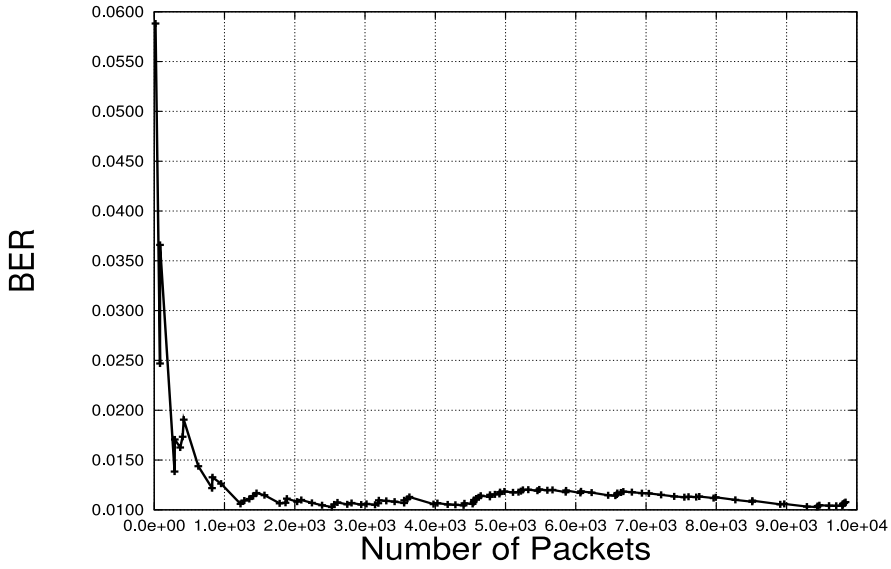

Figure 4.7: Changing in BER versus Number of Packets in P2P Communication Channel

I have seen that, the noisy channel has been simulated correctly. And my platform has verified the correct simulation of the system and the correct injection of the errors in the bit stream.

A sample of the expected packet delivery ratio (Throughput %) or the system transmission efficiency as shown in Figure (4.8) has been determined by the monitor .

For the P2M platform I have set $10 \leq \lambda \leq 10000$ uniformly to ensure that I will get different values of λ, also each channel to process each data packet with different noise. In this simulation, I have got $\lambda_1 = 8400$, $\lambda_2 = 7800$ and $\lambda_3 = 9100$. Figure(4.9) shows the P2M, BER, result with three channels and three receivers.

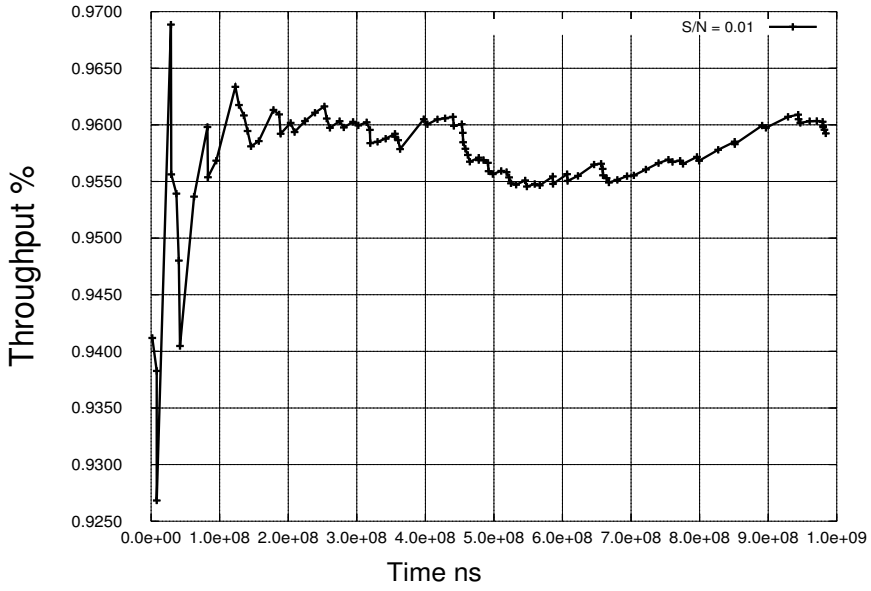

Figure 4.8: The fraction of total data packets successfully delivered packet delivery ratio (Throughput %) as a function of time.

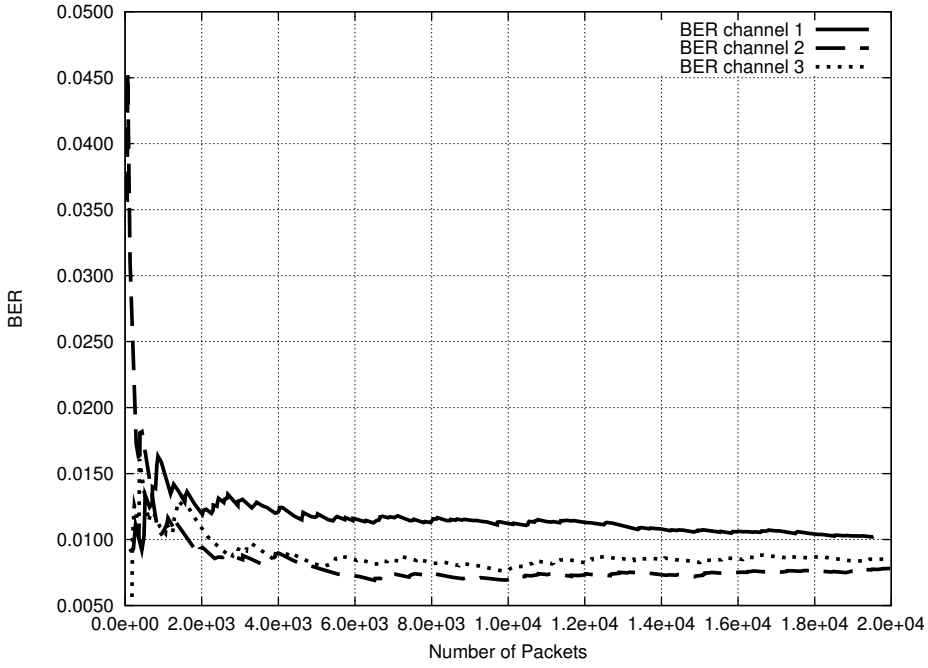

Figure 4.9: *BER* of the P2M Channel Platform for communication system with three nodes

4.7 Conclusion

This chapter has presented a first step towards the integration of communication modelling into the design modeling at the early stages of the system development. The simple noisy digital channel can be used to model the whole communication system interactions. This work demonstrate a simple and computationally efficient way to model a communication channel within a system level. The model is developed at a high level of abstraction which allows for fast simulation and early estimation, which are necessary for successful system development using the SoC design methodology. Furthermore, as systems become more tightly integrated, the ability to evaluate the system performance at early stages of a design becomes increasingly important. This is facilitated by the SystemC design methodology, and by following an IP-based design.

To my knowledge, this is the first time that the modeling of wireless communication system has been undertaken in SystemC and incorporated into a uniform design methodology, suitable for developing new technologies following the SoC design methodology.

5 RTL-Level Modeling of an 8B/10B Encoder-Decoder

This chapter presents an RTL-level model of an 8B/10B encoder/decoder block in SystemC. The use of 8B/10B coding is an important technique in the construction of high performance serial interfaces. These are particularly suitable for alleviating the I/O bottleneck of state of the art systems (which are pinout, rather than bandwidth limited). Moreover, to optimise the use of the transmission medium encoding may be chosen to conserve bandwidth or to minimise errors.

5.1 Introduction

Serial transmission technology is increasingly used for the transmission of digital data. State of the art communication networks make use of serial links for transferring data. This is in part due to the reduction in pinout and cost; but most importantly because it is inherently immune to skew, which plagues high speed parallel interfaces. To improve the performance in serial data transmission systems, block coding is used to ensure sufficient data transitions occur for clock recovery and also to help guard against errors. In the early 80's the 8B/10B block coding technique was introduced by Albert X. Widmer and Peter A. Fransazek of IBM Corporation [97]. Although dated, the technique continues to be employed in state of

the art technologies, such as HyperTransport, IEEE1394b, SATA, DVB, and many others.

This chapter describes the construction of an RTL model of an 8B/10B encoder in SystemC. Although other HDL models have been used, a SystemC model is desirable as it integrates into the SystemC design methodology, which provides a consistent framework for the design and modeling of complex systems at numerous levels of abstraction. This is particularly important for the design of SoC systems, where it is necessary to determine the system performance before a prototype is constructed, in order to evaluate the merit of different implementations.

The majority of published work is centered towards specific 8B/10B implementations, such as in [98] which perform 8B/10B encoding/decoding within Lattice programmable logic devices (PLDs). In [99], Wu et al propose a new peak-to-average power ratio reduction method. They use an 8B/10B code in the time domain of OFDM system to reduce peak-to-average power ratio. DC-balance is ensured because the encoder transmits the same number of ones as zeros. This is important, as providing AC-coupled links eliminate the ground-loop problem. In this novel work, the main objective is to develop a reusable 8B/10B IP core offering flexible interoperability, which can be used to allow efficiently prototyping of serial communication systems. Moreover, the model has been constructed at the RTL level so that it can be efficiently synthesized to hardware, or implemented as a software component if required.

The remainder of this chapter is organized as follows: Section 5.2 provides a brief description of the 8B/10B Encoder-Decoder. Section 5.3 describes the SystemC 8B/10B Encoder-Decoder Model. Software implementation is discussed in sections 5.4, 5.5 and 5.6. Section 5.7 provides results and a brief analysis. Finally, conclusions are drawn in section 5.8.

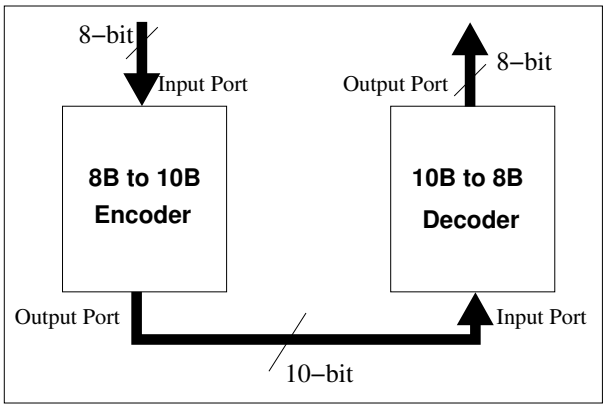

Figure 5.1: 8B/10B Encoder/Decoder Block Diagram

5.2 8B/10B Encoder-Decoder Description

In an 8B/10B encoding process a block of 8 bits of data is converted to a 10-bit block before transmission, with the additional information used to ensure that:

1. Enough data transitions are present.

2. DC balance is achieved

3. To aid in providing data integrity.

The decoder decodes a 10-bit code into 8 bits of data. For ease of reference, the eight input bits are named A, B, C, D, E, F, G, H, where A is the least significant bit (LSB), and bit H is the most significant bit (MSB). They are split into two groups in the encoding process: The five-bit group A,B,C,D,E, and the three-bit group F,G,H. The coded bits are named a, b, c, d, e, i, f, g, h, j (the order is not alphabetical). These bits are also split into two groups in the decoding process: the six-bit group a,b,c,d,e,i, and the four-bit group f,g,h, j.

Figure(5.1) shows a diagram of an 8B/10B Encoder/Decoder block. Since 8 bits of data are converted to 10 bits before transmission, the technique requires transmitting

Table 5.1: 5B/6B Encoding and 3B/4B Encoding

Decimal	Binary	Codeword
0	00000	100111 or 011000
1	00001	011101 or 100010
2	00010	101101 or 010010
3	00011	110001
4	00100	110101 or 001010
5	00101	101001
6	00110	011001
7	00111	111000 or 000111
8	01000	111001 or 000110
9	01001	100101
10	01010	010101
11	01011	110100
12	01100	001101
13	01101	101100
14	01110	011100
15	01111	010111 or 101000
16	10000	011011 or 100100
17	10001	100011
18	10010	010011
19	10011	110010
20	10100	001011
21	10101	101010
22	10110	011010
23	10111	111010 or 000101
24	11000	110011 or 001100
25	11001	100110
26	11010	010110
27	11011	110110 or 001001
28	11100	001110
29	11101	101110 or 010011
30	11110	011110 or 100001
31	11111	101011 or 010100

(a) 5B/6B Encoding

Decimal	Binary	Codeword
0	000	0100 or 1011
1	001	1001
2	010	0101
3	011	0011 or 1100
4	100	0010 or 1101
5	101	1010
6	110	0110
7	111	0001 or 1110

(b) 3B/4B Encoding

encoded data 25% faster than the desired throughput (i.e. 1 Gb/s of encoded data is transmitted at 1.25 Gb/s between terminals [100]). Structurally the 8B/10B code is defined from simpler 5B/6B and 3B/4B codes. For instance, in the encoding side, the 8B/10B Encoder consists of two sub-blocks, the 5B/6B and the 3B/4B encoders, shown in tables 5.2a and 5.2b respectively. In order to aid in clock recovery, the code is designed so that no more than five consecutive 0's or 1's are ever transmitted [97]. The 8B/10B encoder block continuously converts the incoming data to 10-bit symbols. The conversion is done depending upon the value of a signal called *Running Disparity* and the incoming stream of data. The running disparity is a binary parameter, which has either a positive or a negative value, and whose purpose is to ensure DC balance is maintained in the stream. The running disparity of any incoming data is calculated based on the number of logic 1's and 0's present in that data code group. On reset the running disparity value is initialized as negative.

An 8B/10B encoder takes a one byte input, and generates a 10-bit code. Some of these codes are balanced (i.e. they have an equal number of 1s and 0s), while others have a disparity of ± 2 (either four 1s and six 0s, or, six 1s and four 0s). These last codes are always assigned in pairs, such there are always two symbols (with disparities of +2 and -2 respectively) associated to that particular input. The disparity (if any) of the current, and any previous symbols is tracked by the running disparity variable, whose purpose is to maintain an overall balanced stream. This is achieved by selecting the proper encoding symbol so that the running disparity is held at ± 1 (the running disparity is initialized at reset to -1 [97]). The decoder, on the other hand, converts 10-bit symbols to 8-bit data, but does not need to track the running disparity, other than for synchronisation and error correction purposes, as in this case the mapping is surjective. For completeness the decoder implementation described in this chapter does keep track of the disparity.

5.3 8B/10B Encoder-Decoder SystemC Modules Structure

SystemC defines a system modeling and design methodology, which is supported by a C++ class library, that can be used to model systems in a homogeneous environment all the way from requirement capture to system partitioning, cycle accurate modeling and backend implementation. This allows to obtain performance metrics at high levels of abstraction which can be used to asses the impact of different architectural solutions early in the design phase [84, 81].

In this chapter, structural designs for the encoder and decoder models are implemented in SystemC using modules, ports, processes and signals which represent the fundamental constructs of SystemC libraries. The Modules can contain other modules, allowing the hierarchical construction of the system model. Processes communicate to each other via interfaces, channels and ports, and can synchronize with each other via event objects. Also there are a variety of data types are supported to include single bits, bit vectors and fixed-point integers [101, 102]. The encoder and decoder models have been implemented by using modules and can be connected together through the ports which are created from the base class *sc_in<data_type>in_port_name* and *sc_out<data_type> out_port_name*.

5.4 Software Implementation

The overall block diagram of the system is depicted in Figure (5.2). The system is implemented using three main modules: The 8B/10B encoder and decoder modules, which are the devices under test (DUT), the testbench and the stimulus modules. As depicted, the stimulus module has been created as hierarchical construction which instantiates both the encoder and decoder modules. It generates a byte wide data

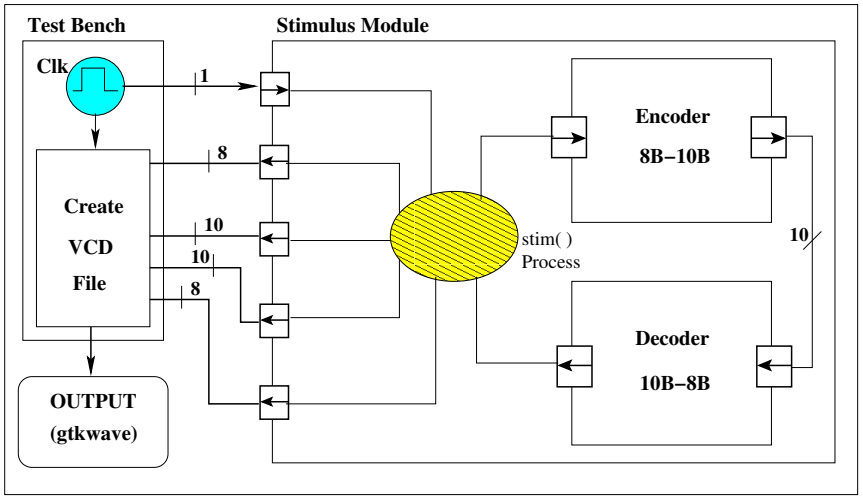

Figure 5.2: Block Diagram illustrates Program Implementation Structure with all Designed Modules

stream of random 1s and 0s then send them to the encoder module input port. After that, the encoder module maps the 8-bit parallel data input to 10-bit output DC balanced stream of 1s and 0s. This 10-bit output is then loaded in and shifted out through its output port which is connected directly to the input port of the decoder module. Next, the decoder module collects the encoded data from its input port and then re-map the 10-bit data back to the original 8-bit data. There are five ports that are created in the stimulus module, one is the input and the other are outputs. All of these ports are used by the test bench which manages the designed modules, captures data for visualisation purposes and verifies the correct operation of the system.

To create the test bench, the main program links-in the various modules and interconnects them. It generates a clock source that need for simulation and apply it to the design and collect output waveforms. Through the four output ports of the stimulus module input data and output data packets (before and after encoding and decoding) of the encoder and decoder have been collected and sent to (Value Dump) VCD file which can be read by GTKWave, a wave viewer program to display output results.

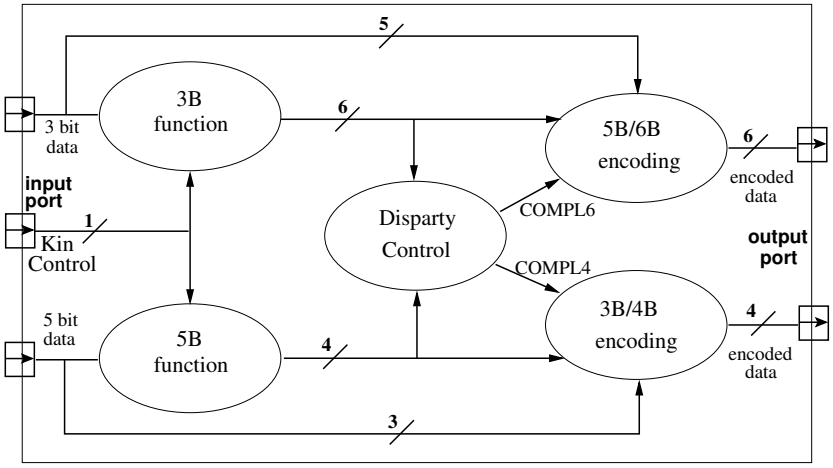

Figure 5.3: 8B/10B Encoder Module

5.5 The 8b/10b Encoder

The 8B/10B encoder is implemented as described in [97]. The first step is to apply the data to the input ports of the encoder module. The input data is represented by an eight bit data line, and a one bit control line, K, which indicates whether the input lines represent data or indicates that the character input should be encoded as one of the 12 allowable control, or K characters (refer to Table 5.2 for description of the core I/O pins). Each incoming byte is partitioned into two sub blocks and applied to a 5B/6B and a 3B/4B encoder respectively. A disparity control block (shown in Figure (5.3) and on the program segment shown below) controls the encoding. . In this implementation, the decoder exhibits a five clock cycle latency. This information is readily available to the designer at the early stages of design, which is an advantage of using the SystemC methodology.

```
SC_MODULE (enc_eight_ten) {
sc_in<sc_logic> RESET;
sc_in<bool> SBYTECLK;
sc_in<sc_logic> KI;
sc_in<sc_logic> AI,BI,CI,DI,EI,FI,GI,HI;
sc_out<sc_logic> AO,BO,CO,DO,EO,IO,FO,GO,HO,JO;
sc_signal<sc_logic> XLRESET, LRESET;
sc_signal<sc_logic> L40 ,L04, L13, L31, L22;
...
```

```
...
sc_signal<sc_logic> NFO, NGO, NHO, NJO, SINT;
// 5b Input Function
void enc_5b(void);
...
...
void ENC3B4B (void);
SC_CTOR(enc_eight_ten) {
SC_THREAD(SYNCRST);
sensitive << XLRESET << SBYTECLK << RESET;
dont_initialize();
...
...
SC_THREAD(ENC3B4B);
sensitive << LRESET << SBYTECLK << COMPLS4;
dont_initialize();
}};
...
...
void enc_eight_ten::enc_5b(void) {
while(true) {
wait(1);// Wait events in sensitivity list
// Four 1's
L40.write(AI & BI & CI & DI); // 1,1,1,1
...
...
L22.write((~((sc_logic)AI) & ~((sc_logic)BI) & CI & DI)
|(~((sc_logic)AI) & BI & CI & ~((sc_logic)DI))|(AI & BI & ~((
   sc_logic)CI) & ~((sc_logic)DI))|(AI & ~((sc_logic)BI) & ~((
   sc_logic)CI) & DI)|(~((sc_logic)AI) & BI & ~((sc_logic)CI) & DI)
    |(AI & ~((sc_logic)BI) & CI & ~((sc_logic)DI)));
wait(2);
}}
```

Table 5.2: Encoder Signals Definition

Name	Type	Description
CLOCK	IN	Clocks all the encoder logic. all input ports have their time referenced to the rising edge of the clock input.
RESET	IN	Global asynchronous reset (active high)
KI	IN	Control (K) input(active high) , is used to indicate that the input is for a special character
HI,GI,FI,EI, DI,CI,BI,AI	IN	Declare unencoded 8-bits Input Data, these 8-bits are named ABCDE_FGH as they represent a 5-bit and a 3-bit sub block of the encoder, where A is the LSB, and H is the MSB.
AO,BO, CO,DO, EO,IO, FO,GO, HO,JO	OUT	Declare Encoded 10-bits output Data, these 10-bits are named ABCDEI_FGHJ as they represent a 6-bit and 4-bit sub blocks, where they arranged from Least significant to Most.

