


2007

Channel adaptive real-time medium access control protocols for industrial wireless networks

Kavitha Balasubramanian
Iowa State University

Follow this and additional works at: <https://lib.dr.iastate.edu/rtd>

 Part of the [Computer Sciences Commons](#), [Electrical and Electronics Commons](#), and the [Industrial Engineering Commons](#)

Recommended Citation

Balasubramanian, Kavitha, "Channel adaptive real-time medium access control protocols for industrial wireless networks" (2007). *Retrospective Theses and Dissertations*. 14885.
<https://lib.dr.iastate.edu/rtd/14885>

This Thesis is brought to you for free and open access by the Iowa State University Capstones, Theses and Dissertations at Iowa State University Digital Repository. It has been accepted for inclusion in Retrospective Theses and Dissertations by an authorized administrator of Iowa State University Digital Repository. For more information, please contact digirep@iastate.edu.

**Channel Adaptive Real-Time Medium Access Control protocols for Industrial
Wireless Networks**

by

Kavitha Balasubramanian

A thesis submitted to the graduate faculty
in partial fulfillment of the requirements for the degree of
MASTER OF SCIENCE

Major: Computer Engineering

Program of Study Committee:
Manimaran Govindarasu, Major Professor
Zhengdao Wang
Wensheng Zhang

Iowa State University

Ames, Iowa

2007

Copyright © Kavitha Balasubramanian, 2007. All rights reserved.

UMI Number: 1448692

UMI[®]

UMI Microform 1448692

Copyright 2007 by ProQuest Information and Learning Company.
All rights reserved. This microform edition is protected against
unauthorized copying under Title 17, United States Code.

ProQuest Information and Learning Company
300 North Zeeb Road
P.O. Box 1346
Ann Arbor, MI 48106-1346

DEDICATION

I would like to dedicate this thesis to my family - my mother, Chitra, my sister Archana and her husband Anand without whose support I would never have come this far.

TABLE OF CONTENTS

LIST OF TABLES	vi
LIST OF FIGURES	vii
ACKNOWLEDGEMENTS	ix
ABSTRACT	x
CHAPTER 1. Introduction	1
1.1 Application landscape of Industrial communication	2
1.2 Field Level Communication Message Characteristics and Requirements	3
1.3 Motivation for Wireless Technology	5
1.4 Challenges of Wireless Technology	6
1.5 Existing wired protocol PROFIBUS	7
1.5.1 Link layer services	7
1.5.2 MAC and Link-layer protocol	8
1.6 Problems with PROFIBUS over Wireless	9
1.7 Inadequacies of existing Wireless standards	10
1.8 Motivation for current work	11
CHAPTER 2. REVIEW OF LITERATURE	14
CHAPTER 3. Real-Time MAC protocols for a Single-level Homogeneous Wireless Network	19
3.1 System Model	19
3.1.1 Network Model	19
3.1.2 Channel Model	19

3.2	Basic Framework	20
3.3	Exchange Protocol	23
3.3.1	Basic Idea and Illustrative Example	23
3.3.2	Protocol Details	24
3.4	Simulation studies	30
3.4.1	Effect of the bad state probability (P_{bb})	31
3.4.2	Effect of the good state probability (P_{gg})	32
3.4.3	Effect of the system load (S_l)	33
3.4.4	Effect of the Number of exchange slots (N_e)	34
3.5	Contributions and Future work	35
CHAPTER 4. Real-Time MAC protocols for a Two-level Heterogeneous		
	Wireless Network	36
4.1	System Model	36
4.1.1	Network Model	36
4.1.2	Message Characteristics	37
4.1.3	Channel Model	38
4.2	Objective	39
4.3	Assumptions	40
4.4	Background Information	40
4.4.1	Vertex coloring	40
4.4.2	Exchange Protocol	43
4.4.3	Exchange Setup	44
4.4.4	Conditions for exchange in a parallel message setup	45
4.4.5	Algorithm to construct Exchange set for a message, m_i given the mes- sages that transmit in parallel with m_i	47
4.4.6	Graph Transformation to develop parallel schedule permitting exchanges	48
4.5	Scheduling Algorithms	51
4.5.1	Framework	52

4.5.2	Parallel Protocol	55
4.5.3	Serial Protocol	57
4.5.4	Adaptive Protocol	61
4.5.5	Clairvoyant Protocol	70
4.6	Simulation Studies	70
4.6.1	Effect of varying P_{bb} on the success rate of periodic messages	72
4.6.2	Effect of varying P_{bb} on the success rate of aperiodic messages	73
4.6.3	Effect of varying P_{bb} on the exchange rate of messages	74
4.6.4	Effect of varying P_{bb} on the rate of parallel transmissions	76
4.6.5	Effect of varying P_{gg} on the success rate of periodic messages	77
4.6.6	Effect of varying P_{gg} on the success rate of aperiodic messages	78
4.6.7	Effect of varying P_{gg} on the exchange rate of messages	79
4.6.8	Effect of varying P_{gg} on the rate of parallel transmissions	80
4.7	Contributions and Future work	81
CHAPTER 5. Conclusions and Future Work		83
BIBLIOGRAPHY		85

LIST OF TABLES

Table 1.1	Industrial message Characteristics and Application requirements	5
Table 4.1	Explanation of symbols	41
Table 4.2	Exchange Set for each message for the determination of parallel schedule	49
Table 4.3	P_i that represents the set of messages transmitting in parallel with every message m_i	66
Table 4.4	Exchanged messages and messages that can transmit in parallel with the exchanged message	68