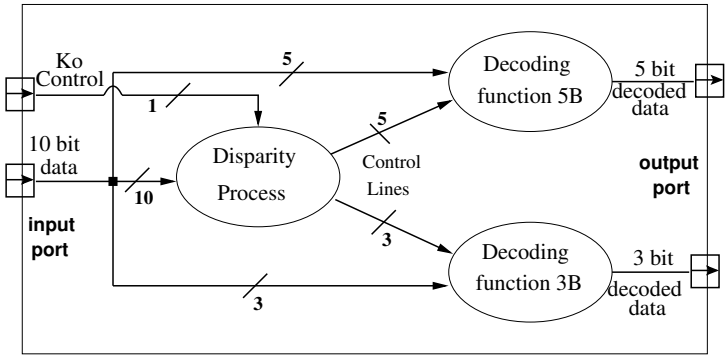

Figure 5.4: 10B/8B Decoder Module

5.6 The 10b/8b Decoder

The 10B/8B decoder converts 10-bit symbols received from the encoder side to 8-bit data. The received code are decoded based on the running disparity process, as previously stated. The block diagram of the decoder module is shown in Figure (5.4). Latency for this modules is the same as for the encoder; i.e. 5 cycle latency, after which the decoded data is passed to the decoder output port and collected by stimulus process.

5.7 Results and Discussion

The system described in 5.2 was implemented in SystemC. The testbench was used to verify the correct operation of the system and to capture data to aid in visualisation as previously described. Figure (5.5) shows a simulation timing diagram for the 8B/10B encoder/decoder operating on random data generated by the stimulus block. For simulation purposes the clock frequency is fixed at 200 MHz. As mentioned in section 5.4, there is a five clock latency for both, the encoding and decoding purposes, which is apparent in the simulation. Furthermore, as the implementation of the stimulus module has not been pipelined, throughput is limited by the ten cycle round-trip latency. This is an accurate representation of bursty communication

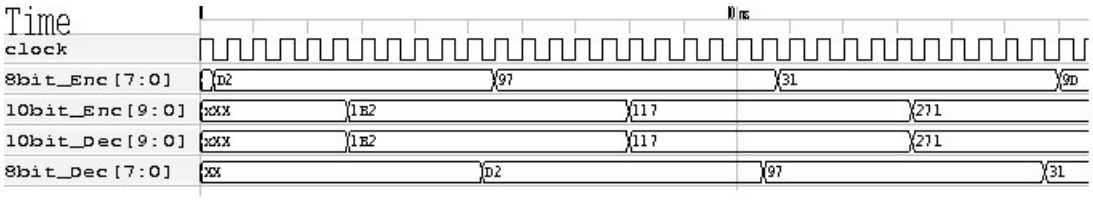

Figure 5.5: Encoder Decoder Simulation Timing Diagram

systems with short packets.

All signals in the designs are latched on the rising edge of the clock. In a second test, the 8B/10B encoder/decoder successfully completed an exhaustive test, to cover all possible input vectors.

5.8 Conclusion

In this chapter, an RTL-level SystemC model of an 8B/10B Encoder/Decoder core has been described. Although other HDL models of 8B/10B decoders have been published, to the best of my knowledge, none target the SoC and IP design methodologies. Although implementation and prototyping of the core is out of the scope of this work, it is important to note that both are relatively simple tasks, as the model developed is already at the RTL-level. The implementation can be carried out by exporting the SystemC code into VHDL or Verilog for synthesis, or directly by using the synthesisable subset of the SystemC language. Once the design has been mapped into a target technology, it is possible to back-annotate timing information directly into the SystemC model, which can aid in providing more accurate timing information for the high level modeling of complete systems.

6 Network Performance Evaluation Based on SoC design Methodology

This chapter presents an original development methodology for the use of SystemC to model an executable model of a wireless network system. SystemC methodology allows the model to be reconfigurable for extensive early architectural analysis and easy re-mapping to new wireless standards and applications at numerous levels of abstraction. On the other hand network protocols have many complex concurrent and distributed characteristics, thus it is very difficult to be analysed theoretically. One of the most promising solutions to this problem is system-level modelling and simulation, which have been covered in this work. Performance results for the simulations as well as development effort are presented thus showing how this methodology is well suited to the modelling of a wireless network systems. Moreover the experiences of using SystemC design methodology to analyse the performance properties of wireless network has been presented.

6.1 Introduction

With the advent of digital communication systems, more and more components to support new functionalities is being integrated in a single package [11, 83, 84, 85]. This trend is likely to continue, as devices continue to incorporate an ever-

growing number of components to provide inter-operability with the large plethora of standards and protocols from previous and the present state of the art. It is an important issue to verify the interaction and the integration of communication modelling into the design modelling before actually realising a system in terms of hardware and software components. Waiting until silicon is available to validate system interactions and find bugs would be a costly and time consuming.

While new design methodologies have emerged for the design of complex systems, in particular for SoC, they have not, to the best of my knowledge, been extended to incorporate, within the same framework, the design of a communication system. This is surprising, considering the fact that the system performance cannot be accurately determined otherwise, and that this performance is used to guide the designer through the architectural exploration phase and arrive at the final implementation. Moreover, the design implementation and IPs are not available till late in the design process. At that stage design issues found through the co-simulations are more difficult to correct and can have a time-to-market delay impacting revenue streams. This chapter will provide a further step towards introducing the modelling of digital wireless communication system in SystemC for structuring the earliest phases of the design process with the intention to find a feasible design before actually realising a system in terms of hardware and software components.

The modelling of wireless communication systems is not new and has been approached in different ways. Co-simulation of analogue parts using Matlab, Simulink and SPICE are common and digital parts in HDL would be too simulation intensive for large and complex systems. Behavioural modelling has been proposed in [95], but has not been incorporated into a homogeneous design environment. In [103], SystemC was used to model the lowest layers of the Bluetooth protocol stack and to reproduce the behaviour of a simple noisy channel. The simulations gave indications to the designer to reduce power consumption by using specific transmission modes. Paper [95] is one of the few papers so far about modelling using SystemC. The paper shows a systematic approach to modelling and simulating an OFDM trans-

ceiver for wireless Local Area Network (LAN) using SystemC. However, in my work I have presented in chapter 4 that SoC methodology using SystemC is very useful for modelling wireless communication systems at different levels of abstraction.

Network protocols for reliable transmission of data over wireless communication channels have been intensively investigated in the field of computer science. They usually involve a subtle interaction of a number of distributed components and have a high degree of parallelism, so it is very difficult to analyse their performance characteristics by mathematical analysis [104]. The most promising solutions to this problem are the use of system-level modelling and simulation.

I modelled a wireless network system which supports a multipoint to multipoint Data link control protocol which encompasses the well known Stop-And-Wait-ARQ protocols for reliable transmission in a multi node communication scenario. Many popular network protocols such as Transmission Control Protocol (TCP) and HDLC are based on the ARQ. A number of key-parameters like the number of nodes, timeout period and packet size have a major effect on the overall performance of the ARQ protocol, and consequently the overall network system performance. How to determine the key-parameters before the implementation of the protocols is important, while designing novel protocols with the ARQ protocol for reliable data transmission.

The novelty of this work is the modelling of a complete digital wireless networking system. All protocols needed for the performance analysis have been developed and integrated in the system for reliable data transmission at the system level of the design phases.The modelling methodology at the system level allows for early estimation and analysis of system performance at the early stages of the design. My methodology is based on the popular SystemC design methodology, which allows, rapid prototyping of these systems for early implementation-dependent architectural analysis, fast simulation and an excellent path to implementation with potential reuse for RTL and schematic verification. In this chapter a digital wireless network

system such the one shown in Figure (6.1), which represents multiple communication nodes that exchange information through a shared communication channel has been developed and the performance properties are investigated for a number of configurations of the system parameters. The channel model presented in chapter 4 has been developed and carefully integrated in this network model and verified with the same methodology.

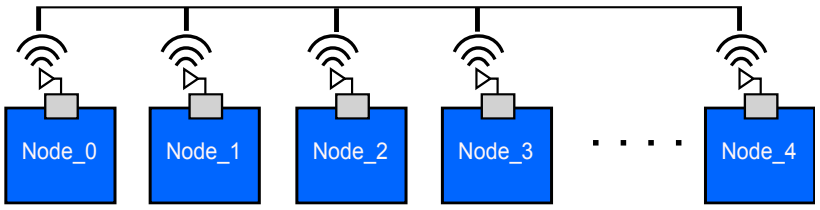

Figure 6.1: Block Diagram of shared channel wireless network system consists of N nodes

The remainder of this chapter is organised as follows. Section 6.2 highlights the communication node structure and a description of the different network layers that are crucial to the implementation of the whole wireless network system. Section 6.3 introduces the CSMA development scenario. Section 6.4 describes the implementation of the noisy digital communication channel. And also highlights the details of using that communication channel to construct a P2P, P2M and MP2MP communication mechanisms. Section 6.5 emphasises on the techniques used to measure the performance "throughput and latency" of the system with a presentation of the simulation results. Finally the conclusions are presented.

6.2 Modelling of Communication Node

At the beginning of the modelling phase a concept for the design was created. The communication nodes are the main construction units of the network system, which will communicate with each other through the communication medium modelled in section (4.2). The nodes of the network are modelled as modules with different meth-

ods, which will reflect the Application, Transport and Data Link Control (DLC) and Physical layers (PHY) of the standard reference model of an Open Systems Interconnection (OSI). The OSI model was developed by the International Organisation for Standardisation (ISO) as a model for a computer communications architecture and as a framework for developing protocol standards. Each node interacts with the network to send or receive data packets. Each node is assumed to have a baseband transmitter and receiver for wireless communication with other network nodes as shown in Figure (6.2).

The advantages of a layered architecture is that the design and implementation of each layer is simplified and can be done independent of the implementation of the other layers, as long as the interfaces with the layer above and below it are standardized. This freedom in implementation allows many of these layers to be reused in many different systems. Moreover, it is well suited for the modelling methodology used in this work. Since different abstraction levels can be applied. Also, changes in the implementation of one layer are transparent to the other layers.

Most of OSI/ISO network layers have been modelled in the communication node, because it is a wireless network and wireless is subject to errors. These errors will force retransmissions under certain conditions. I want to investigate these error conditions because errors and retransmissions will add latency to the system. In addition I need to have absolute certainty that the data sent is equal to the data received. The point to model it here is to find a settle interaction between the network layers namely transport and physical layer and the network topology. So probably certain changes in the network topology and/or other network layers will imply that the transport layer which I was using does not work as it expected for the wireless. Therefore I need to find out these complex interactions, which does not mean it is always present in wired networks.

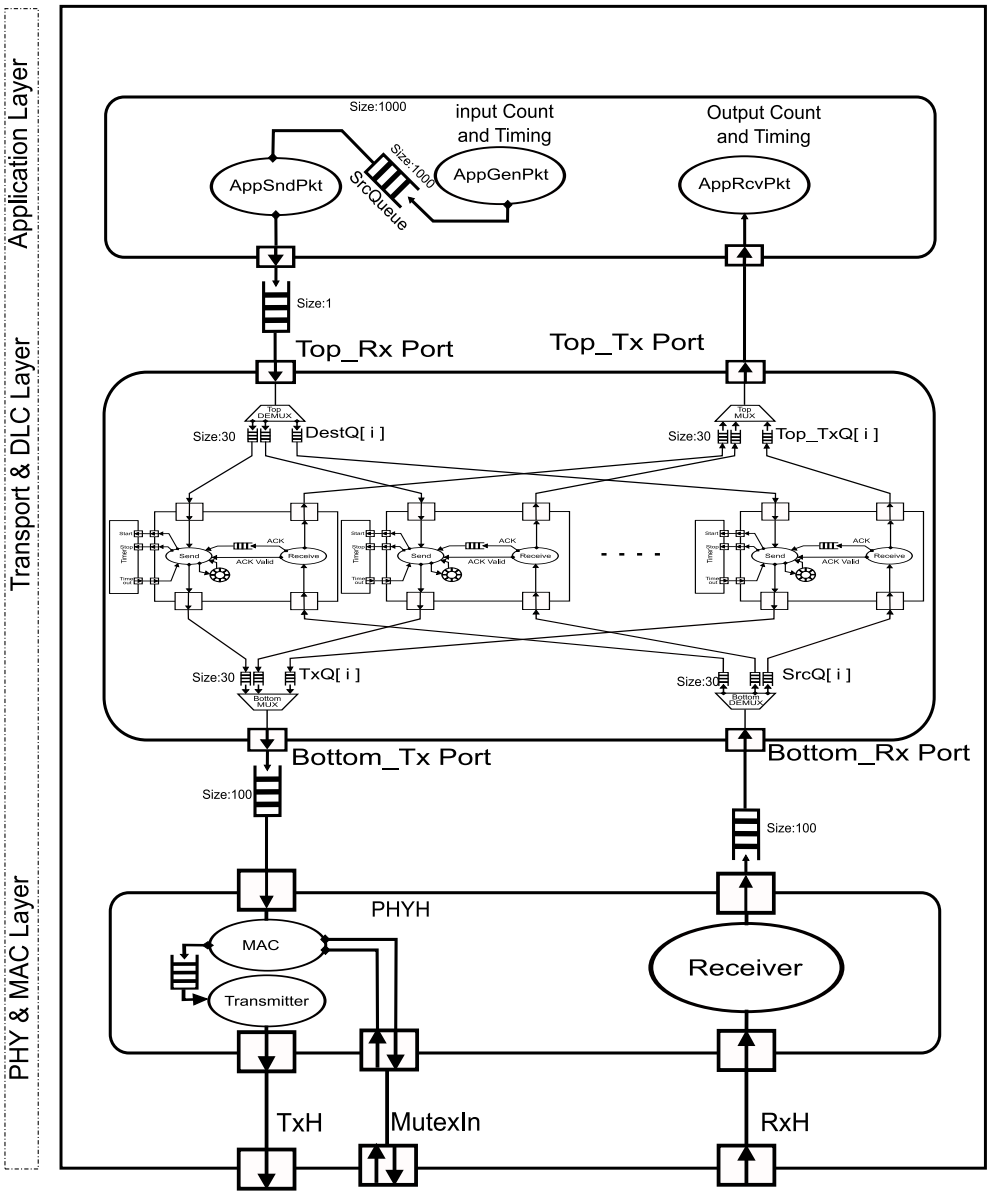

Figure 6.2: Conceptual model of the Network communication node

6.2.1 Application layer

The application layer plays a crucial role on the performance measurement of the network system. To measure the performance of the network modelled, I have included a kind of instrumentation in the application layer. The instrumentation includes a packet source process that generates packets according to a exponential

distribution. And assigns packets destinations randomly and uniformly in the network unless specified otherwise. Since most of the performance evaluation results in literature [105, 26] have only considered a uniform distribution of packet destinations. In this distribution, the probability of node i sending a packet to node j is the same for all i and j, $i \neq j$. The case of nodes sending messages to themselves is excluded because I interested in packet transfers that use the network. In this work for simplicity and to match published results I modeld the network under the exponential traffic distribution. Yet, my methodology allows for any traffic distribution to be simulated; what is even more important, it allows for higher level applications to interact and then dynamically modify the traffic pattern used; which is something that is normally not simulated but has a very large impact on the real performance obtained in the network.

Packets generated at the application layer are equally-likely to occur at any instant of time. This packet source has been separated from the network at each node with a large source queue. Between the packet source and the source queue, packets are counted and time stamped with the start time of each packet injected. It is important that this measurement process be placed before the source queue rather than after the queue. So that packets that have been generated by the source queue, but not yet injected into the network, are considered. And so that packet latency will includes the time spent in the source queue. In addition, without the source queues, a packet source may attempt to inject a packet at a time when the network node is unable to accept any packet. In such a case, the traffic produced by the source is influenced by the network and is not the traffic pattern originally specified. Because my goal is generally to evaluate the network on a specific traffic pattern.

After the source queue another interfacing process has been implemented for sending data packets over the wireless channel through the underlying layers of the communication node. Whenever the application layer has data to be sent, it delivers it to the transport layer buffer as a data structure containing the payload or data, source and destination address.

A complementary measurement process at each receiving side of the application layer records each packet's finish time. Throughput is measured by counting the packets arriving at each receiving application process and latency is measured by subtracting the start time and the finish time for each packet. This measurement configuration enables the traffic parameters to be controlled independently of the network itself.

6.2.2 Transport Layer

A Transport layer provides the logical communication between application process running on different nodes. Hence it is responsible for packet scheduling from multiple processes in different nodes. This process-to-process packet delivery is simply called transport layer multiplexing and demultiplexing. A transport layer protocol with multipoint-to-multipoint support presented in Figure (6.3) has been developed to allow significant performance gain in the system

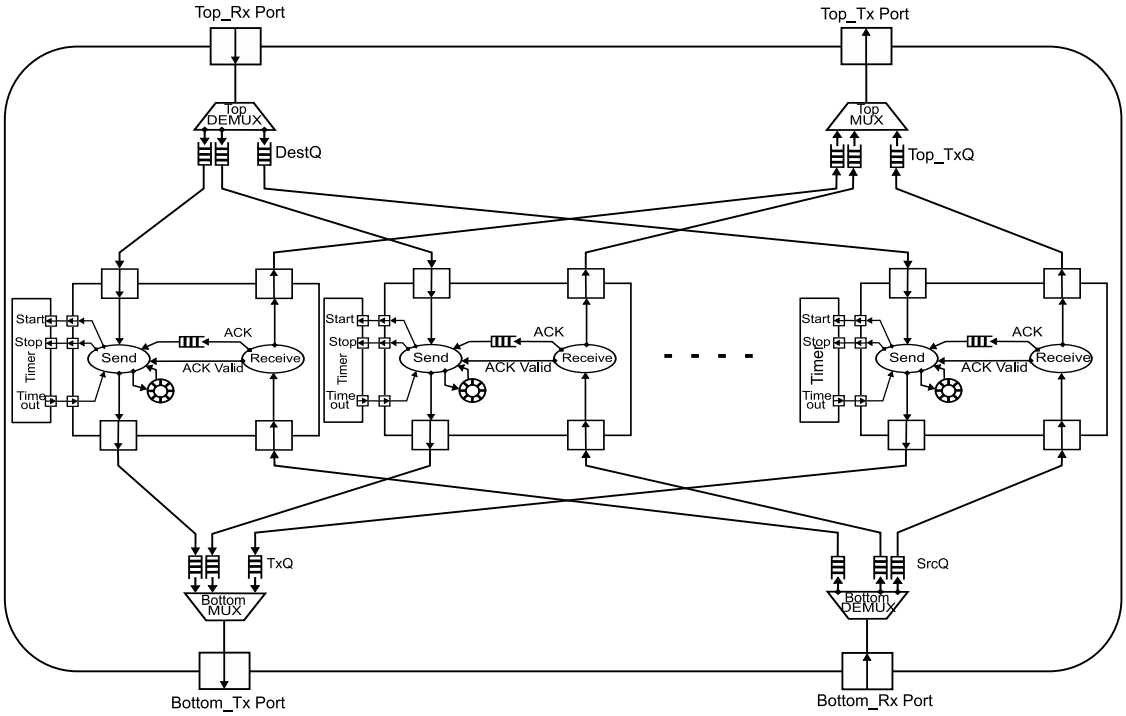

Figure 6.3: Transport Layer Logic diagram

6.2.2.1 Multiplexing and Demultiplexing

Transport layer multiplexing and demultiplexing is essential to a provide delivery service to a process-to-process for multi-node communication scenarios. It provides the logical communication links between processes running on different nodes.

1. **Packet De-Multiplexing**: At the transmission node the transport layer receives packets from the application layer addressed to different destinations. It has the responsibility of directing the data to the appropriate DLC process running in the node and then the DLC will ensure a reliable transmission to the destination node. At the receiving end, the transport layer examines the packet arriving from the PHY layer, identify the receiving DLCs from the arriving packet and then directing the packet to it.

2. **Packet Multiplexing**: At the transmitter node the transport layer gathers the data packets from different DLCs and passes the packets to the PHY layer. At the receiving side the packets are gathered from the different DLCs and forward it to the application layer.

6.2.3 Data Link Control Layer

The DLC layer is responsible for moving a data packet reliably from one node to an adjacent node over a single communication link. In my model it defines the format of the data packets exchange between the communication nodes using the HDLC protocol. HDLC protocol defines the action need when sending or receiving a packet from any source to any destination node. Services provided by this layer includes data packetization, Cyclic Redundancy Check (CRC) error detection, reliable delivery and flow control. For the CRC algorithm, I created a simple C++ class to calculate CRC-16 checksums. It has functions for calculating using a straight-

forward slow method or for using an optimized lookup table. Most of the theory for the C++ class code is taken from the well-known painless guide to crc error detection algorithms article written by [106] and the description provided by [96].

6.2.4 Data Packetization

The data link control layer protocol encapsulates each application layer data packet within a standard HDLC packets format as shown in Figure (6.4) before transmitting it over the communication medium. The HDLC packet that has been constructed consists of the fields shown in Figure (6.4).

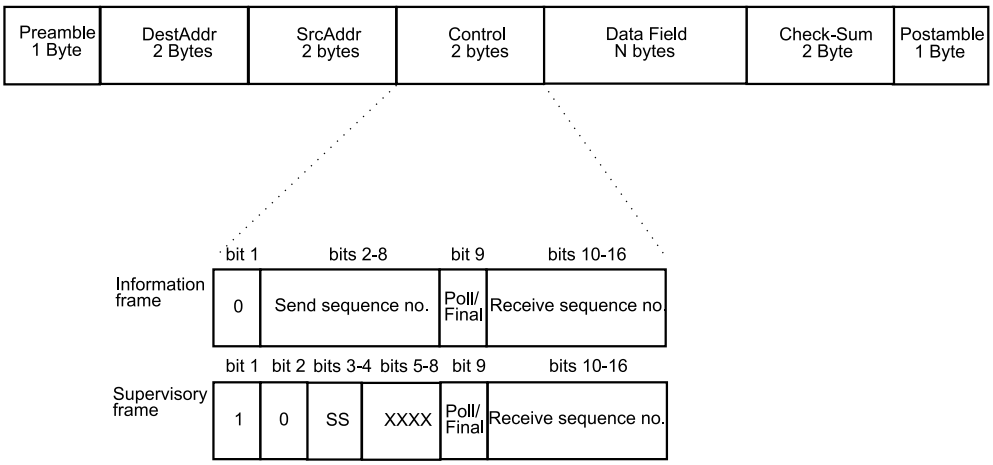

Figure 6.4: Packet format

Within HDLC as described in [107, 96] there are three types of stations defined by the HDLC Primary, Secondary or Combined Station. In this work, I have used Asynchronous Balanced Mode (ABM) where either combined stations may initiate a transfer without waiting for a poll from the other station. Since any node can start transmission at any instant of time.

6.2.5 Flow and Error Control

As described in [35, 96, 108] A DLC sublayer has been designed as shown in Figure (6.5) and the layer logic has been further detailed in a Finite State Machine (FSM) definition for reliable data transfer protocol for transmitters and receivers, this can be modelled as shown in Figure (6.6) and Figure (6.7).

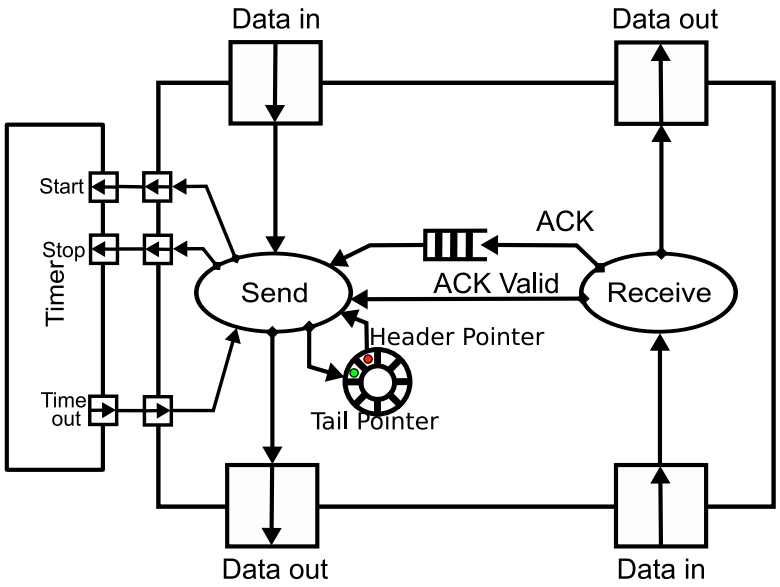

Figure 6.5: DLC sublayer a SystemC representation logic diagram

The Go-Back-N (LAN)protocol has to respond to three types of events:

1. **Data available for transmission**: The sending side of the DLC simply accepts data from the upper layer via its buffer, first it checks if the window is not full, then it creates a DLC packet as described in section 6.2.4 containing the data and headers and sends the packet to the physical layer buffer. After that it has to update the protocol variables accordingly. The DLC keeps a copy of the sent packet for the case of retransmission. But if the window is full the process does not accept the data from the upper layer.

2. **A timeout event**: A Timer has been used to recover from lost data or Acknowledgement (ACK) packets. When the time expires, the sender resends all

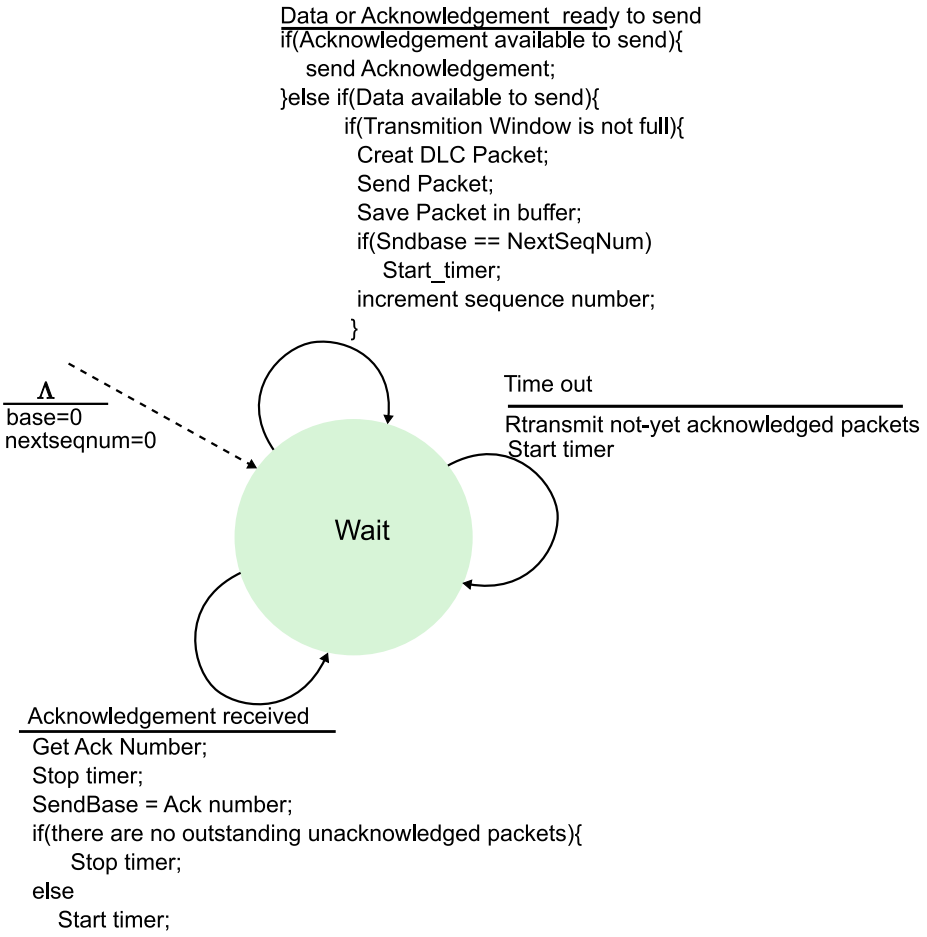

Figure 6.6: FSM description of GBN Transmitter

outstanding packets. And the timer is restarted again. For this protocol only one timer have been used for the oldest transmitted but not yet acknowledged packet.

3. **Acknowledgement event**: All acknowledgements are taken as a cumulative acknowledgement to indicate that all packets with a sequence number up to and including n have been received correctly. And the timer will be restarted again for each correctly received acknowledgement.

On the receiving side the DLC receives a packet from the underlying PHY via its buffer, perform identity and error check using CRC and if a packet with sequence number N(S) is received correctly and is in order, the receiver sends an acknowledgement for packet N(S) and extracts the data and passes it up to the upper layer. If

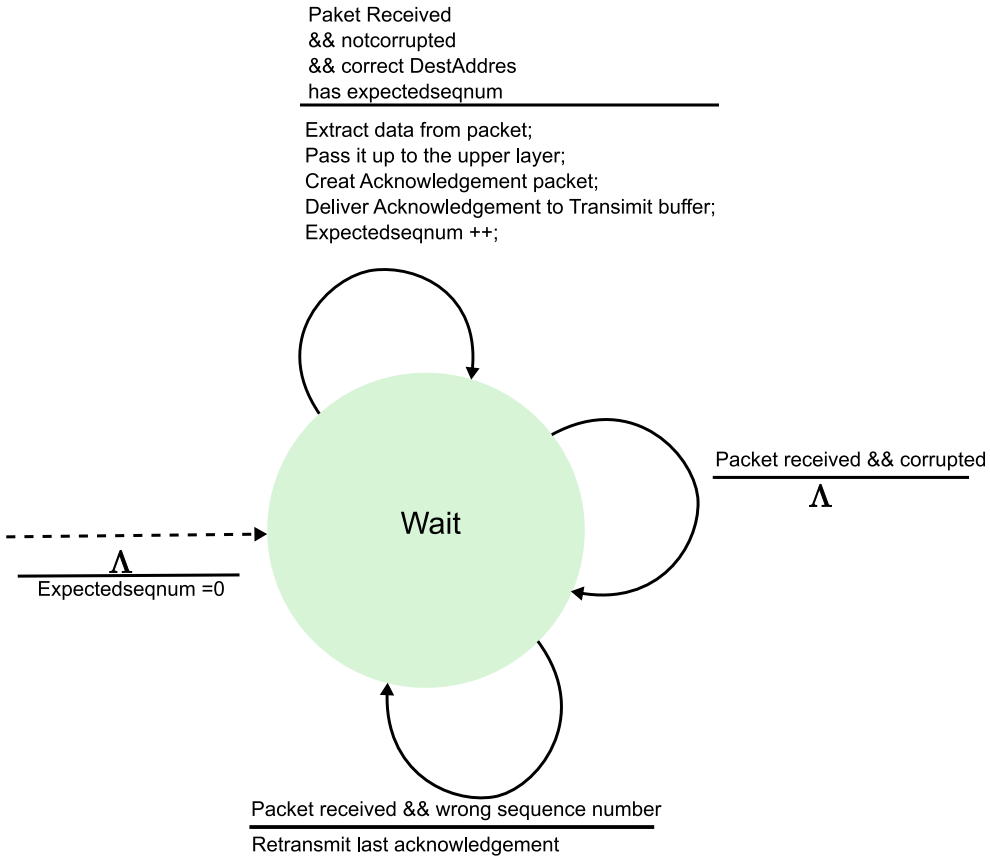

Figure 6.7: FSM description of GBN Receiver

a packet is damaged or received out of order, the receiver remains silent and will discard all subsequent packets until it receives the one it was expecting. The silence of the receiver causes the timer of the unacknowledged packet at the sender site to expire. This forces the transmitter to go back and retransmit all packets in the transmission buffer beginning with the one with the expired timer.

6.2.6 Medium Access Control Layer & Physical Layer

In this work I have integrated the functionality of the MAC into the physical layer to reduce the processing and buffering time between the layers. The MAC and PHY layers provide medium access and transmission functions. The PHY layer covers the physical interface between a data transmission device and a transmission medium or network. This layer is concerned with specifying the characteristics of the

transmission medium, the nature of the signals, the data rate, and related matters.

The shared media used by wireless network, grant exclusive rights for a node to transmit a packet. Wireless medium access control (MAC) dense a set of rules for multiple nodes to effectively and fairly share the radio resources [33]. Access to this media is controlled by the MAC protocol. In designing MAC protocols for my wireless network system, I started by adapting the existing wireless media-access control approaches and then integrated it into my design process. For instance, I will show in section 6.3, how I have developed the CSMA protocol to share the communication medium among all nodes using SystemC design methodology.

The main services provided by this combined layer firstly is to provide the mechanism for accessing the communication link. Secondly deciding which modulation technique will be used to generate symbols to be sent over the communication link. And thirdly perform a byte, phit and/or bit stuffing of the packets before they are transmitted to maintain a link synchronisation. Finally MAC protocol uses a backoff algorithm to avoid collisions when more than one node is requesting access to the channel. Typically, only one of the nodes has access to the channel, while other contending nodes enter a backoff state for some period [109].

6.2.7 Byte stuffing

The PHY transmitter processes the MAC packets and generate PHY symbols based on the modulation technique chosen. In my case for simplicity 256QAM modulation technique has been chosen. Since it allows 8-bit symbols to be generated for simplicity and fast simulation. The packet data and control information appears as a stream of bits, each of which has been divided into phits of 8-bits long.

The packets begins and ends with a unique sequence of bits '01111110' called flags

in HDLC protocol [107, 35, 96]. The receiver starts accepting the packets, when it encounters the first flag in the bit stream until the flag is found again, which indicates the end of the packet. However, it is possible that the data itself may contain the flag sequence causing misidentifying of packet boundaries. To guarantee that will not occurs in the bit stream, a byte stuffing has been deployed. For 8-bits phits/symbols which has been used in this work. Also in this work, symbol is often used interchangeably with phit. The character-oriented version of HDLC has been used, which supports a byte stuffing. The flag is 0x7E (01111110) Control escape 0x7D (01111101) any occurrence of flag or control escape inside of frame is replaced with 0x7D followed by original octet XOR-ed with 0x20 (00100000).

6.2.8 Exponential Backoff Algorithm

Beckoff is a well known method to resolve contention between nodes willing to access communication medium [35, 96, 108]. The Backoff mechanism is a basic part of a MAC protocol. Since only one transmitting node uses the channel at any given time, the MAC protocol must suspend other nodes while the media is busy. The exponential backoff is the backoff mechanisms that wireless networks have adopted from Ethernet. Similar to Ethernet, wireless networks use a shared media. In the exponential backoff method , the station waits an amount of time between 0 and 2^N + (maximum propagation time) before it access the medium, always checking whether a different node has accessed the medium before. In other words, it waits between 0 and 2 + (maximum propagation time) for the first time, between 2^2 + (maximum propagation time) for the second time, and so on.

6.3 The CSMA development scenario

This section deals with the modelling of the nonpersistent CSMA protocol as described in [37, 110] with additional Go-Back-N ARQ flow and error control as described in section 6.2.5, and simple backoff mechanism as described in [108, 96]. Also the CSMA channel access mechanism has been used without Request-to-Send/Clear-to-Send (RTS/CTS) option.

Because CSMA in wireless does not use collision detection, once a station begins to transmit a packet, it transmits the packet in its entirety; that is, once a station gets started, there is no turning back. So the communication scenario of CSMA, is to avoid collisions whenever possible.

In nonpersistent CSMA present in Figure (6.8), the idea here is to limit the interference among packets by always rescheduling a packet which finds the channel busy upon arrival. More precisely, if two stations sense the medium is busy, they both immediately enter random backoff, hopefully choosing different backoff values. These random values should be different to allow one of the two stations to begin transmitting before the other in the next sense of the medium. The other will hear the medium busy and refrain from transmitting until the first station has completed its transmission. This protocol is very effective when the medium is not heavily loaded, since it allows stations to transmit with minimum delay, but there is always a chance of stations transmitting at the same time, caused by the fact that the stations sensed the medium free and decided to transmit at once.

At the receiver node the PHY layer individual phits are received from the wireless channel module. It de-stuffs and reassembles them into a data packet and deliver it to the upper layer which is the DLC Layer. The first operation at the DLC sublayer is the packet identity check, if the packet indicates the local node address then it continues and performs an error check using CRC check-sum. Once the packet

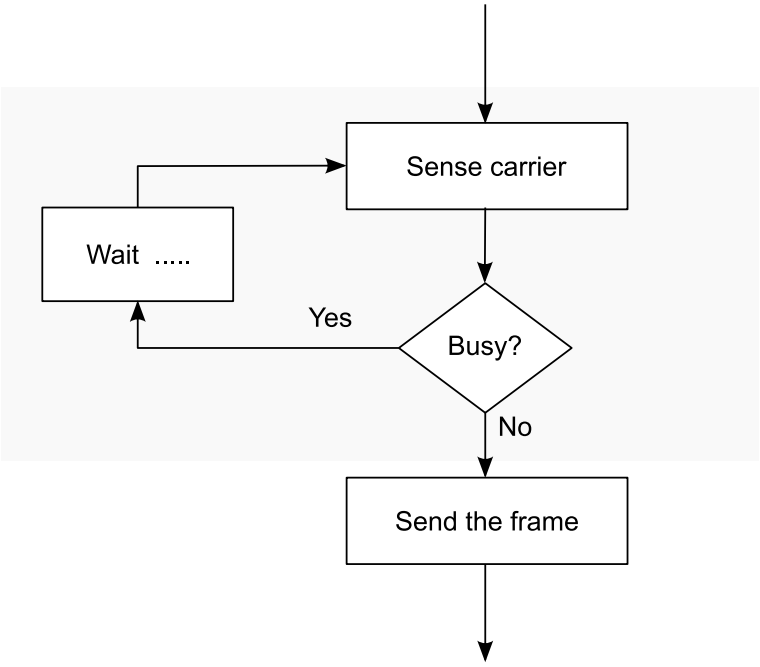

Figure 6.8: Nonpersistent strategy

received is free of errors the DLC generates an acknowledgement packet with the appropriate sequence number and passes it to the transmitting side of the node.

6.4 The Communication medium Module

At the first stage of the implementation phase, a channel module has been constructed to support contention and noncontention based wireless channel access and includes P2MP and MP2MP communication scenarios. The different channel configurations shown in Figure (6.9), Figure (6.10) and Figure (6.11) have been modelled.

The wireless channel module is responsible for carrying the data packets to all stations and it will represent my communication medium in this network. The wireless channel module behaves like a multi-tap bus, where multiple nodes are connected through it. The behaviour of the link has been modelled as a wireless communication medium. In Addition, the module has been implemented in a simple and efficient

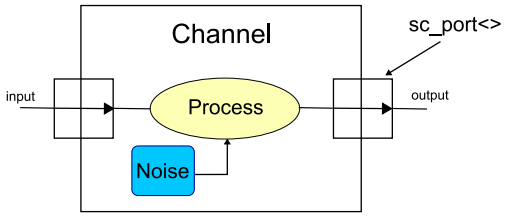

Figure 6.9: Conceptual model of the Point to Point.

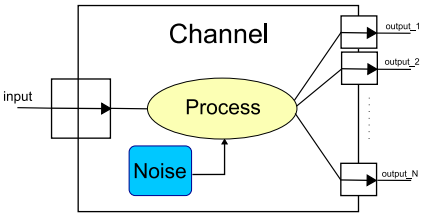

Figure 6.10: Conceptual model of the Point to Multipoint.

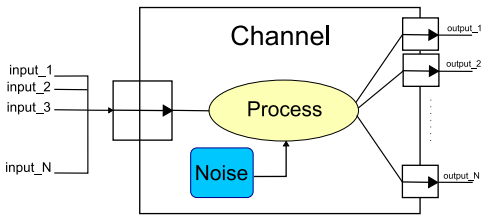

Figure 6.11: Conceptual model of the Multipoint to Multipoint.

way to capture the channel behaviour. At this level of abstraction an external object 'mutual exclusion lock or mutex' has been used to control the access to the shared medium. The behaviour of a mutual exclusion lock as used to control access to a resource shared by concurrent processes. A mutex will be in one of two exclusive states: unlocked or locked. Only one process can lock a given mutex at one time. Whenever a node has a packet to send it tries to lock the mutex. When it locks the mutex it has access to the channel, the rest of the nodes have to wait (exponential backoff time) until the node has unlocked it. The other nodes may be subsequently locked by the mutex .

6.5 Performance Analysis and Simulation Results of Single shared channel

Network performance is measured in terms of latency vs. offered traffic (the elapsed time to cross the network) and throughput (the number of packets that cross the network per unit of time). Performance optimisation of the whole system is a crucial step in the process of design and validation of the new system.

The packet latency is:

$$L = (S_i - A_i) \tag{6.1}$$

and the average latency L_{ave} for n packets for each node is:

$$L_{ave} = \frac{\sum (S_i – A_i)}{n} \tag{6.2}$$

Where; A_i is the packet arrival time of packet i and S_i is the sending time of packet i.

Then the total network latency is simply the average latency for N nodes which is:

$$L_{net} = \frac{\sum L_{ave}}{N} \tag{6.3}$$

Latency is measured in time units. However, I am comparing several design choices, the absolute value is not important. As many comparisons can be performed using

simulations, latency is measured in the SystemC simulator clock cycles. accourding to [2] latency vs. traffic curves give the most accurte view of the ultimate performance of a networked communication system.

The throughput S in $^{\text{phits}}/_{\text{Cycle}}$, the formula that has been used is:

$$S = \left(\frac{\text{Total no.of phits received}}{\text{Simulation time}} \right) \tag{6.4}$$

Throughput is measured in bits per clock cycles.

To provide a basic performance evaluation, it is initially assumed that the network in question will use the following important assumptions:

1. The network is deployed on an area of S square meters. The network nodes are distributed uniformly within a rectangular area (X, Y).

2. The channel width is W bits and the packet length is L_m bits, the packet is broken up into $M = {}^{L_m}/_{W}$ phits, each phit is of W bits and is transferred over the channel in one cycle unit; M is referred to as the packet aspect ratio.

6.5.1 Noiseless Channel Results

Let us first have an ideal communication channel in which no phits are corrupted. This can not be the case in real environment, but they serve as a basis for understanding and developing the protocols of noisy channels. There is no need for flow control and error control in this part. However I have implemented all protocols to facilitate the comparison of the results.

My experiments concentrate on the modelling to imptove the overall performance of the protocol stack of a network rather than a local improvment of each layer. Network performance is measured in terms of latency and throughput. Performance improvement is relative to other standard interconnection networks. However, the quantification will be a relative improvement measure between latency and throughput in the standard single channel interconnection network and other types of interconnection networks. Moreover it investigates the interaction of modelling of the whole network system under the same design methodology. My simulations at this stage have been verified by published results [2, 108, 37]. I focused on the change in the packet size and the change in the number of nodes to study the performance of the network. Two different packet sizes were used; 34 and 54 phits. These packet sizes were chosen to represent a short and a normal size packet, respectively, and have been used in similar studies [5]. Moreover, the average latency of different packets sizes and network sizes were evaluated using the model developed in Section 6.2, and the results are shown in Figure (6.12) and Figure (6.13).

Where the average latency (y-axis) of a packet as a function of offered traffic (x-axis) and the average amount of traffic ($^{Pkt}/_{Cycle}$) generated by each communication node in the network has been shown in Figure (6.12) and Figure (6.13). The simulation results matches those curves presented in references [2, 108]. At lower traffic rates I get a lower bound of the average static latency, which represents that the a packet never contends for network resources with other packets. Note that the latency is tending to increase as the packet size has been increased in the network. This is due to the increase in channel acquisition time due to the CSMA protocol behaviour. This is to be expected, since it is a contention based network system. It is worth noting that single channel (contention based network) network saturate sooner when large packets sizes are used, than for networks with small packet sizes. Since large decision times and large acquisition times are required, for the network developed. It provides a small latency for all operating conditions except when the traffic is close to saturation. Furthrmore, the average latency of different network sizes (64 and 144 nodes) were evaluated and the results are shown in Figure 6.13.

While each node offers a particulare amount of traffic to the network, the throuput, is the rate that traffic is delivered to the destination nodes. At traffic levels less than saturation, the throughput equals the demand and the curve is straight line. Continuing to increase the offered traffic, I eventually reach saturation, the highest level of offered traffic for which throughput equals offered traffic. As offered traffic is increased beyond saturation, the network is not able to deliver packets as fast as they being created.

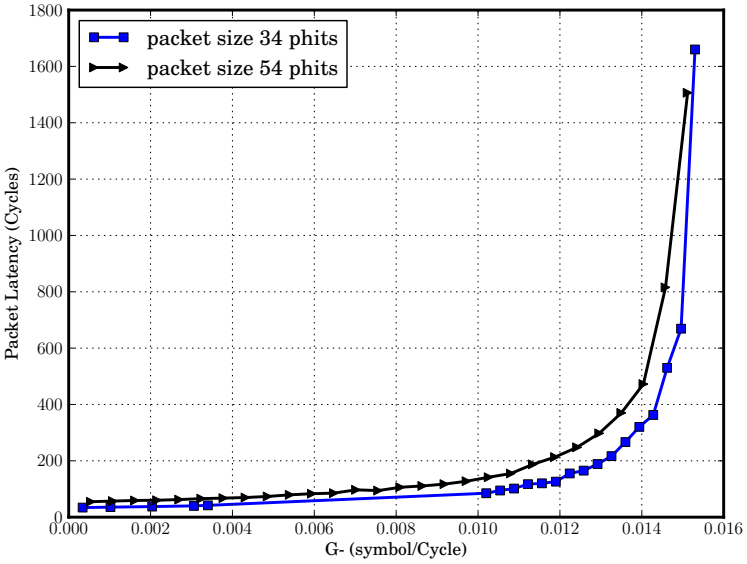

Figure 6.12: Model Latency for different packet sizes shows Latency versus Traffic or packet arrival rate for 64 node model

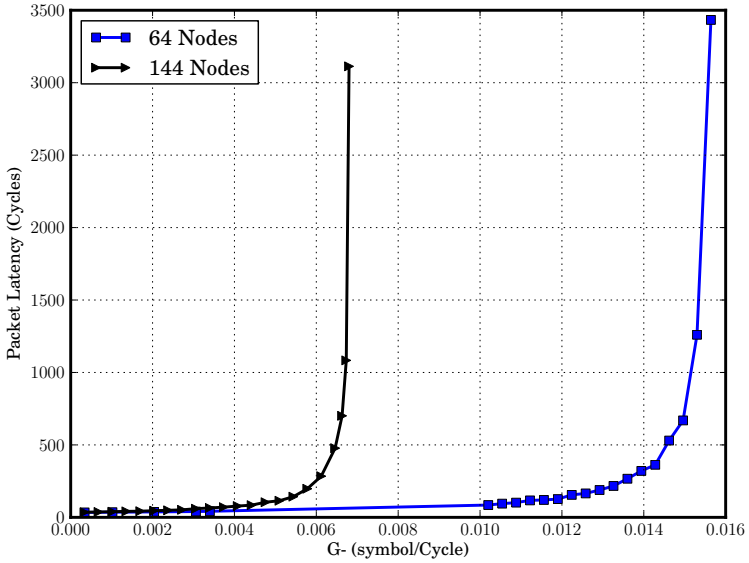

Figure 6.13: Latency for 64 and 144 nodes with a packet size of 34 phits. Latency goes to infinity at saturation throughput

6.5.2 Noisy Channel Results

Although the flow and error control protocols gave us a well developed network that is ready for real environment testing. Since noiseless channels are non-existent in reality. Figure (6.14, 6.15) and Figure (6.16, 6.17) show the results of the Stop-and-Wait-ARQ Protocol and by useing noisy (error-creating) channel. The average latency of different packets sizes and network sizes were evaluated using the model developed in Section 6.2. Similar network behaviour as in noiseless channel system with a relatively higher latency due to the flow control employed.

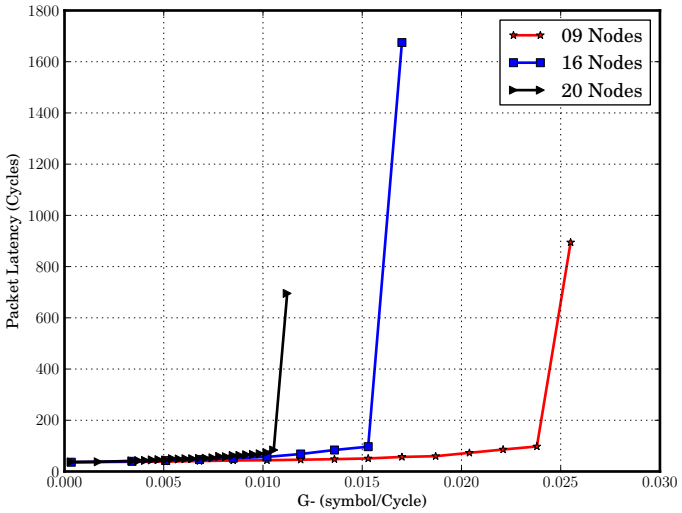

Figure 6.14: Noisy Channel System Latency for 9, 16 and 20 nodes with a packet size of 34 phits

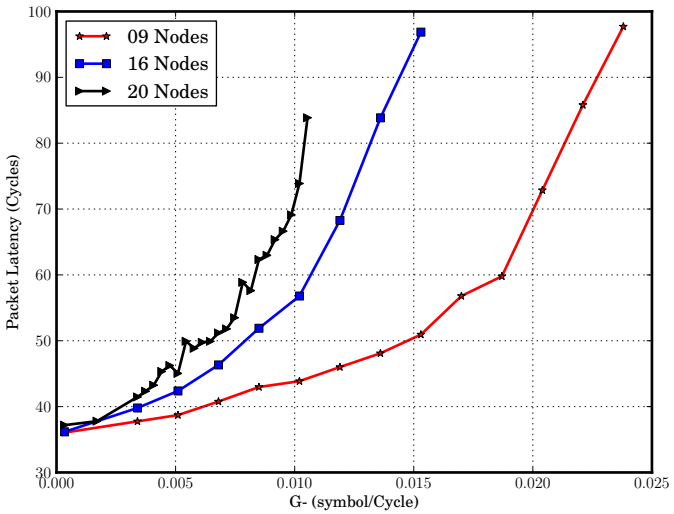

Figure 6.15: Noisy Channel System Latency for 9, 16 and 20 nodes with a packet size of 34 phits

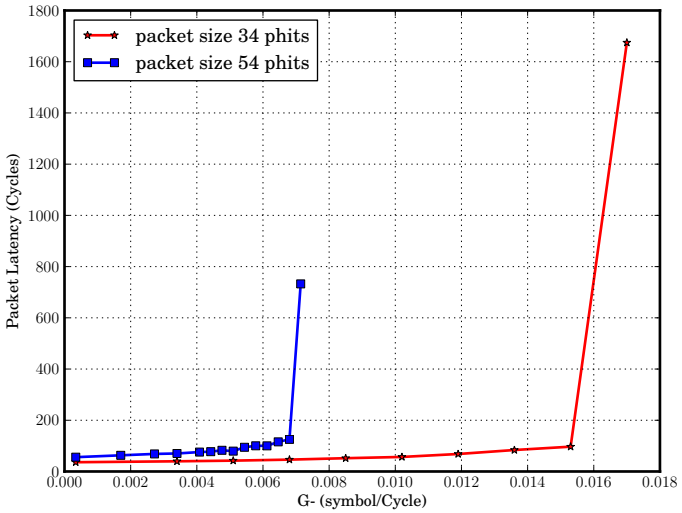

Figure 6.16: Noisy Channel System Latency for different packet sizes shows Latency versus Traffic for 16 node model

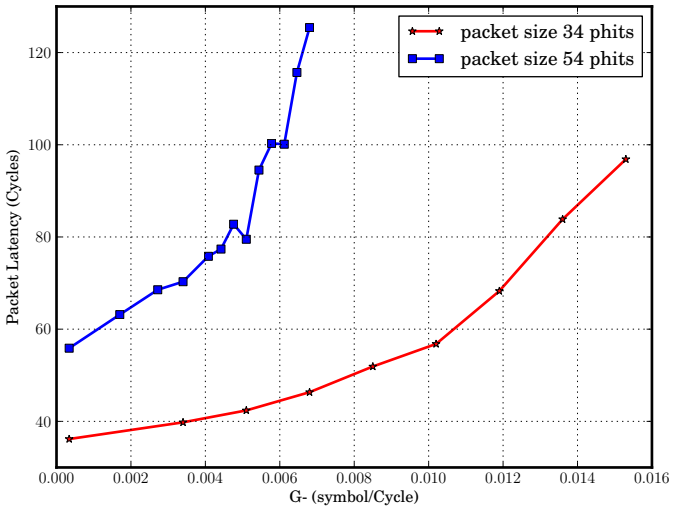

Figure 6.17: Noisy Channel System Latency for different packet sizes shows Latency versus Traffic for 16 node model

6.6 Conclusion

In this chapter I have addressed the modelling of contention based wireless networks using SystemC design methodology. For this purpose, I have developed a conceptual model of the network communication node and the commutation channel and verified our findings by appropriate simulations.

This work demonstrates a computationally efficient way to model a wireless communication network within a system level. The model is developed at a high level of abstraction which allows for fast simulation and early estimation, which are necessary for successful system development using the SystemC design methodology.

To my knowledge, this is the first time that the modelling of wireless communication network has been undertaken in SystemC and incorporated into a uniform design methodology, suitable for developing new technologies following the SystemC design methodology.

7 Single and Multi-Channel Networks: Performance Comparison at System Level

This chapter presents an original development methodology for the use of SystemC to compare two different networks at the system level. A single cluster of a single channel network based on CSMA and multi-channel network cluster based on non-overlapping channels available for each node have been developed. In both network configurations a noiseless (error-free) channels are used, since the focus in this chapter on channel assignment and a comparison of the results against theory to show the accuracy of the developed model. Performance results for the simulations as well as development effort are presented thus showing how this methodology is well suited to the modelling of relatively large networks.

7.1 Introduction

The goal of the topological design of a computer communication network is to achieve a specified performance at a minimal cost [35]. A reasonable approach is to start with a potential network topology and see if it satisfies the connectivity and delay constraints. If not, the starting network topology is subjected to a small modification

yielding a slightly different network, which is now checked to see if it is better. If a better network is found, it is used as the base for more network configurations. If the network resulting from the modification is not better, the original network is modified or reconfigured in some other way.

The traditional ways of network designing mainly depend on experience. However, the ways that simply depend on experience to design networks can not keep up with the development step of new systems. Moreover traditional design methodologies increasingly fail to handle such reasoning in a cost- and time-effective manner, when product time-to-market is the key to success [11, 12, 13].

One of the most promising solutions to this problem are system-level modelling and simulation for instance SystemC methodology which is covered in this work. My methodology reveals that based on the popular SystemC design methodology fast simulation and an excellent path to implementation with potential reuse for different implementation blocks can be easily done. Another important application of SystemC modelling techniques is to perform meaningful comparative studies of different protocols, or new implementations to determine which communication scenario performs better and the ability to modify models to test system sensitivity and tune performance. The drawback is that models of well-known protocols have to be completely rewritten in the system description language together with models of nodes outside the design scope.

The novelty of this chapter is the modelling of a two different networks at the system level. All protocols needed for the performance analysis have been developed and integrated in the system at the system level of the design phases. Also, this chapter will provide a further step towards introducing the modelling of digital wireless communication system in SystemC. Moreover, it will present the use of this methodology for structuring the earliest phases of the design process with the intention to find a feasible design before actually realising a system in terms of hardware and software components.

I confine my comparison study to systems that relatively contain the same protocols with different channel assignment method. This chapter is organised as follows. First I will introduce channel module refinement process. This refinement can be done using one of the distinct capabilities and features of SystemC. I then introduced the analytical model for single shared channel network model to compare it with my simulation results. Also I will present a design and modelling of multi-channel network in SystemC. Different performance results, which used to compare two different networks, are presented. Finally the conclusions are presented.

7.2 Channel Module Refinement

The channel module developed in section 4.3 in Chapter 4, is initially refined in section 6.4 to support contention and noncontention based wireless communication. Moreover, it will support different noise and corrupting methods. The initial results conducted in section 6.5 shows that the packet transfer delay requires additional clock cycles to cross the communication medium. To reduce the number of communication cycles a packet can take to cross the communication medium, the communication medium needs further refinement. This refinement can be done using one of distinct capabilities and features of SystemC. SystemC has custom hierarchical channels mechanism. Hierarchical Channels are intended to model quite complex behaviours such as PCI Express [14], HyperTransport [111], or Advanced Microcontroller Bus Architecture (AMBA) [112]. Hierarchical channels and the interface concept forward a very powerful refinement strategy. It is easy to replace a channel with another one; they only have to share the same interfaces.

Primitive channels, which were used in my initial implementation phase, on the other hand are intended to provide very simple and fast communications [102, 11, 84, 81]. (e.g., `sc_signal` provide a piece of wire behaviour). To build complex system level models, SystemC defines hierarchical channels as modules that implement one or

more interfaces, and serves as a container for communication functionality. An interface provides a declaration of methods for accessing a given channel. No implementations or data are provided in a SystemC interface. In `sc_signal`, for example, the interfaces are defined by two classes `sc_signal_in_if` and `sc_signal_out_if`, and these define methods (e.g., `read()` and `write()`).

Separating the definition of an interface from its methods implementation, SystemC has a unique coding style in which communication is separated from behaviour, a key feature to facilitate refinement from one level of abstraction to another.

Channels in SystemC create connections between module ports allowing modules to communicate. Figure (7.1) shows a SystemC hierarchical channel representation. A port acts as an agent that forwards method calls up to the channel on behalf of the calling module.

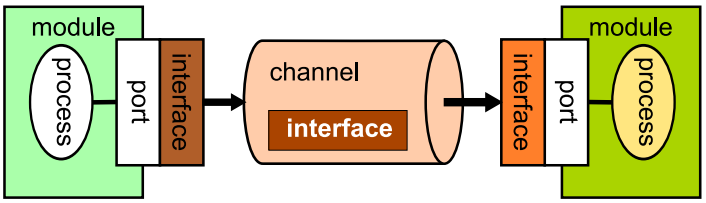

Figure 7.1: Hierarchical channels representation

Hierarchical channels are implemented as modules in SystemC: in fact, they are derived from `sc_module`. Primitive channels have their own base class, `sc_prim_channel`. This section will show only the implementation of the hierarchical channel model used in this work. Figure (7.2) shows the communication medium refinement process done in SystemC.

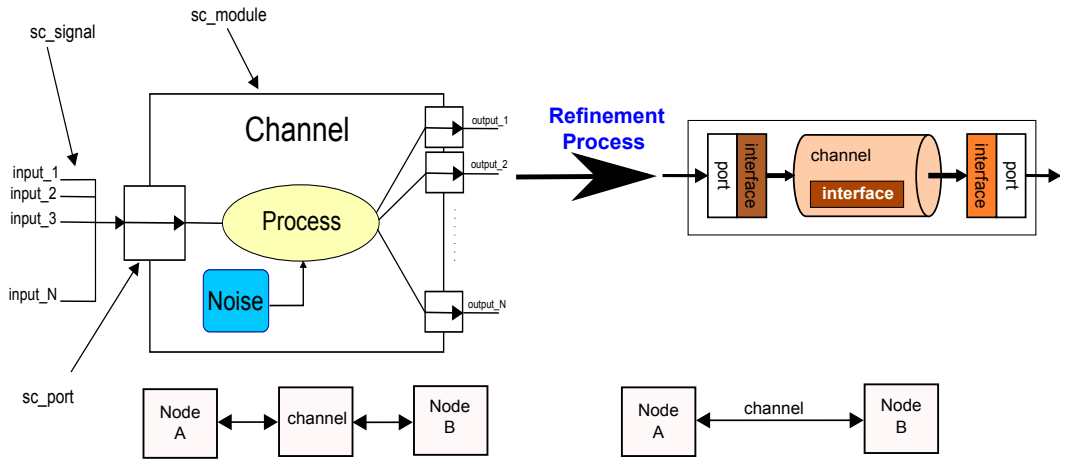

Figure 7.2: Channel refinement process

7.2.1 Define the interface

The wireless signal will provide at the early stages of the development two methods, a read method and a write method. This means it will represent a noiseless channel. A read method returns the value currently carried by the signal immediately. The reading of a signal does not remove the value stored in the signal. A write method write to a signal overwrites the old value without any noise effect. Value is overwritten regardless if the last value has been read or not.

These methods are declared in two separate interfaces, a write interface `wireless_out_if`, and a read interface, `wireless_in_if`. Both the read interface and the write interface derive from interface base class `sc_interface`, which defines a method `register_port()`, which can be used by channels to do static design rule checking when binding ports to channels. And also defines a method `default_event()`, which can be used by channels to return the default event for static sensitivity. Note that all interface methods are pure virtual methods - this means that it is mandatory that they be implemented in any derived class.

Here is the program segment shown in Listing (7.1) containing the declarations,

wireless_if.h.

Listing 7.1: Define the SystemC Hierarchical Interface

```
1  #ifndef WIRELESS_IF_H
2  #define WIRELESS_IF_H
3  #include "systemc.h"
4  template <class T>
5
6  class wireless_out_if :  virtual public sc_interface
7  {
8  public:
9      // write the new value
10     virtual void write(const T&) = 0;
11     virtual const sc_event& default_event() const = 0;
12
13 protected:   wireless_out_if()
14     {
15     };
16 private:
17     wireless_out_if (const wireless_out_if&);   // disable copy
18     wireless_out_if& operator= (const wireless_out_if&);    // disable
19 };
20
21 template <class T>
22 class wireless_in_if :  virtual public sc_interface
23 {
24     ...
25     ...
```

7.2.2 Defining the Wireless Channel

The wireless channel overrides the pure virtual methods declared in the wireless interface. Here is a code snippet of the wireless channel presented in Listing (7.2)

Listing 7.2: SystemC Hierarchical Wireless Channel

```
#ifndef WIRELESS_H
#define WIRELESS_H
#include "wireless_if.h"

// this class implements the virtual functions
// in the interfaces

template <class T>
class sig :
public sc_prim_channel,
public wireless_out_if<T>,
public wireless_in_if<T>
{
    public:
    // constructors
    ...
    ...
    virtual void write(const T& value_) {
        m_new_val = value_;
        request_update();
        }

    virtual const T& read() const {
        return m_cur_val;
        }
    ...
    ...
protected:
    T    m_cur_val;
    T    m_new_val;
    sc_event     m_value_changed_event;

};
#endif
```

There are some local (protected) template data type variables to store the values of the signal, and indicating the current value of the signal.

The `write` and `read` functions are defined here. Also there is a `update()` function which shall be called back by the scheduler during the update phase in response to a call to `request_update`.

Finally is the definition of the `register_port` method. It is defined in `sc_interface` itself, and may be overridden in a channel. Typically, it is used to do checking when ports and interfaces are bound together. For instance, the primitive channel `sc_signal` uses `register_port` to check that a maximum of 1 interface can be connected to the channel read or write ports. In my implementation it merely prints out some information as a binding success.

7.2.3 Creating a Port

To use the channel, it must be instanced. In this work, there are several modules of the same type (the communication node). All nodes have been occupied with a transmitter and a receiver. Here is a code snippet Listing (7.3) of physical layer module, which uses the channel interfaces.

The PHY module declares a port that interfaces to the channel. This is done with the line `sc_port<wireless_out_if<sc_uint<8> > > Tx;`

which declares a port that can be bound to a `wireless_out_if`, and has a name `Tx`.

To actually write to the channel, call the method `write` via the port: `Tx->write(123)`

Listing 7.3: Creating a Port

```
#include "systemc.h"
#include "packet.h"
#include "wireless_if.h"

SC_MODULE(phy) {
    sc_in<bool> clock;

    // output ports
    sc_port<sc_signal_out_if<packet_type> > Top_Tx; // data to upper Layer
    sc_port<wireless_out_if<sc_uint<8> > > Tx;      // to be sent to channel

    // input ports
    sc_port<sc_fifo_in_if<packet_type> > Top_Rx;    // data from upper TL Layer
    sc_port<wireless_in_if < sc_uint<8> > >  Rx;    // data input from
    ...
    ...
    // process
    void Transmitter();                    // Transmitter process
    void SndSymbolProc();                    // send pkt symbol per clock form SndSymbolBuffer
    void Receiver();                         // Receiver process

SC_HAS_PROCESS(phy);
phy(sc_module_name nm, string node_name): sc_module(nm){
    ...
    ...
    SC_THREAD(Transmitter);
        dont_initialize();
        sensitive << clock.pos();
    ...
    ...
}
//Module destractor
~phy(){ }

private:
// definition of local variables
    ...
    ...
};
```

This calls `write(123)` via the `wireless_out_if`. You must use the pointer notation -> when doing this.

The receiver module that reads from the channel looks very similar, except it declares a read port `sc_port<wireless_in_if> Rx;`

and calls `read`, e.g. `Rx->read(val);`

where `val` is of type `unsigned int.`

Note that `read` and `write` both return the value true if they succeed. Perhaps the most interesting thing about this is that the functions `write` and `read` execute in the context of the caller. In other words, they execute as part of the `SC_THREADs` declared in the PHY module.

The implementation shown above is quite simple, and yet there is a lot to add, such as different noise types. The implementation above shows just a starting point through the refinement process of the wireless channel. The key point to note is the communication and behaviour is separated using interfaces; Interfaces may be accessed from outside a module using ports. This technique can be used to have different methods within the same interface. For instance the channel described above has been built for additive white noisy channels using the channel model described in section 4.2. Different noisy channels can also be build at different levels with different methods. For example a method for each channel model, one for additive white noise, other for fading channel and so on.

7.3 Shared Channel Network Model analytical model

I consider the shared channel network model developed in chapter 6 to provide an analytical model of the results obtained from the developed SystemC model. The work in chapter 6 is based on a network model such the one shown in Figure (6.1), which represents multiple communication nodes that exchange information through a shared communication channel. It has been developed and the performance properties are investigated for a number of configurations of the system parameters (e.g., packet sizes, number of nodes). The node structure for this configuration is further shown in figure (7.3).

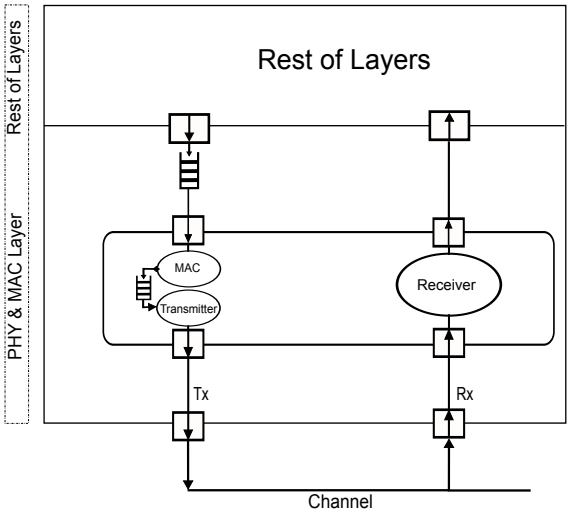

Figure 7.3: Node structure in a one-dimensional single shared channel cluster

For modelling-based approaches to be effective, it is important to understand how well the analytical models are able to capture the performance as seen in simulation model, which will provide a solid model for a future development. With this in mind, in this section I provide some preliminary results on a comparison of analytical model predictions with SystemC model performance results on simple shared communication medium network. In my model I do not consider channel errors and packet retransmissions. However I focused on contention model in single-hop wireless network model.

7.3.1 Assumptions

As described in section (6.4) packet transmissions at the individual nodes are carefully scheduled by an external object (mutual exclusion lock or mutex) to control the access to the shared medium. This has the effect of an ideal situation where no collisions can occur during packets transmission. Moreover, it is assumed that, there is no model of a PLL in the system, and it is assumed to be PLL locked, which may not be true in reality, since the nodes cannot have a central synchronization object. Nevertheless, using SystemC model, I showed at this level of abstraction that the system behavior is predicting the system performance of a contention based protocol (CSMA) that is required at the early stages of the design process to make design decisions.

Based on my systemc model which assumes a perfect time synchronization among the various nodes connected to the shared medium. And also as stated in [38], if the coordination for the transmission instants of the different packets was perfect (i.e., one transmission at once, no collisions: offered traffic equals accepted traffic) the system would have a single queuing model with a mean arrival rate of packets λ and packet transmission time T. However, a real Aloha based system (with collisions and retransmissions without any coordination) is characterized by an M/D/∞ model, since packet transmissions (new arrivals and re-transmissions) are made according to a Poisson process with mean rate λ, the packet transmission requires a deterministic time T, infinite packet transmissions (due to the presence of an ideally infinite number of traffic sources) can be simultaneously made.

Extending the work to include PLL and other features of extensions to CSMA to represent a realistic system, such as collision avoidance mechanisms with a Request-to-Send/Clear-to-Send (RTS/CTS) option is left for future work. Moreover, errors due to contention need to be taken into account in the future to represent a real system.

To see how well the analytical model matches with the simulation results, it is important to set up an analytical model that meets the simulation model's assumptions well. In this model the following assumptions have been made:

- Zero channel status detection time
- The time required to detect the carrier due to packet transmissions is negligible
- Noiseless channel is assumed
- Network topology and propagation delays are as follows:
  - Every node can reach any other node in one hop (no hidden terminal)
  - The propagation delays (small compared to the packet transmission time) are identical for all source-destination pairs
- Each node can be seen as a buffer filled by incoming packets and served by a single server that performs the CSMA multiple access protocol as shown in Figure (7.4).

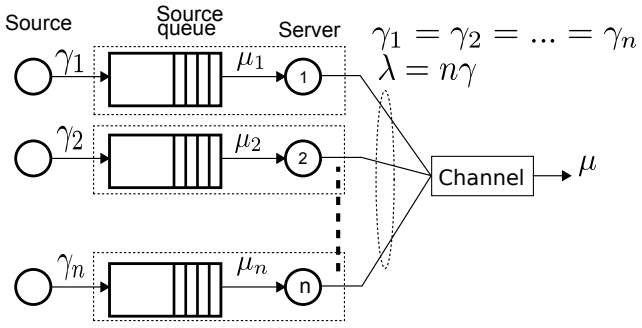

Figure 7.4: Model of the shared communication medium

7.3.2 Traffic Model

- Queues have infinite length.

- Input interarrival times are exponentially distributed.

- Output service time is exponentially distributed.

- There is n number of elemental traffic source as shown in Figure (7.4), therefore queuing is permitted at each of the n nodes.

- Each packet requires a time T to be transmitted. If the packet size is m symbols then the packet transmission time is equal:

$$T = mt \tag{7.1}$$

, where t is the cycle time and the service rate $\mu = {}^{1}/_{T}\ \frac{Pkts}{Cycle}$

- The input traffic rate for all nodes is equal, $\gamma_1 = \gamma_2 = ... = \gamma_n$, and with queuing permitted. The n nodes collectively form an independent Poisson source with an aggregate mean packet generation rate of $\lambda\ {}^{pkts}/_{Cycle}$.

$$\lambda = \sum_{i=1}^{n} \gamma_i = n\gamma \tag{7.2}$$

- $\rho = {}^{\lambda}/_{\mu}$ utilisation or duty factor of the system and it is invalid for $\lambda \geq \mu$ or $\rho \geq 1$

7.3.3 Delay Analysis for a Noiseless Channel

The latency or the packet delay T using results from *M/M/1* queuing theory [113] can be computed. This system has a Poisson input (with an average arrival rate λ) and the average service time is $\bar{T} = 1/\mu$.

The expected number of entries in the queue is given by

$$\bar{N} = \frac{\rho}{1-\rho} = \frac{\lambda}{\mu - \lambda}. \tag{7.3}$$

With variance

$$\sigma_N^2 = \frac{\rho}{(1-\rho)^2} \tag{7.4}$$

Using Little's result, which relates the average number in the system to the average arrival rate and the average time spent in the system, namely $\bar{N} = \lambda T_w$, I can obtain T_w, the expected waiting time as follows:

$$\begin{aligned} T_w &= \frac{\bar{N}}{\lambda} \\ T_w &= \left(\frac{\rho}{1-\rho}\right)\left(\frac{1}{\lambda}\right) \end{aligned}$$

When channels are error free, each nodal delay becomes similar to the system delay

of an *M/M/1* queuing system. So I have

$$T_w = \frac{1/\mu}{1-\rho} \tag{7.5}$$

Using 7.2 and $\rho = \lambda/\mu$, I get the latency for a shared communication medium as:

$$L = \frac{T}{1-n\gamma T} \tag{7.6}$$

7.3.4 Results of shared channel network

The SystemC model has been compared against the M/M/1 queuing model that mimics the behaviour of a shared communication channel as a single queue with multiple inputs at the phit level. The x-axis in the figures represents the rate at which a node injects packets into the network in packets per cycle. The y-axis gives the mean packet latency to cross the network. The statistics gathering was inhibited for the first 3000 packets to avoid distortions due to the initial start-up conditions. The simulation stops when it reaches the accuracy required, which is set to be 0.05 relative half width, i.e., a confidence limit of 95%.

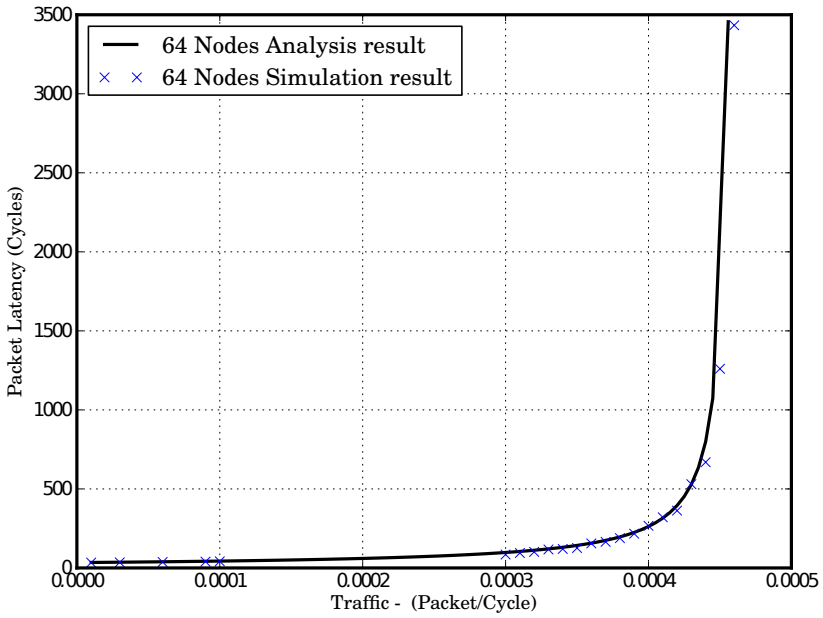

Figure 7.5: Latency vs. throughput of 64 nodes with packet size of 34 phits

Numerous experiments have been performed for several combinations of network sizes, packet lengths to get a good agreement between the models. However, for the sake of specific illustration Figure (7.5) and Figure (7.6) depict latency results for 64 nodes and 144 nodes network model. The figures reveal that in all cases, the analytical model predicts the mean packet latency with a good degree of accuracy in the all network regions, i.e., in the steady state region, heavy traffic region and in the saturation region. In addition, as shown in Figure (7.7) that the network saturates earlier as the number of nodes increase. This is due to the increase in channel acquisition time due to the contention on accessing the medium.

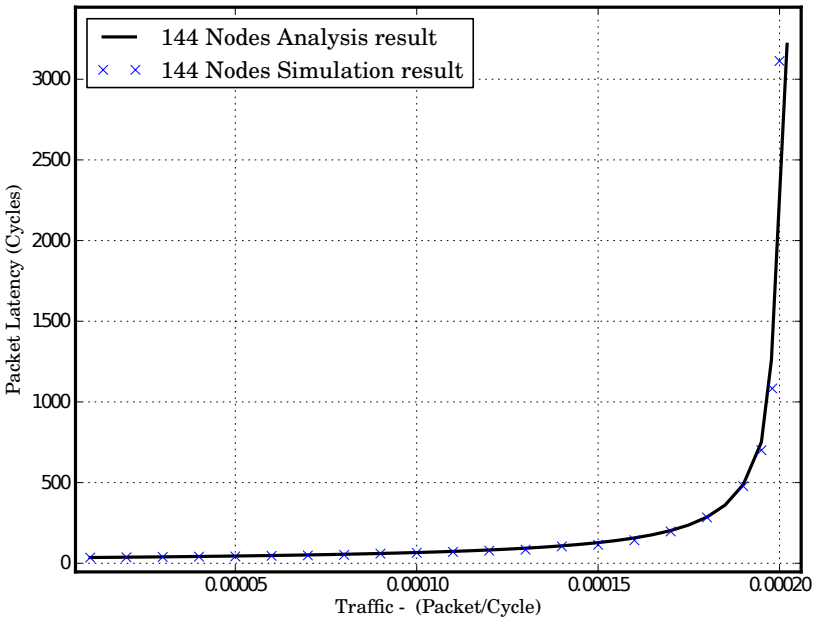

Figure 7.6: Latency vs. throughput of 144 nodes with packet size of 34 phits

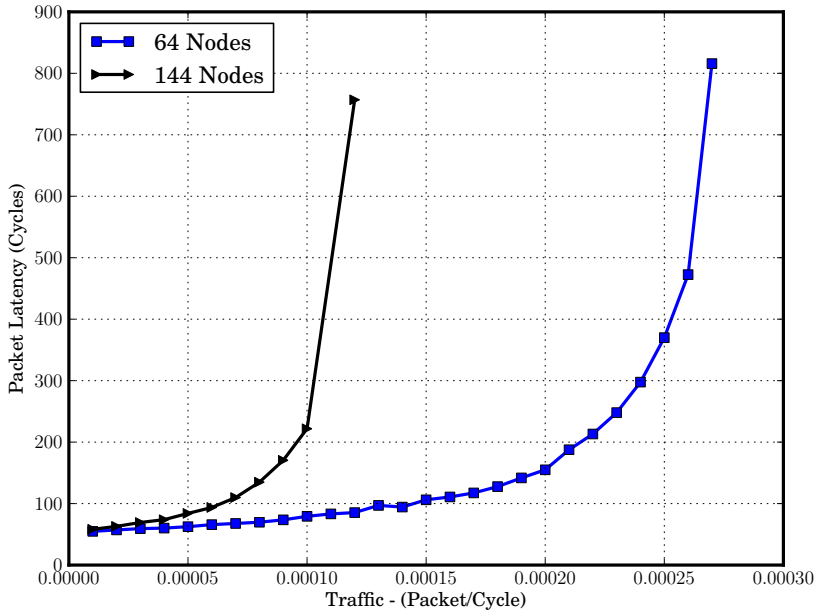

Figure 7.7: Latency vs. throughput for 64 and 144 nodes with a packet size of 54 phits

7.4 One-dimensional Multiple Channels Network

The shared medium is no more a single channel but a group of channels. Full connection of the nodes to all the channels is performed, both for transmission and for reception. The one-dimensional multiple channels network cluster has been proposed by [30, 114] is depicted in Figure (7.8). Their proposed configuration is known as the Distributed Crossbar Switch Hypermesh (DCSH). This configuration is from the hypergraph type of networks. The basic structure consisting of N nodes connected by unidirectional and not shared channels. Every node has its own channel that connects it to the other (N-1) nodes in the cluster. These multiple channels do not suffer the performance penalties associated with contention based single channel structure. In addition, this structure will contribute to maximise throughput and to minimise latency. According to Old-Khaoua et al. [30, 114] their architecture proposal also offers scalability. More network capacity is easily obtained by increasing the number of channels; however the receiver realisation should be modified to admit more channels.

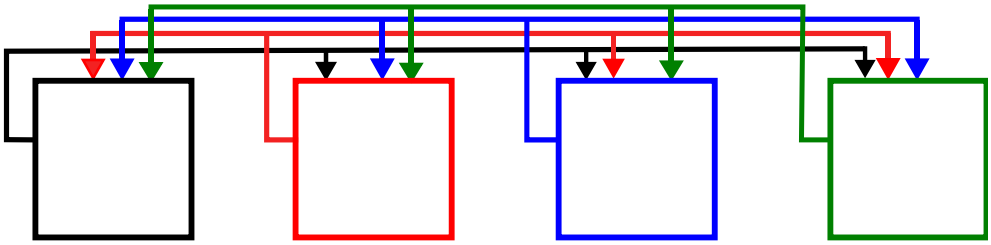

Figure 7.8: One-dimensional multiple channels network cluster

Figure (7.9) shows the node structure in a one-dimensional multiple channels cluster. The node structure is also similar to that of the shared communication channel. Except that there are (k-1) inputs in a one-dimensional multiple channels and one output channel. For simplicity only the difference from the shared channel node structure that has been presented in Figure (7.3), is shown here. Buffers are provided at both inputs and outputs of the PHY layer. There is one *(N-1)-to-1* multiplexer per cluster. An input queue consists of a packet buffer, each of which has enough storage to temporarily hold a few numbers of packets. This multiplexer is to prevent

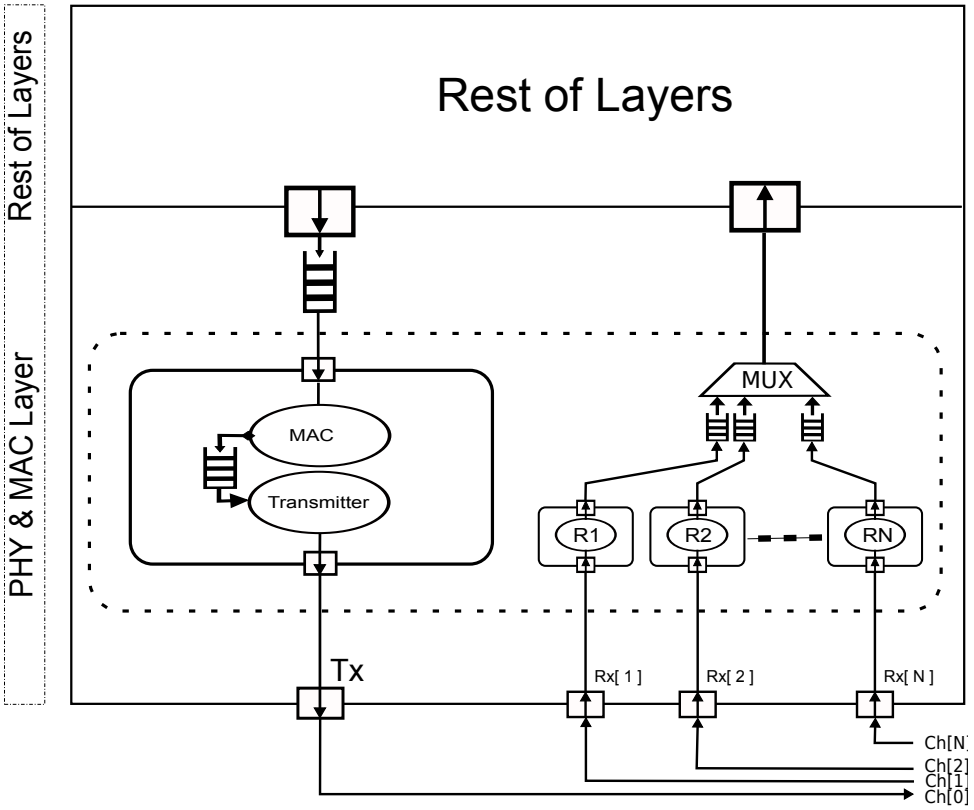

Figure 7.9: Node structure in a one-dimensional multiple channels cluster

a contention, if multiple packets arrive at the same time to the multiplexer and find it busy.

7.4.1 Results of one-dimensional multiple channels network

Figure (7.10) shows the relationship between the offered traffic load and the latency. The horizontal axis in the figures shows the traffic in Packets/Cycls while the vertical axis shows the mean packet latency to cross the network from source to destination. Packet length is 34 symbols or phits. This packet sizes were chosen to represent a short size packet, and have been used in similar studies [5]. The network comparisons take in to account the bisection width of the network. As described in [30, 114, 8] the bisection width across a cluster in DCSH with is givin by

$$B_{DCSH} = NW_{DCSH} \tag{7.7}$$

To compare the results of the two networks the bisection width is held constant for both networks, the channel width of the first and second DCSH networks are given by

$$B_{DCSH1} = N_1 W_{DCSH1} \tag{7.8}$$

$$B_{DCSH2} = N_2 W_{DCSH2} \tag{7.9}$$

If the bisection width is held constant for both networks then we have

$$W_{DCSH2} = \frac{N_1}{N_2} W_{DCSH1} \tag{7.10}$$

The figure reveals that in all cases for small and large networks the network saturates at different points. Because the network under test has different network sizes. Moreover, the figure shows that as the network size increase the network latency is becoming higher at low traffic rates. This is due to the small channel widths as the network become large. This confirmes the results given by [5].

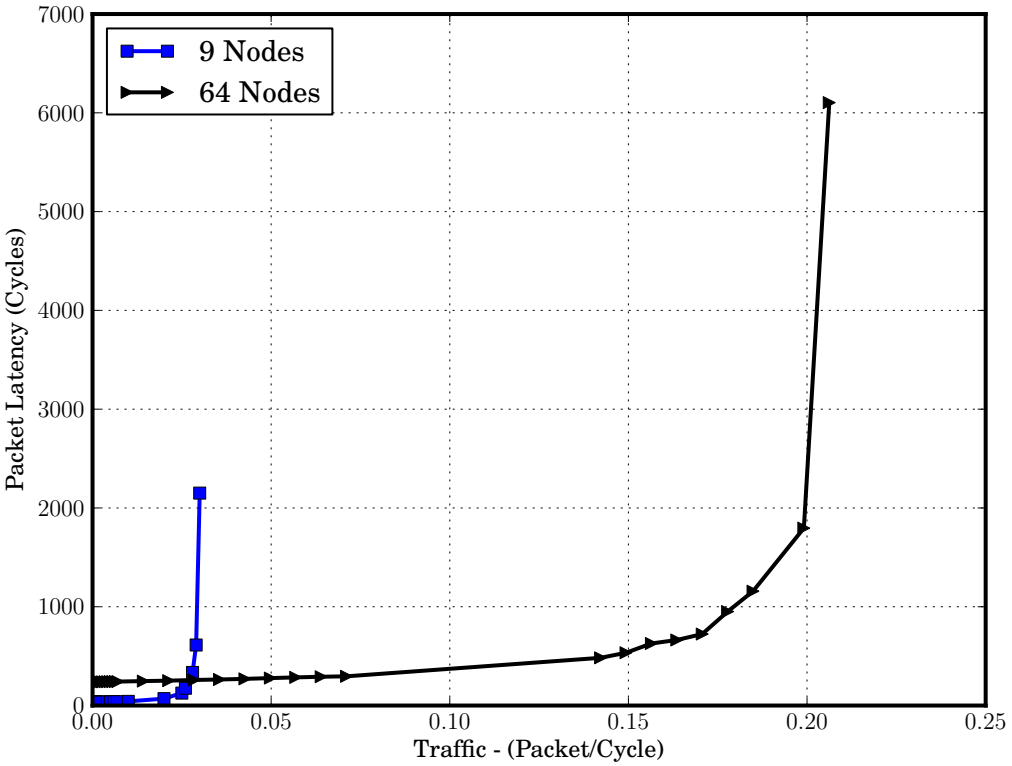

Figure 7.10: Latency versus Traffic rate for 9and 64 nodes 1D multichannel network model.

7.5 Performance Comparison

Performance comparison of any interconnection network should take into account implementation costs to give a meaningful evaluation. Several criteria have been proposed in the literature to make fair comparison between networks of fixed cost including chip pin-out and wiring density [1, 105, 26]. Bisection width as a measure of network cost has been used by Dally [1] and other researchers thereafter [114, 6, 8] to account for wiring density. However, I use in my discussions the term channel to refer to a link as a wireless channel. The bisection width of a network is the minimum number of channels cut when the network is divided into two equal halves. One way to approximate the channel density of a network is the calculation of the bisection

width [114, 8]. The bisection width across a cluster in Bus and DCSH with a channel width of W_{Bus}, W_{DCSH}, respectively, are given by

$$B_{Bus} = W_{Bus} \tag{7.11}$$

$$B_{DCSH} = NW_{DCSH} \tag{7.12}$$

If the bisection width is held constant for both networks, the channel width of the DCSH in terms of shared media (Bus) channel is given by

$$W_{DCSH} = \frac{1}{N} W_{Bus} \tag{7.13}$$

For illustration I will use network sizes of 64 and a packet size of 34 phits, typical figures which has been used by [30, 8], but the results shown in figure (7.11) are applicable across other network sizes and packet lengths.

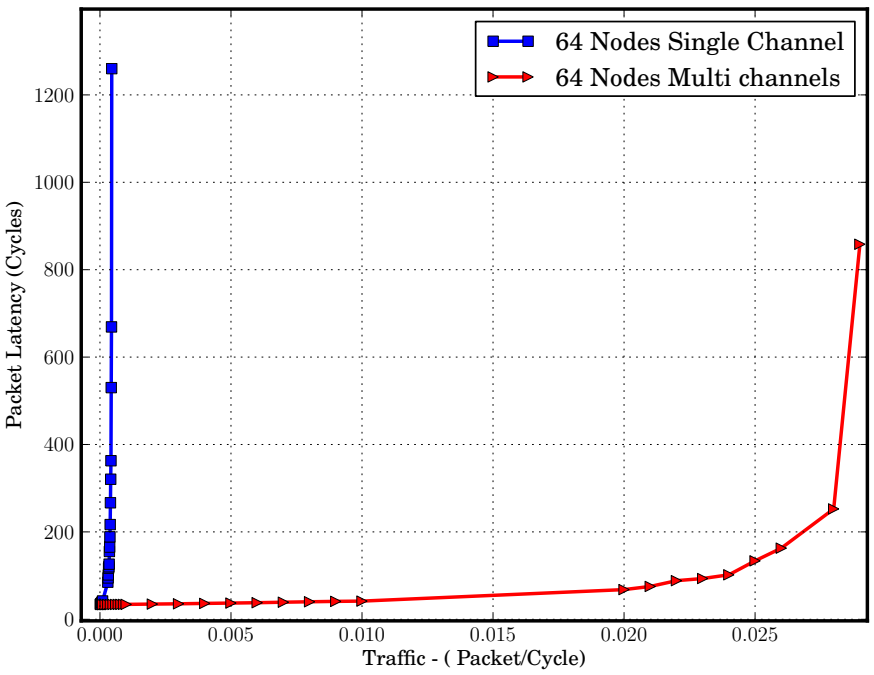

Figure 7.11: Latency comparison for 64 nodes network model

7.6 Conclusion

In this chapter I addressed the performance comparison of contention based wireless networks, which was introduced in chapter 6, and Multi-channel network, using SystemC design methodology. For this purpose, I developed a conceptual model of the communication node of the multichannel network based on my previous node architecture present on chapter 6 and the communication channel which was introduced in chapter 4 and compared my findings analytically and by appropriate simulations.

This chapter has investigated the relative performance merit of the two different implementations. The results have revealed that the multi-channel network has superior performance characteristics over the shared communication network. To my knowledge, this is the first time this kind of performance comparison has been un-

dertaken in SystemC and incorporated into a uniform design methodology, suitable for developing new technologies following the SystemC design methodology.

Further research should be done to optimise the interconnection and its protocols towards the multicomputer system network applications needs. Multi-channel wireless or multidimensional interconnections will provide the required throughput at high bit rate and a minimum latency.

8 Dual-Radios Hypermesh Network based on CSMA Protocol

Recently, multi-radio mesh technology in wireless networks has been put under extensive research. This is because of its potential to overcome the inherent wireless multi-hop throughput, scalability and latency problems caused by the half-duplex nature of the IEEE 802.11. This chapter introduces a design and modelling of different elements on a distinct type of multicomputer networks, the dual-radio wireless hypermesh, based on SystemC methodology.

The hepermesh is a well-known topology that belongs to the hypergraph family of networks. Hypermeshes have been proposed as potential alternatives to the graph networks for the future System Area Networks. In this work, I consider a two dimensional dual-radio wireless hypermesh network, where each router node is equipped with two radio interfaces and two non-overlapping channels are available for each node. I address the problem of assigning channels to communication links in the network with the objective of keeping overall network latency low and provide a relatively high throughput.

The simulations and analysis have shown that my design achieves a significant increase in network throughput with less average network latency for large number of communication nodes, compared with the CSMA shared channel model, which is currently the de facto MAC protocol for most wireless networks. My simulations

have been validated analytically to show the accuracy of the developed model. In addition, simulation results have shown that the wireless hypermesh outperforms shared medium wireless networks under the constant total bandwidth argument, especially in large networks.

8.1 Introduction

The increasing demand for high performance computing from the military and research centres in industry and academia for the simulation of complex problems makes SANs a very interesting research area. SANs are Hardware/Software systems designed to perform specific applications in which network communications are essential. For this reason, their processing functionality is strictly connected with communication functionality. Typically, the processing part is implemented through CPU, memory, and application-specific components; the communication part is implemented through HW components (e.g., network interface and wired or wireless links) and SW components (e.g., the protocol stack). The design of SANs is the context of this thesis. It requires the capability of modelling and simulating both their behaviour/architecture and the complex communication environment in which SANs operate.

The success of Large SANs for multicomputer networks is highly dependent on the design and the efficiency of the interconnection network used. Interconnection network is constructed normally from routers, switches and channels. It provides the means by which the information is transferred between nodes. The design and implementation of a communication network for applications such as scientific computing research applications, for instance, fluid dynamics or finite element methods, modelling of nuclear explosions, climate modelling, and others has a direct impact on the cost and performance of the whole system.

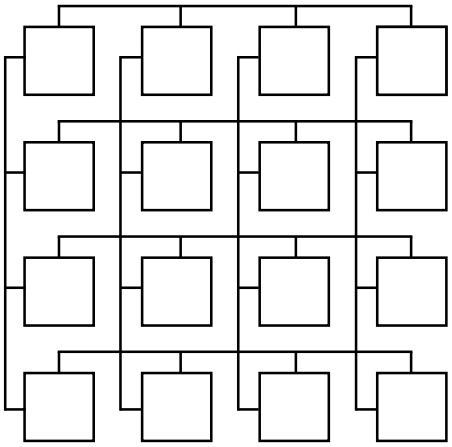

Figure 8.1: A two-dimensional hypermesh

There are several network architectures and topologies for implementing interconnection networks have been introduced in literature to use an interconnection networks with mixed level of success. Some of the most important networks are: Fully connected or all-to-all, a Circular Ring, a Star, a Binary tree, Mesh (Torus), Hypermeshes, or even Random networks. Most of the above mentioned interconnections are designed to implement wired networks, which can have more applications in a wireless sensors network.

Recently hypermeshes have been shown to be a good promising candidate for interconnection networks of parallel systems. Hypermeshes have desirable features over other interconnection networks such as a low diameter, high bandwidth, and low latency network. Those advantages make the hypermeshes to embed naturally a wide range of communication patterns [9]. A number of different hypermesh implementations have been suggested in the literature including shared buses, crossbar switches, and distributed crossbar switch with different implementation technologies, costs and constraints [10, 5, 6, 8, 9]. Figure 8.1 shows a typical example of a hypermesh implementation, which is the spanning bus hypercube (SBH) proposed by [10]. And it has been further studied by [6, 8]. In their introduced model nodes in a given dimension are connected together with a shared bus. A bus in the SBH is time multiplexed among the cluster nodes, which is not suitable for a large number of wirelessly communicating nodes.

Multi-radio multi-channel wireless networks have emerged as a promising form of technology to improve the network performance. Many researchers have studied the method to use multi non-overlapping channels based on the well known knowledge which is that channel capacity is proportional to bandwidth (Shannon Theorem)[1]. Since single-radio mesh networks are unable to effectively scale in order to exploit the increasing system bandwidth available, the use of multiple radio nodes in a network appears to provide one of the most promising avenues to network scaling. In such networks each node is equipped with several network interference cards and able to access multiple non-overlapping channels simultaneously [25, 23, 48, 115, 43]. While so far most works in the literature deals with issues related to practical and efficient use of hypermeshes in wired networks, in this work, I proposed a design called the Dual-Radio Wireless Hypermesh (DRWH) Figure (8.2), which, I believe, is more efficient kind of wireless mesh network. The aim of this work is on improving the network throughput while maintaining a relatively low latency of a wireless network system by means of a CSMA-based design of the MAC protocol. Moreover based on the desirable features of hypermesh network topology the design can boost the required throughput.

Before any communication node connected to the shared medium tries to send a packet, a node has to gain access to the shared media. Once it is granted access, it uses the full media bandwidth to transmit its packets. After that it releases the media to allow other nodes, which might compete on the media, to have access to the communication medium.

The network topology shown in Figure (8.1) represents a two-dimensional hypermesh network configuration. The trick in modelling such a network is the design of the switching element. But the construction of the network itself is a Cartesian product of the bus topology of one cluster from itself. In this network each dimension requires a shared channel to connect. This network has a very low diameter and it scales

[1]Shannon's formula: $Capacity = Bandwidthe \times \log_2(1+SNR)$, where $Bandwith$ is the bandwidth of the channel, SNR is the signal to noise ratio and $Capacity$ is the capacity of the channel in bits per second

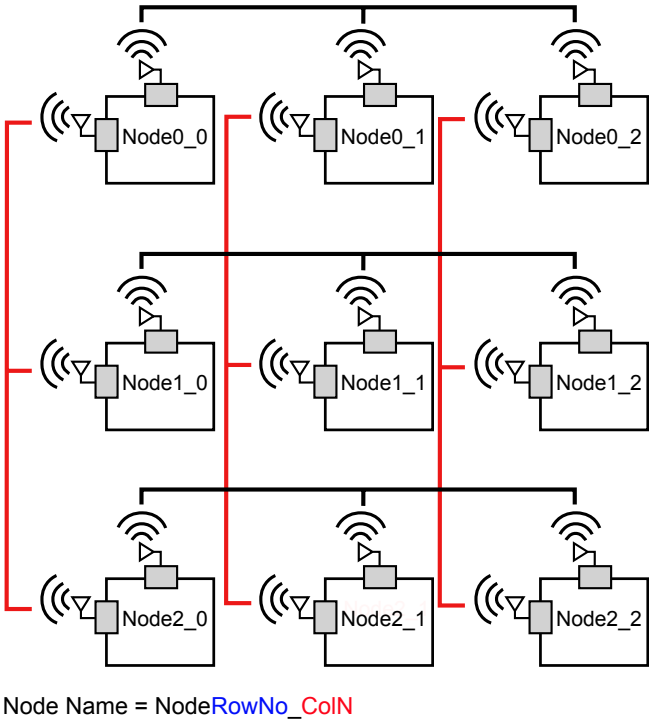

Figure 8.2: Dual-Radio Hypermesh Topology

very well with the network size [26].

The remainder of this chapter is organized as follows. Section 8.2 highlights the channel assignment strategy based on the hypermesh network configuration. Section 8.3 introduces the design and implementation of the communication node and a description of how it can be efficiently equipped with to radio interfaces to utilise the channel assignment method described in section 8.2. Section 8.4 introduces the router architecture and the routing technique used in this work. Section 8.5 briefly outlines the analytical model developed to validate my simulation results in this chapter. In Section 8.6, simulation results are presented and compares the performance of the implementation scheme with the shared medium implementation described in chapter 6. And finally, section 8.7 is the conclusion.

8.2 Network architecture and channel assignment strategy

One of the fundamental challenges in wireless network research is how to increase the overall network throughput while maintaining low latency for packet processing and communications. The low throughput is attributed to the harsh characteristics of the radio channel combined with the contention-based nature of medium access control (MAC) protocols commonly used in wireless networks such as IEEE 802.11.

The hypermesh network, as illustrated in Figure (8.2), consists of communication nodes, which are constructed from routers and switches. Therefore any node in the network can receive and forwarde data packets on behalf of other nodes that may not be within direct transmission range of their destination.

A fixed channel assignment solution based on the hypermesh network configuration mitigates many disadvantages of the conventional wireless network constraints as on a single channel networks using a single radio, where the network is not scalable and suffer from high contention for large number of nodes . Since the number of radios is much higher than the number of available channels, the channel assignment must obey the constraint that many links between the nodes will be operating on the same set of channels [41, 42]. Therefore, the main issue in such network configuration is the channel assignment problem which involves assigning (binding) each radio to a channel in such a way that efficient utilisation of available spectrum can be achieved. And also an adequate level of connectivity among the network nodes is guaranteed. In other words, the assignment of channels to radios should ensure that multiple paths are available among network nodes. This is a major characteristic of the hypermesh network configuration which will allow us to assign multiple nonoverlapping channels to the nodes, which can significantly alleviate the capacity problem and increase the aggregate bandwidth available to the network.

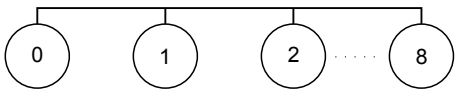

Figure 8.3: One channel and one interface.

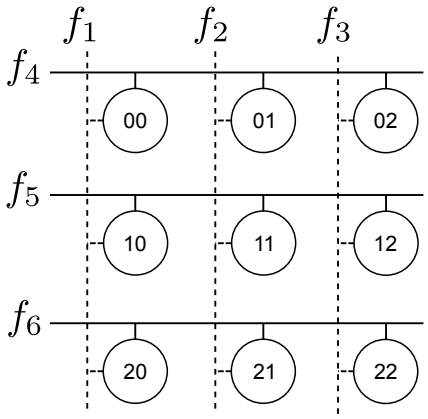

Figure 8.4: 6-available channels 2-interfaces/node

A hypermesh node needs to share one channel with each of its neighbours in a given dimension. In other words, node shares a channel with the nodes in the x-dimension cluster and shares other channel with nodes in the y-dimension cluster. Utilising the communication spectrum in this way the node will minimise the number of neighbours that shares a common channel with and therefore will reduce network interference.

The example in Figure (8.3) and Figure (8.4) illustrats the idea used for assigning channels to different nodes. Figure (8.3) shows the connectivity of the network when a single channel is operating on a single radio. In this scenario, a shared communication link is placed between all nodes to represent a shared communication medium between communicating nodes. This scenario achieves a maximum network connectivity since a single common channel is shared between all nodes. However this configuration suffers from throughput degradation and high latency for large number of nodes due to contention.

For the multi-channel multi-interfaces scenario represented in Figure (8.4) there are six orthogonal non-overlapping channels available for communication, given that every node is equipped with two radios. In this scheme the assignment of channels to radios results a better utilisation of the spectrum, since this has reduces the

contention by distributing the nodes on clusters. Since each node has been equipped with two radios, there is a radio for each dimension. The radios of each node in the same cluster or dimension are all assigned the same frequency.

In a multi-channel system, the transmitter and receiver must both use an agreed upon channel for communication. This introduces the hidden and exposed node problem. The hidden/exposed node problem is a well known issue in wireless networks [116]. A hidden node refers to a node which is outside the coverage area of the transmitting node but within the coverage area of the receiving node. A hidden node is unable to sense the ongoing transmission, and therefore it may try to transmit and inevitably create interference (or possible packet collision) to the receiving node. Exposed is a node that is located within the coverage area of the transmitting node but outside the coverage area of the receiving node. As a result, the exposed node will sense the ongoing transmission and will defer its transmission while it should be able to transmit (to another available receiver) since its transmission will not interfere with the ongoing transmission. The reason the hidden and exposed node problems happen is because nodes lack knowledge about channel usage. Idle nodes that overhear channel negotiation may help other nodes make informed decisions.

Proposed solutions to the hidden/exposed node problem are the transmission of a busy tone from the receiver in a separate channel [19, 116] and the exchange of signaling between transmitter and receiver (request-to-send (RTS) and clear-tosend (CTS)) Virtual Carrier Sensing in CSMA/CA before the actual transmission takes place [117]. The exposed or hidden node problem is not considered in my proposed model and it has been left as an exercise for a future work.

8.3 Design and Implementation

When two or more radio interfaces are placed on a node, I am faced with a number of choices as to the way they are interconnected. Two main alternatives are bridging at the link layer and connecting the interfaces at the network layer. While bridging at the link-layer may be acceptable in a wired network, in its wireless counterpart traffic unnecessarily relayed can significantly degrade performance. In this work, I adopted a more generic solution by connecting the interfaces at the routing level. The individual PHYs act as separate entities with different addresses for routing purposes.

In this section, I further expand on this design and describe my implementation of a dual-interface node in SystemC. The communication nodes are the main construction units of the network system, which will communicate with each other through the underlying interconnection network. The nodes of the network are modelled as modules with different methods, which will reflect the application, network and DLC and PHY of the standard reference model of an OSI/ISO. Each node interacts with the network to send or receive data packets. Each node is assumed to have two baseband transmitters and receivers for communication with other network nodes as shown in Figure (8.5).

The two dimensional node structure has been modelled by adding the necessary components to the one dimensional model introduced in Chapter 6. By adding a network layer for routing and switching data packets and another PHY layer for transmission of data packets to the second dimension the hypermesh node has been constructed.

Throughout the analysis and simulation the following assumptions have been made.

- Traffic generated by nodes independently of each other, and follows a Poisson

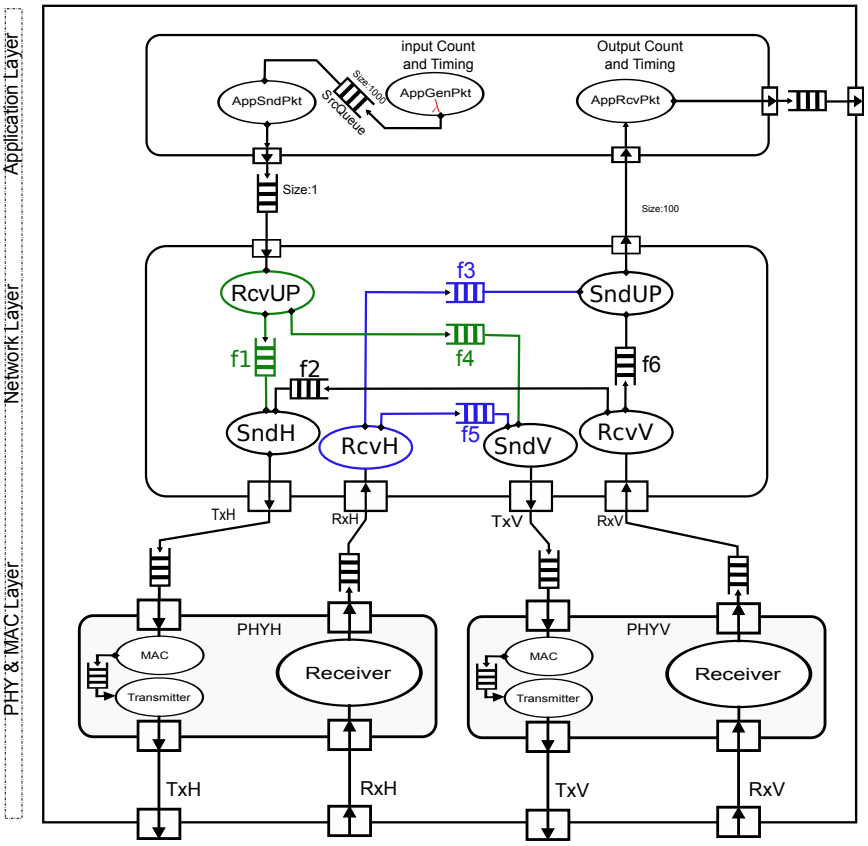

Figure 8.5: Two Dimensional model of the communication node

process of a mean rate λ (packets/cycle). Furthermore destinations are chosen randomly and independently. This assumption is widely used in the literature as it greatly simplifies the analysis [1, 5, 118, 9].

- All packets generated are of equal length, M phits, each of which requires M-cycle transmission time.

- Packet destinations are uniformly distributed across the network unless specified otherwise. Although many network evaluation studies make this simplifying assumption, it is rarely true in practice [105].

- Routing is restricted; packets are transmitted between routers using packet switching (store-and-forward). Dimension-ordered routing [5, 8], where packets visit network dimensions in a strict order (it is assumed here that dimen-

sions are visited in a decreasing order), is sufficient to avoid packet deadlock.

- Packet queues are assumed to have infinite capacity. This assumption has been shown to be realistic under uniform traffic.

- Balanced network to avoid complexity in my analysis and in my proposed topology this assumption is easily realizable. A network is said to be balanced if the utilization factor of all of its channels is the same.

- Negligible channel propagation delay.

8.4 Router Architecture

The network layer is concerned with the exchange of data between an end system and the network to which it is attached. The sending node must provide the network with the address of the destination node, so that the network may route the data to the appropriate destination.

Figure (8.6) shows the router structure in a two-dimensional DIWH. Since dimension-ordered routing is used, packets from the local node may access x-dimension or y-dimension, depending on the destination address. Furthermore, only packets arriving on y-dimension from other routers can access x-dimension. Within a dimension (or cluster), say i, a packet generated by a node is sent on the channel of that dimension, and every other node of that dimension compares its node address with the packet destination address. If the addresses are equal, the packet is consumed locally at that node through the local output channel. If the destination address is not in that dimension i, only the node which has coordinates in all dimensions j $(0 \leq j \leq (i-1)$ equal to those of the destination node can buffer the packet and switches it to an outgoing channel, leading it towards its destination [5, 8, 26].

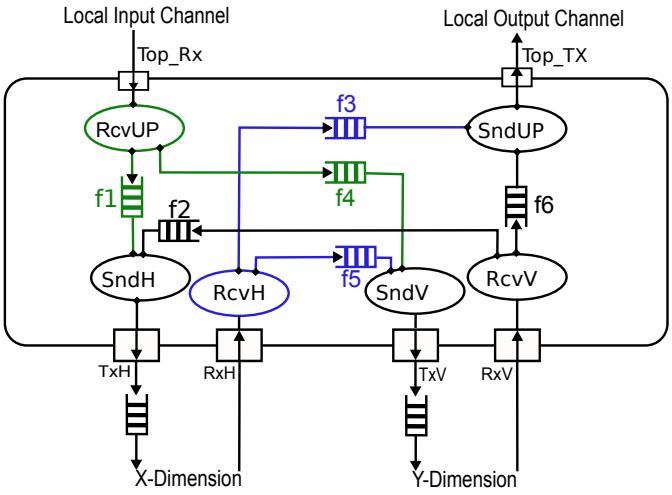

Figure 8.6: The router structure in the two-dimensional DRWH

The implemented switch belongs to the class of output-queuing switches, since the buffering function is performed after the routing function. Packets generated from the same node are copied into one of the two output queues associated with the required channel.

When a node generates a packet, a routing decision is made to select the output queue into which the packet is copied. When a packet arrives at a given router it is totally buffered. A routing decision is made to select the next output channel. It is then copied into its corresponding output queue if it has not reached its destination. Otherwise, it is transferred to the output queue associated with the channel leads to the node itself. The average waiting time at the output queue depends on the rate of packets that switch from one dimension to another.

8.5 Analytical model

This section outlines the detailed derivation and validation through simulation experiments of the modelling approach used to develop the models for the presented work. To build simple mathematical models of packet switching hypermesh, we have

adapted the approach described in [118, 114, 7, 8].

I initially assumed that the networks in question use packet switching where packets visit network dimensions in a strict order. In packet switching [26], a packet is individually routed from source to destination. A packet is buffered at each intermediate node before it is forwarded to the next node. Packet switching technique is advantageous when packets are short and frequent. This technique is also referred in literature as store-and-forward (SAF) switching.

Hypermesh structure is formed from the one-dimensional cluster of shared communication nodes. Basically each cluster is a hypergraph consisting of k nodes connected within a single cluster [8]. A k-ary n-dimensional hypermesh, is a regular hypergraph with $N = k^n$ nodes, $\left(k = \sqrt[n]{N},\ n = \log_k N\right)$, formed by taking the Cartesian n-product of the cluster topology. This has the effect of imposing the cluster organization in every dimension, making each node equally a member of n independent orthogonal clusters. As described in [1], there are two components of latency, distance and diameter.

The network diameter, D, which is the maximum distance between two nodes in the network and the average message distance, a, which is the number of hops required to get from the source to the destination, in the hypermesh are given by [1, 118, 7]

$$D = n \tag{8.1}$$

$$a = n\frac{(k-1)}{k}\frac{N}{(N-1)} \tag{8.2}$$

The first term accounts for the average number of channels that a packet visits to cross the network. And the $({}^{N}/_{(N-1)})$ factor accounts for the fact that a node is not

allowed to send packets to itself. The node degree, d, which represents the number of channels that a node is connected to, is

$$d = n \tag{8.3}$$

Since, for each of its dimension, a node has one common channel for sending and receiving data packets from the other nodes.

In the single cluster implementation described in Chapter (6), nodes in a given cluster are connected by means of a shared medium. Nodes are accessing the shared medium by means of non-persistent CSMA technique. In non-persistent CSMA, if two stations sense the medium is busy, they both immediately enter random backoff, hopefully choosing different backoff values. These random values should be different to allow one of the two stations to begin transmitting before the other in the next sense of the medium. The other will hear the medium busy and refrain from transmitting until the first station has completed its transmission. The node, which has granted access to the medium, it uses the full channel bandwidth to transmit its packet.

Under light traffic ($\lambda \approx 0$), packet blocking is negligible compared with packet transmission time. Therefore, with store-and-forward routing, the packet latency (static latency), L_0 (in cycles) is the product of a and packet transmission time (in cycles).

$$L_0 = a \times M \tag{8.4}$$

Where a is given by Equation (8.2). The first term accounts for the average number of channels that a packet visits under uniform traffic while the second accounts for the transmission time of the whole packets.

The probability ρ that in a given cycle there is a packet on a network channel at dimension $i\,(1 \leq i \leq n)$ (which is the channel utilization given our initial assumption of M-phits packet) is composed of two components.

ρ_t: the probability that the node injects a packet into dimension i. In the steady state, ρ_t is also the probability that a packet exits the network from dimension i.

ρ_s: the probability that a packet switches from dimension $i+1$ to i.

The probability that a packet is generated by a node in a cycle is λ, and the probability that this message routs to a given network channel is $\left(\frac{1}{n}\right)$, yield in

$$\rho_t = \frac{\rho}{n} \tag{8.5}$$

Since the probability that a message exits the network from a dimension is ρ_t, the probability that is stays in the network is $(\rho - \rho_t)$. If a packet does not exit the network it can switch only to the next dimension, and therefore the probability that it does so is given by:

$$\rho_s = (\rho - \rho_t) \tag{8.6}$$

Given that a router is connected to n channels and packets visit, on average, a channels (a is given by Equation (8.2)), the traffic rate on channel i $(1 \leq i \leq n)$ is given by [105]

$$\rho = \frac{\lambda a M}{n} \tag{8.7}$$

ρ_t and ρ_s can be rewritten using the term defined by [119] which gives the probability p_t, that a packet arriving at the switch is destined for its corresponding node.

$$p_t = \frac{1}{a} \tag{8.8}$$

To make the analysis more tractable, the k output queues that compete for the bus i are treated as a single queue with $2k$ input streams; k streams have a rate ρ_t $^{Packet}/_{Cycle}$ coming from the local node as given by Equation (8.5) and another k streams with rate of ρ_s coming from the previous dimension, as given by equation (8.6). That will give a total of $k\rho$ $^{Packet}/_{Cycle}$ joining the queue during a cycle.

To determine the mean waiting time, w, at the output queues at the output side of a router the channel is treated as an $M/M/1$ queuing with a mean waiting time and total traffic of $k\rho$ [113]

$$w = \frac{^1/_\mu}{(1 - k\rho)} \tag{8.9}$$

Where ρ is given by equation (8.7). This intermediate result is similar to that of shared medium network, since it applies the M/M/1 queue but with different traffic rate. The average total latency though the 2-dimensional hypermesh network for store-and-forward with one cycle transit time through a switch, the average delay through a switch is thus $(1 + w)$. Given that the total number of network channels traversed is n, and taking into account the static latency given by equation (8.4), we have

$$Latency = L_0 + n\,(w + 1) \tag{8.10}$$

Each simulation was run until the network reached steady states, that is, until a further increase in simulated network cycles did not change the measured network latency appreciably.

8.6 Results and Performance Comparison

A performance comparison has been carried out between the two-dimensional wireless hypermesh to the shared communication medium for a fixed size N and equal implementation cost. Each channel entering a node has a bandwidth of W $^{bits}/_{Cycle}$. All packets generated are of equal length, M phits, each of which requires one-cycle to be transmitted. A packet length M phits is broken into $B = M \times W$ phits, each of which contains W bits. Implementation cost should be taken into account to compare different network configuration to give a meaningful evaluation of the network performance. There are several measures have been proposed in the literature to make fair comparison between networks of fixed cost including pin-out [26] and wiring density [1, 105, 26].

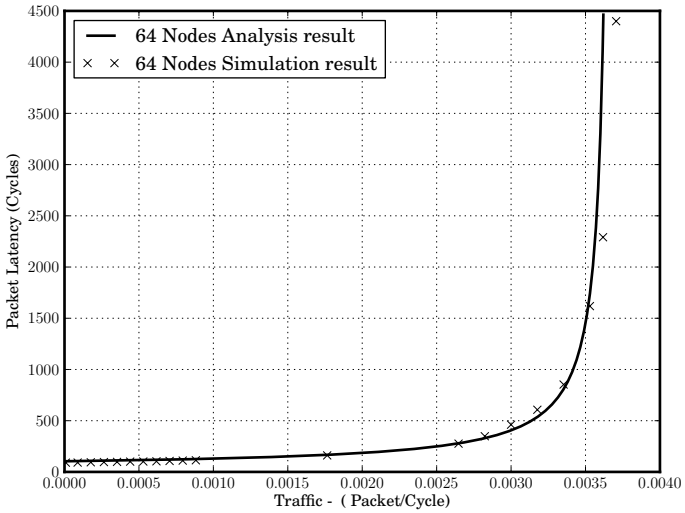

Figure 8.7: Latency vs traffic for 2D Hypermesh for 64 nodes

Under a constant pin-out argument the total bandwidth entering a node is the same for both networks under comparison [30, 8, 9]. As described in section (7.5), to approximate the channels density of a network is the calculation of the bisection width. The bisection width across a cluster in Bus and SBH with a channel width of W_{Bus}, W_{SBH}, respectively, are given by

$$B_{Bus} = W_{Bus} \tag{8.11}$$

$$B_{SBH} = \frac{N}{k} W_{SBH} \tag{8.12}$$

If the bisection width is held constant for both networks, the channel width of the SBH in terms of shared media (Bus) channel is given by

$$W_{SBH} = \frac{k}{N} W_{Bus} \tag{8.13}$$

Figure (8.7) and Figure (8.8) respectively compare the network latency predicted by our model (8.10), and through simulation for 64 and 144 network nodes.

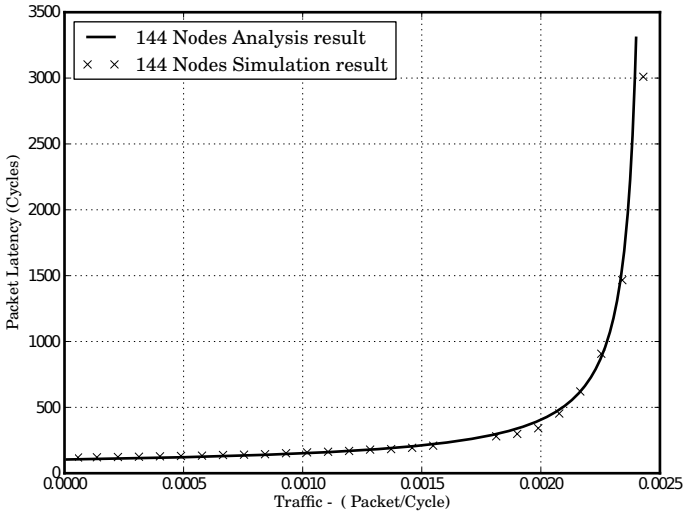

Figure 8.8: Latency vs traffic for 2D Hypermesh for 144 nodes

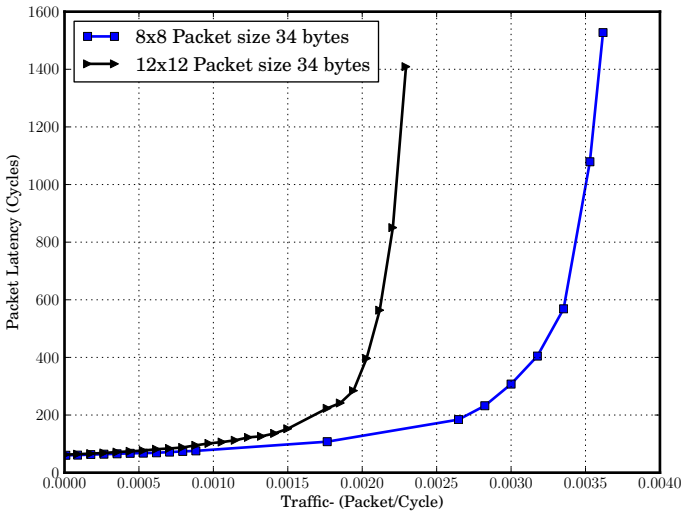

Figure 8.9: Latency vs traffic for 2D Hypermesh for 64 and 144 nodes

To mimic the Hypermesh and make the comparison with the shared medium network the following simulation sitting have been used:

1. Number of nodes $N = 144$ node.

2. Packet size for both networks is (34 Symbols each of which is 8 bits)

3. Number of channels in the hypermesh is $C = i + j = 12 + 12 = 24$ channel, where i and j is the number of channels in each dimension $i = j = 12$

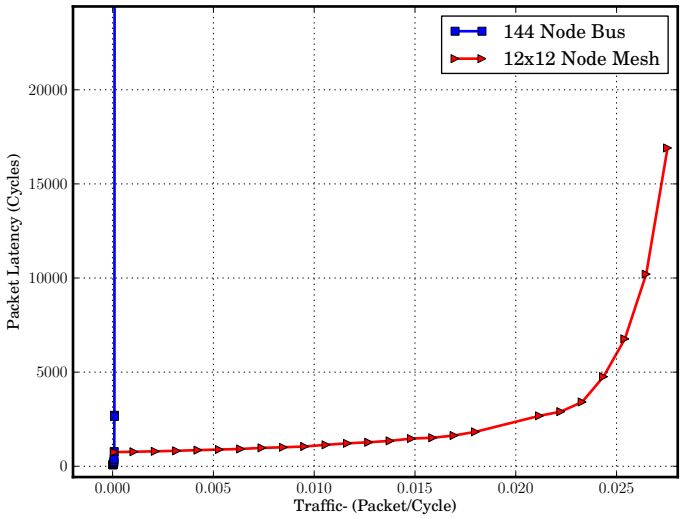

Figure 8.10: Latency vs traffic for 2D and 1D Hypermesh.

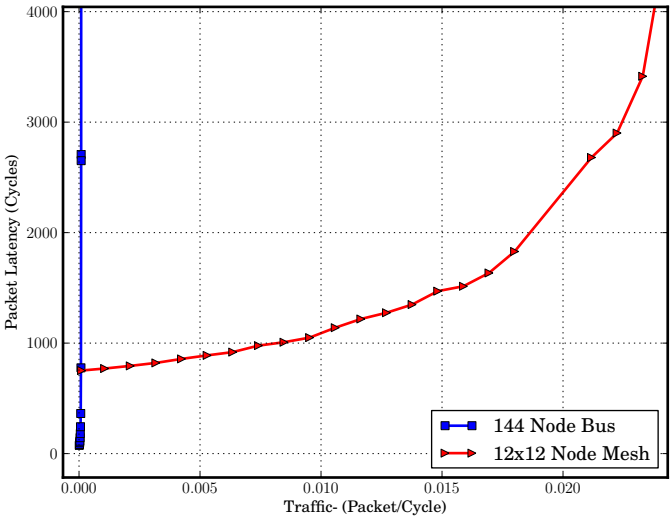

Figure 8.11: For clarity the same graph but with different scale shows the latency vs traffic for 2D and 1D Hypermesh.

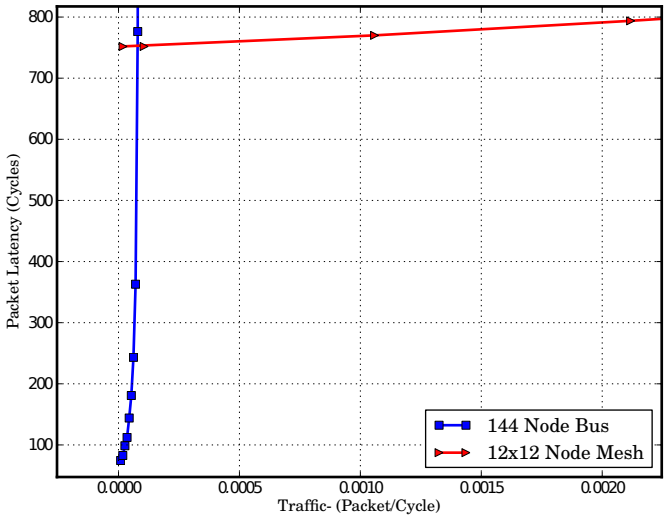

Figure 8.12: For clarity the same graph but with extra zoom on the crossing point of the curves.

We again note that these results were obtained without introducing noise into the communication channels of the system. Although the SystemC model is designed so that random noise and signal fading can be easily introduced, and we showed results of the introduction of such noise in chapter 6, a detailed noise analysis of multichannel and DRWH systems is outside the scope of this thesis.

However it is important to have some indication of the type and magnitude of the effects of noise upon such systems. As noted in chapter 6, the prime effect of such noise is to increase both the traffic in a system, and the latency, due to the need for data re-transmission events. In chapter 2 the effect of noise due to multipath transmission was estimated at between 20-30dB. As a worked example, considering equations 2.8-2.11, if the communication links were set to a bit error rate of 1/100, signal fading of 20dB in the noisy environment would raise the bit error rate to ¼. Assuming hardware error correction led to an equivalent change in packet loss from 1% to 25% and each packet re-transmission took twice the latency of the original transmission (assuming a message returned to the transmitter, but minimal computational cost) then the overall packet latency would increase by 50%, with a

commensurate increase in network congestion.

These noise induced increases in latency and network congestion are not trivial, and emphasise the need for networks, like the DRWH, which are robust in the face of high congestion, to be used.

8.7 Conclusion

In order to deal with fundamental limitations of single frequency wireless networks, I proposed a multi-channel solution based on hypermesh network topology that leverages more than one radio-interface on each node. With this approach, one node can potentially simultaneously communicate with two neighbours on different channels at the same time. I have shown that the hypermesh can be used for assigning channels to radio interfaces in a multi-radio wireless mesh network. Hypermesh channel assignment method reduce the number of contending nodes in each cluster and provide multiple orthogonal non-overlapping channels that allow multiple concurrent nodes to communicate simultaneously on different channels. After presenting the channel assignment scheme based on the hypermesh network topology, I have provided a modelling of the network using SystemC design methodology. Simulation results have shown that the wireless hypermesh outperform shared medium wireless networks under the constant total bandwidth argument, especially in large networks. In addition to the simulation results, the performance improvements have been compared against theory.

When looking at the design space, this innovative channel assignment scheme also addresses its usage and applicability on different wireless network systems. The implementation is portable; it is not restricted to a single specific system area network.

9 Conclusion and Future Work

9.1 Conclusion

In this thesis I modelled and evaluated the performance of wireless networks using SystemC design methodology. SystemC was chosen as it provides a homogeneous platform for the design and modelling of complex systems. Furthermore, as systems become more tightly integrated, the ability to evaluate the system performance at early stages of a design becomes increasingly important. This is facilitated by the system level network modelling techniques in particular SystemC design methodology, and by following an IP-based design. Single interface, single channel and multi-interface multi-channel nodes have been considered in my work. In order to deal with the fundamental limitations of single frequency wireless networks, I proposed a multi-channel solution that leverages more than one radio-interface on each node. With this approach, one node can potentially simultaneously communicate with two neighbours at the same time.

The first part of this thesis has presented a first step towards the integration of communication modelling into the design modelling at the early stages of the system development. This part demonstrates a simple and computationally efficient way to model a noisy communication channel within a system level. The simple noisy digital channel can be used to model the whole communication system interactions. The model is developed at a high level of abstraction which allows for fast simu-

lation and early estimation, which are necessary for successful system development using the SoC design methodology. To my knowledge and to date of our published work, this was the first time that the modelling of wireless communication system was undertaken in SystemC and incorporated into a uniform design methodology, suitable for developing new technologies following the SoC design methodology.

The modelling of a digital communication channel is representing just one component of any communication system. Other components are essential in implementing digital communication systems such as 8B/10B encoder which can increase the transmission performance. The next step was the modelling of an RTL-level SystemC model of an 8B/10B Encoder/Decoder core. As it has been stressed earlier in this thesis that, the use of 8B/10B coding is an important technique in the construction of high performance serial interfaces. In addition, to optimise the use of the transmission medium encoding may be chosen to conserve bandwidth or to minimise errors. Although other HDL models of 8B/10B decoders have been published, to the best of my knowledge, none targeted the SoC design methodology and IP reuse. Although implementation and prototyping of the core is out of the scope of this work, it is important to note that both are relatively simple tasks, as the model developed is already at the RTL-level. The implementation can be carried out by exporting the SystemC code into VHDL or Verilog for synthesis, or directly by using the synthesisable subset of the SystemC language. Once the design has been mapped into a target technology, it is possible to back-annotate timing information directly into the SystemC model, which can aid in providing more accurate timing information for the high level modelling of complete systems.

Furthermore I have addressed the modelling of contention based wireless networks using SystemC design methodology. For this purpose, I have developed a conceptual model of the network communication node and the commutation channel and verified our findings by appropriate simulations. In addition, simulation results have been compared against theory using a M/M/1 queuing model that mimics the behaviour of a shared communication channel as a single queue with multiple inputs at the

phit/symbol level. The results reveal that in all cases, the analytical model predicts the mean packet latency with a good degree of accuracy compared with systemc model in all network regions, i.e., in the steady state region, heavy traffic region and in the saturation region.

Moreover, this work has demonstrated a computationally efficient way to model a wireless communication network within a system level. To our knowledge and to date of our published work, this was the first time that the modelling of wireless communication network was undertaken in SystemC and incorporated the system development into a homogeneous design environment, suitable for developing new technologies following the SystemC design methodology.

Finally, in order to deal with fundamental limitations of single frequency wireless networks, in this work further research has been done in this part to address the communication using multiple orthogonal non-overlapping channels towards the investigation of other design choices under the same framework. Multi-channel wireless interconnection has shown performance improvements in terms of the required throughput at high bit rate and a minimum latency. In addition, I proposed a multi-channel solution based on hypermesh network topology that leverages more than one radio-interface on each node. With this approach, one node can potentially simultaneously communicate with two neighbours on different channels at the same time. We have shown that the hypermesh can be used for assigning channels to radio interfaces in a multi-radio wireless mesh network better than the work present by Raniwala et al. [23]. Hypermesh channel assignment method reduce the number of contending nodes in each cluster and provide multiple orthogonal non-overlapping channels that allow multiple concurrent nodes to communicate simultaneously on different channels, which could not be done in other way. My solution confirms Nasipuri et al. [25] solution, which argues that in the extreme case when channel assignment is perfect and each pair of nodes has a dedicated channel, contention and collision disappear.

After presenting the channel assignment scheme based on the hypermesh network topology, I provided a modelling of the network using SystemC design methodology. Simulation results have shown that the wireless hypermesh outperform shared medium wireless networks under the constant total bandwidth argument, especially in large networks. In addition to the simulation results, the performance improvements were compared against theory. When looking at the design space, this innovative channel assignment scheme also addresses its usage and applicability on different wireless network systems. The implementation is portable; it is not restricted to a single specific system area network. Unfortunately integrating the noisy wireless channel into the hypermesh was challenging and requires further investigations using different techniques. This is due to the SAF switching technique used was not suitable with the use of contention based protocol required for the proper operation of the channel access mechanism. The SAF was accumulating the packets at the input queues of the intermediate nodes causing long delays, since packets are waiting to get channel access for the next dimension. This intern triggers the timer timeout event of the source node and forces multiple retransmissions of the same packets, which saturates the network.

9.2 Future Work

There are several aspects of this work that can be extended either to support the main system or to improve its performance.

First, the channel model is sufficiently flexible to support various numbers of channel models with a configurable numbers of paths per channel. The implementation needs further extensions to include more realistic and accurate radio propagation channel models. A useful method of presenting these models is through the three main factors that affect signal propagation: path loss attenuation, slow-fading (shadowing), and fast-fading (Rayleigh fading) [53, 120].

Second there is further refinement that could enhance the physical layer from digital channel modelling level to the modulation channel and then to radio channel modelling. This should provide real implementation of different modulation techniques such as ASK, PSK, QAM etc.. In addition, hardware prototyping of the physical layer is becoming an indispensable technique in the design and verification of rapidly-evolving modern wireless systems. Thus, the developed channel should be parametrised and become a baseband verification system for single and multiple-antenna wireless systems. By mapping the computationally-intensive signal processing algorithms in the simulation chain to dedicated Field-Programmable Gate Array (FPGA).

Third the communication nodes design can be enhanced to support the ad hoc features, which does not require any pre-existing infrastructure. I believe, nodes could benefit from the provided connectivity without the need to configure their software manually. Moreover the CSMA mechanism could be used with RTS/CTS option for better channel access. To deploy this method, the system designer will need to integrate other protocols, which might affect routing and switching technique used. This should allow the nodes to place them self on a way to meet the hypermesh topology to provide adequate coverage and sufficient capacity.

Bibliography

[1] W. J. Dally, "Performance analysis of k-ary n-cube interconnection networks," *IEEE Trans. Comput.*, vol. 39, no. 6, pp. 775–785, 1990.

[2] W. J. Dally and B. Towles, *Principles and Practices of Interconnection Networks*. Morgan Kaufmann Publishers, stranded ed., 2004.

[3] F. Rodrguez-Salazar, *Small-World Interconnection Networks for Large Parallel Computer Systems*. Phd thesis, Department of Electronics and Electrical Engineering, Glasgow University, April 2004.

[4] J. Marczynski and D. Tabak, "A wireless interconnection network for parallel processing," *IEEE Symposium on Digital Systems*, pp. 386–389, Sep. 2001.

[5] M. Ould-Khaoua, L. Mackenzie, R. Sutherland, and R. Sotudeh, "Constraint-based evaluation of hypergraph and graph networks," *Simulation Practice and Theory*, vol. 4, no. 2-3, pp. 119 – 140, 1996.

[6] M. Ould-Khaoua and L. M. Mackenzie, "On the design of hypermesh interconnection networks for multicomputers," *Journal of Systems Architecture*, vol. 46, no. 9, pp. 779 – 792, 2000.

[7] M. Ould-Khaoua, "Distributed crossbar switch hypermeshes: Efficient networks for large-scale multicomputers," *Journal of Parallel and Distributed Computing*, vol. 61, no. 8, pp. 999 – 1012, 2001.

[8] S. Loucif, M. Ould-Khaoua, and A. Al-Ayyoub, "Hypermeshes: implementation and performance," *Journal of Systems Architecture*, vol. 48, no. 1-3, pp. 37 – 47, 2002.

[9] F. Rodriguez and J. R. Barker, "Hamming hypermeshes: High performance interconnection networks for pin-out limited systems," *Elsevier B.V.*, vol. Performance Evaluation, pp. 759–775, Aug. 2005.

[10] L. Wittie, "Communication structures for large networks of microcomputers," *Computers, IEEE Transactions on*, vol. C-30, pp. 264 –273, april 1981.

[11] T. Groetker, S. Liao, G. Martin, and S. Swan, "System design with systemc," *Kluwer Academic Publishers*, 2002.

[12] G. Martin, B. Bailey, and A. Piziali, *ESL Design and Verification: A Prescription for Electronic System Level Methodology (Systems on Silicon)*. Morgan Kaufmann, April 2007.

[13] C. Widtmann, *High-level System Modeling with SystemC and TLM: Introduction and practical application of an Electronic System Level design flow*. VDM

Verlag, 2009.

[14] P. Express, "Pci express 2.0 specification." http://www.pcisig.com, May 2006.

[15] Infiniband, "Infiniband." http://www.infinibandta.org/index.php, May 2007.

[16] HyperTransport, "Hypertransport." http://www.hypertransport.org/, May 2007.

[17] CPRI, "Common public radio interface." http://www.cpri.info/, May 2007.

[18] M. Rubinstein, I. Moraes, M. Campista, L. Costa, and O. Duarte, "A survey on wireless ad hoc networks," *International Federation for Information Processing,IFIP*, vol. 211, pp. 1–33, 2006.

[19] S.-L. Wu, C.-Y. Lin, Y.-C. Tseng, and J.-P. Sheu, "A new multi-channel mac protocol with on-demand channel assignment for multi-hop mobile ad hoc networks," p. 232, 2000.

[20] Y. Yang, J. Wang, and R. Kravets, "Designing routing metrics for mesh networks," *Proceedings of the IEEE Workshop on Wireless Mesh Networks (WiMesh). IEEE Press*, 2005.

[21] P. Kyasanur and N. H. Vaidya, "Routing and link-layer protocols for multi-channel multi-interface ad hoc wireless networks," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 10, no. 1, pp. 31–43, 2006.

[22] J. So and N. Vaidya, "A routing protocol for utilizing multiple channels in multi-hop wireless networks with a single transceiver," October 2004.

[23] A. Raniwala, K. Gopalan, and T. cker Chiueh, "Centralized channel assignment and routing algorithms for multi-channel wireless mesh networks," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 8, no. 2, pp. 50–65, 2004.

[24] J. Mo, H.-S. So, and J. Walrand, "Comparison of multichannel mac protocols," *Mobile Computing, IEEE Transactions on*, vol. 7, pp. 50 –65, jan. 2008.

[25] A. Nasipuri, J. Zhuang, and S. Das, "A multichannel csma mac protocol for multihop wireless networks," *Wireless Communications and Networking Conference, 1999. WCNC. 1999 IEEE*, pp. 1402 –1406 vol.3, September 1999.

[26] J. Duato, S. Yalamanchili, and L. Ni, *Interconnection Networks: An Engineering Approach*, vol. Revised Printin. Morgan Kaufmann Publishers, 2003.

[27] R. Knopp, "Two-way radio networks with a star topology," *Int. Zurich Seminar on Communications (IZS)*, pp. 154 –157, February 2006.

[28] R. Nadiv and T. Naveh, "Wireless backhaul topologies: Tree vs. ring," tech. rep., Ceragon Networks Ltd., August 2010.

[29] L. Mackenzie, M. Ould-Khaoua, R. Sutherland, and T. Kelly, "Cobra: a high performance interconnection for large multicomputers," tech. rep., Department of Computer Science, University of Glasgow, mar. 1991.

[30] M. Ould-Khaoua, *Hypergraph-based interconnection networks for large multicomputers.* PhD thesis, Department of Computing Science, University of Glasgow, September 1994.

[31] I. D. Scherson and A. S. Youssef, eds., *Interconnection networks for high-*

performance parallel computers. Los Alamitos, CA, USA: IEEE Computer Society Press, 1994.

[32] R. McTasney, D. Grunwald, and D. Sicker, "Low-latency multichannel wireless mesh networks," *IEEE, Computer Communications and Networks, Proceedings of 16th International Conference*, pp. 1082–1087, Aug. 2007.

[33] A. C. V. Gummalla and J. O. Limb, "Wireless medium access control protocols," *IEEE Communications Surveys and Tutorials*, vol. 3, no. 2, pp. 2 –15, 2000.

[34] A. F. Jost Weinmiller, Morten Schlaeger and A. Wolisz, "Performance study of access control in wireless lans ieee 802.11 dfwmac and etsi res 10 hiperlan," *Mobile Networks and Applications*, vol. 2, pp. 55–67, 1997.

[35] A. S. Tanenbaum, *Computer Networks*. Pearson Education, july 2002.

[36] N. Abramson, "The aloha system: another alternative for computer communications," pp. 281–285, 1970.

[37] L. Kleinrock and F. Tobagi, "Packet switching in radio channels: Part i–carrier sense multiple-access modes and their throughput-delay characteristics," *Communications, IEEE Transactions on*, vol. 23, pp. 1400 – 1416, dec 1975.

[38] G. Giambene, *Queuing Theory and Telecommunications: Networks and Applications*. Springer, Jun 2005.

[39] D.Sivaganesan and Dr.R.Venkatesan, "Comparative analysis of wireless mac protocol for collision detection and collision avoidance in mobile ad hoc networks," *International Journal of Computer and Network Security (IJCNS)*, vol. 2, pp. 40–45, march 2010.

[40] P. Kyasanur and N. H. Vaidya, "Routing and interface assignment in multi-channel multi-interface wireless networks," vol. 4, pp. 2051 – 2056 Vol. 4, 13-17 2005.

[41] H. Skalli, S. Ghosh, S. Das, L. Lenzini, and M. Conti, "Channel assignment strategies for multiradio wireless mesh networks: Issues and solutions," *Communications Magazine, IEEE*, vol. 45, pp. 86 –95, november 2007.

[42] A. P. Subramanian, H. Gupta, S. R. Das, and J. Cao, "Minimum interference channel assignment in multiradio wireless mesh networks," *IEEE Transactions on Mobile Computing*, vol. 7, no. 12, pp. 1459–1473, 2008.

[43] H. M. K. Alazemi, A. Das, R. Vijaykumar, and S. Roy, "Fixed channel assignment algorithm for multi-radio multi-channel mesh networks," *Wirel. Commun. Mob. Comput.*, vol. 8, no. 6, pp. 811–828, 2008.

[44] C. M. Gug, P. J. Soo, and K. H. Jim, "Multi-interface multi-channel hybrid channel assignment with ieee 802.11 in using a reactive routing protocol in manet," *Information Science and Applications (ICISA), 2010 International Conference on*, pp. 1–5, apr. 2010.

[45] R. Draves, J. Padhye, and B. Zill, "Routing in multi-radio, multi-hop wireless mesh networks," pp. 114–128, 2004.

[46] M. Marina and S. Das, "A topology control approach for utilizing multiple

channels in multi-radio wireless mesh networks," pp. 381 – 390 Vol. 1, 3-7 2005.

[47] W. Fu, B. Xie, X. Wang, and D. Agrawal, "Flow-based channel assignment in channel constrained wireless mesh networks," pp. 1 –6, 3-7 2008.

[48] Y. Zhang, J. Luo, and H. Hu, *Wireless Mesh Networking: Architectures, Protocols and Standards.* Auerbach Publications, 1 ed., 2007.

[49] J. So and N. H. Vaidya, "Multi-channel mac for ad hoc networks: handling multi-channel hidden terminals using a single transceiver," pp. 222–233, 2004.

[50] P. Bahl, R. Chandra, and J. Dunagan, "Ssch: slotted seeded channel hopping for capacity improvement in ieee 802.11 ad-hoc wireless networks," pp. 216–230, 2004.

[51] K. N. Ramachandran, E. M. Belding, K. C. Almeroth, and M. M. Buddhikot, "Interference-aware channel assignment in multi-radio wireless mesh networks," pp. 1 –12, april 2006.

[52] R. Steele and L. Hanzo, *Mobile Radio Communications: Second and Third-generation Cellular and WATM Systems.* Wiley-IEEE Press, 2nd ed., May 1999.

[53] B. Sklar, *Digital Communications: Fundamentals and Applications.* 2 ed., Jan 2001.

[54] M. Schiff, *Introduction to Communication Systems Simulation.* 2006.

[55] J. Proakis and M. Salehi, *Digital Communications.* McGraw-Hill, November 2007.

[56] J. Cavers, *Mobile Channel Characteristics (The Springer International Series in Engineering and Computer Science).* Springer, 1 ed., September 2000.

[57] M. Aguado, J. Astorga, N. Toledo, and J. Matias, "Simulation-based methods for network design," in *Wireless Network Design* (J. Kennington, E. Olinick, and D. Rajan, eds.), vol. 158 of *International Series in Operations Research & Management Science*, pp. 271–293, Springer New York, 2011.

[58] F. Xiong, *Digital Modulation Techniques.* Artech House Publishers, 1 ed., January 2000.

[59] D. Tse and P. Viswanath, *Fundamentals of Wireless Communication.* Cambridge University Press, June 2005.

[60] I. Howitt and Y. Hawwar, "Evaluation of outage probability due to cochannel interference in fading for a tdma system with a beamformer," in *Vehicular Technology Conference, 1998. VTC 98. 48th IEEE*, vol. 1, pp. 520–524, may 1998.

[61] D. Matolak, "Wireless channel characterization in the 5 ghz microwave landing system extension band for airport surface areas," tech. rep., Ohio University, Athens, May 2007.

[62] D. R. Jeff Kennington, Eli Olinick, *Wireless Network Design: Optimization Models and Solution Procedures*, vol. 1. Springer, 2010.

[63] J. D. Parsons, *The Mobile Radio Propagation Channel*. Pentech Press UK, 1992.

[64] M. Engels, *Wireless OFDM Systems: How to make them work?* Springer; illustrated edition edition, illustrated edition ed., July 2002.

[65] F. Hendessi, A. U. Sheikh, and R. M. Hafez, "Co-channel and adjacent channel interference in wireless cellular communications," *Wireless Personal Communications*, vol. 12, pp. 239–253, 2000.

[66] A. Goldsmith, *Wireless Communications*. Cambridge University Press, 2005.

[67] R. Hekmat and P. Van Mieghem, "Interference in wireless multi-hop ad-hoc networks and its effect on network capacity," *Wireless Networks*, vol. 10, pp. 389–399, 2004.

[68] M. D. Kumar A. and K. J., *Communication Networking : An Analytical Approach*. Elsevier, 2004.

[69] A. Arsal, "A study on wireless channel models: Simulation of fading, shadowing and further applications," Master's thesis, School of Engineering and Sciences of Izmir Institute of Technology, Aug. 2008.

[70] F. Stremler, *Introduction to Communication Systems*. 1992.

[71] W. H. Tranter, K. S. Shanmugan, T. S. Rappaport, and K. L. Kosbar, *Principles of Communication Systems Simulation with Wireless Applications*. first ed., 2004.

[72] J. Zyren and A. Petrick, "Tutorial on basic link budget analysis," tech. rep., Intersil Corporation, 1998.

[73] W. Stallings, *Data and Computer Communications*. Prentice Hall, 7 ed., May 2003.

[74] I. OPNET Technologies, "Opnet," 2010.

[75] QualNet, "Qualnet - scalable network technologies," 2010.

[76] OmNet++, "Omnet++ - discrete event simulation system.." http://www.omnetpp.org, 2010.

[77] NS-2, "Ns network simulator," 2010.

[78] N. Bombieri, F. Fummi, and D. Quaglia, "System/network design-space exploration based on tlm for networked embedded systems," *ACM Trans. Embed. Comput. Syst.*, vol. 9, no. 4, pp. 1–32, 2010.

[79] MathWorks, "Matlab." http://www.mathworks.com/, July 2010.

[80] J. Hiner, A. Shenoy, R. Lysecky, S. Lysecky, and A. G. Ross, "Transaction-level modeling for sensor networks using systemc," *Sensor Networks, Ubiquitous, and Trustworthy Computing, International Conference on*, vol. 0, pp. 197–204, 2010.

[81] O. S. I. OSCI, "Systemc," 2007.

[82] L. Cai and D. Gajski, "Transaction level modeling: an overview," pp. 19–24, 2003.

[83] W. Muller, W. Rosenstiel, and J. Ruf, "Systemc: Methodologies and applica-

tions," *Kluwer Academic Publishers*, Mar. 2003.

[84] D. C. Black and J. Donovan, *SystemC from the Ground Up.* first ed., 2004.

[85] J. Bhasker, *A SystemC Primer.* Star Galaxy Publishing, 2 ed., January 2004.

[86] A. Mignogna, M. Conti, M. D'Angelo, M. Baleani, and A. Ferrari, "Transaction level modeling and performance analysis in systemc of ieee 802.15.4 wireless standard," pp. 839 –843, 3-5 2008.

[87] N. Calazans, E. Moreno, F. Hessel, V. Rosa, F. Moraes, and E. Carara, "From vhdl register transfer level to systemc transaction level modeling: a comparative case study," *IEEE International Symposium on Circuits and Systems*, 2003.

[88] S. M. Aziz, "A cycle-accurate transaction level systemc model for a serial communication bus," *Computers & Electrical Engineering*, vol. 35, no. 5, pp. 790 – 802, 2009.

[89] OSCI, *SystemC User's Guide.* Open SystemC Initiative, 2002.

[90] F. Ghenassia, *Transaction-Level Modeling with SystemC: TLM Concepts and Applications for Embedded Systems.* October 2005.

[91] N. R. Adam, "Achieving a confidence interval for parameters estimated by simulation," *Management Science*, vol. 29, pp. 856–866, July 1983.

[92] K. Pawlikowski, D. C. McNickle, and G. Ewing, "Coverage of confidence intervals in sequential steady-state simulation," *Simulation Practice and Theory*, vol. 6, no. 3, pp. 255 – 267, 1998.

[93] D. E. Knuth, *The Art of Computer Programming: Seminumerical Algorithms v. 2*, vol. 2. Addison Wesley, 3 ed., Nov 1997.

[94] M. H. MacDougall, *Simulation Computer systems: Techniques and Tools.* The MIT Press, 1987.

[95] M. Petrov, T. Murgan, P. Zipf, and M. Glesner, "Functional modeling techniques for a wireless lan ofdm transceiver," *IEEE International Symposium on Circuits and Systems*, vol. 4, pp. 3973–3973, May 2005.

[96] B. A. Forouzan, "Data communications and networking," *McGraw Hill*, 2004.

[97] A. Widmer and P. Franaszek, "A dc-balanced partitioned-block 8b/10b transmission code," *IEEE*, vol. 27, pp. 440–451, Sep. 1983.

[98] Lattice, "8b/10b encoder/decoder," *Lattice Semiconductor*, vol. Reference Design RD1012, Nov. 2002.

[99] J. Wu and Y. Hsu, "8b/10b codec for efficient papr reduction in ofdm communication systems," *IEEE International Conference on Wireless Communications, Networking and Mobile Computing*, pp. 1198–1201, 2005.

[100] S. Sumner, "A brief overview of 10 gigabit ethernet," tech. rep., EXFO, 2005.

[101] J. Booth, "Systemc modeling of a parallel processor broadcast interconnection system," *IEEE SoutheastCon*, pp. 76 – 81, 2002.

[102] A. Ghosh, S. Tjiang, and R. Chandra, "System modeling with systemc," *ASIC, 2001. Proceedings. 4th International Conference on*, pp. 18–20, 2001.

[103] M. Conti and D. Moretti, "System level analysis of the bluetooth standard," pp. 118 – 123 Vol. 3, 7-11 2005.

[104] D. Chkliaev, J. Hooman, and E. de Vink, "Verification and improvement of the sliding window protocol," 2003.

[105] A. Agarwal, "Limits on interconnection network performance," *IEEE Trans. Parallel Distrib. Syst.*, vol. 2, no. 4, pp. 398–412, 1991.

[106] R. N. Williams, "A painless guide to crc error detection algorithms." http://www.ross.net/crc/crcpaper.html, May 2008.

[107] J. D. Spragins, J. L. Hammond, and K. Pawlikowski, *Telecommunications Protocols and Design*. 1992.

[108] J. F. Kurose and K. W. Ross, *Computer Networking: A Top-down Approach featuring the internet*. 3 ed., 2005.

[109] C. Hu, H. Kim, and J. C. Hou, "An analysis of the binary exponential backoff algorithm in distributed mac protocols," tech. rep., University of Illinois, Urbana Champaign, 2005.

[110] H. Saiedian, "A multi-purpose simulation project for engaging students and teaching object concepts," *Computer Science Education*, vol. 8, pp. 64 – 81, Mar. 1998.

[111] HyperTransport-Technology-Consortium, "A memory-efficient implementation of the hypertransport coupon-based flow control protocol," tech. rep., HyperTransport Technology Consortium, Oct. 2006.

[112] AMBA, "Advanced microcontroller bus architecture," 2010.

[113] L. Kleinrock, *Queueing Systems, Computer Applications*, vol. Volume 2. John Wiley and Sons, New York., 1976.

[114] L. Mackenzie and M. Ould-Khaoua, "Comparative modeling of network topologies and routing strategies in multicomputers," *Int. J. High Perform. Comput. Appl.*, vol. 14, no. 3, pp. 252–267, 2000.

[115] K. Yang and X. Wang, "Cross-layer network planning for multi-radio multi-channel cognitive wireless networks," *Communications, IEEE Transactions on*, vol. 56, pp. 1705 –1714, october 2008.

[116] Y. Moon and V. R. Syrotiuk, "A cooperative cdma-based multi-channel mac protocol for mobile ad hoc networks," *Comput. Commun.*, vol. 32, pp. 1810–1819, November 2009.

[117] Z. Kewang, Z. Deyun, and J. Weihua, "Mitigation of exposed terminal problem using packet sensing," in *Communication Networks and Services Research Conference, 2008. CNSR 2008. 6th Annual*, pp. 263 –269, May 2008.

[118] S. Loucif and M. Ould-Khaoua, "On the relative performance merits of hypercube and hypermesh networks," *J. Syst. Archit.*, vol. 46, no. 12, pp. 1103–1114, 2000.

[119] S. Abraham and K. Padmanabhan, "Performance of multicomputer networks under pin-out constraints," *J. Parallel Distrib. Comput.*, vol. 12, no. 3, pp. 237–248, 1991.

[120] P. M. Shankar, *Introduction to Wireless Systems*. John Wiley & Sons, October 2002.