LIST OF FIGURES

Figure 1.1	A typical Industrial automation plant	1
Figure 1.2	Communication Networks within the Industrial Automation Hierarchy	2
Figure 1.3	Classification of Industrial Applications	3
Figure 1.4	Clusters of communication at the field level	4
Figure 1.5	Process Control Application with periodic messages and aperiodic alarms	5
Figure 1.6	PROFIBUS protocol	9
Figure 1.7	Existing wireless protocols	10
Figure 1.8	Comparison between IEEE 802.11, Zigbee and BT protocols	12
Figure 1.9	Gamut of solutions providing real-time guarantees over wireless networks	13
Figure 2.1	Classification of References	14
Figure 3.1	Channel model	20
Figure 3.2	Super Frame with different phases	21
Figure 3.3	Illustrative Example	25
Figure 3.4	Timing diagram	28
Figure 3.5	State Transition diagram	29
Figure 3.6	(a) Effect of P_{bb} (b) Effect of high P_{bb}	31
Figure 3.7	(a) Effect of P_{gg} (b) Effect of high P_{gg}	32
Figure 3.8	Effect of the system load	33
Figure 3.9	(a) Effect of N_e for different M_s (b) Effect of N_e for different N_m .	34
Figure 4.1	Hierarchical Industrial Network Setup	38
Figure 4.2	Channel Model	39

Figure 4.3	Interference and Exchange Sets for inter-cell messages	42
Figure 4.4	Conflict Graph	42
Figure 4.5	Parallel Schedule with all messages in the system	42
Figure 4.6	Parallel Schedule permitting exchanges of Table 4.2	49
Figure 4.7	Parallel Schedule permitting full exchange	50
Figure 4.8	Schedule permitting full exchanges with control slots	51
Figure 4.9	Framework	53
Figure 4.10	Schedule Produced by the Parallel Offline Schedule Construction algorithm	55
Figure 4.11	Schedule Produced by the Serial Offline Schedule Construction algorithm	58
Figure 4.12	Schedule Produced by the Adaptive Offline Schedule Construction algorithm	62
Figure 4.13	Adaptive Run-time schedule followed when no exchange is initiated . .	67
Figure 4.14	Adaptive Run-time schedule followed when exchanges are initiated . .	68
Figure 4.15	Success rate of periodic messages with variation in P_{bb}	72
Figure 4.16	Success rate of inter-cell and intra-cell periodic messages with variation in P_{bb}	73
Figure 4.17	Success rate of aperiodic messages with variation in P_{bb}	74
Figure 4.18	Exchange rate of messages with variation in P_{bb}	74
Figure 4.19	Rate of Inter-cell and Intra-cell exchanges with variation in P_{bb}	75
Figure 4.20	Rate of parallel transmissions with variation in P_{bb}	76
Figure 4.21	Success rate of periodic messages with variation in P_{gg}	77
Figure 4.22	Success rate of Inter-cell and Intra-cell periodic messages with variation in P_{gg}	78
Figure 4.23	Success rate of aperiodic messages with variation in P_{gg}	78
Figure 4.24	Exchange rate of messages with variation in P_{gg}	79
Figure 4.25	Rate of Inter-cell and Intra-cell exchanges with variation in P_{gg}	79
Figure 4.26	Rate of parallel transmissions with variation in P_{gg}	80

ACKNOWLEDGEMENTS

I would like to take this opportunity to express my thanks to those who helped me with various aspects of conducting research and the writing of this thesis. First and foremost, Dr. Manimaran Govindarasu for his guidance, patience, support and motivation throughout the period of my Masters program and helping me realize that I too could do research. I would also like to thank my committee members for their efforts and contributions to this work and being flexible to help me graduate on time: Dr. Zhengdao Wang and Dr. Wensheng Zhang. I thank Anil for his contributions and helping me out throughout my research and for making the lab atmosphere lively. My thanks to Viswanathan for being there always ready to help and Vasanth for all the codes that he wrote and patiently explained. I also would like to express my gratitude towards Kamna for being the best roommate I could have ever had.

The acknowledgement list is way too incomplete without mentioning about all my other friends who made my stay at ISU so totally memorable - the Oaklanders - Vasanth, Puvi, Srikanth, Visu, Atul, Hari and Richard for bearing with me, providing me with food whenever I required and for being as nice as they were, Vatsan for helping me get all the jobs that I actually got and all the advices and suggestions he patiently has given me - every one of them have been valuable, Satyadev who was always ready to help out with acads, Ganesh for assisting me during my initial days at ISU, Rahul for his patient driving lessons; KK, Muthu, Vikram and everyone at the coffee room who patiently have listen to my mindless talks without rebuking. I consider myself as fortunate to have meet all these wonderful people and am glad I came to ISU to pursue my graduate studies.

ABSTRACT

Wireless technology is increasingly finding its way into industrial communication because of the tremendous advantages it is capable of offering. However, the high bit error rate characteristics of wireless channel due to conditions, such as attenuation, noise, channel fading and interference seriously impact the timeliness and reliability guarantee that need to be provided for real-time traffic. Existing wireless protocols either do not adapt well to erroneous channel conditions or do not provide real-time guarantees. The goal of our work is to design and evaluate novel real-time MAC (Medium Access Control) protocols for combined scheduling of periodic and aperiodic messages taking into account the time-varying channel condition.

Our first contribution is the design of a combined scheduling algorithm that exploits both spatial and temporal diversity of the wireless channel through "exchange of slots" among nodes, to effectively mitigate bursty channel error conditions. Simulation results show that the proposed algorithm achieves significant improvements in message success ratio compared to baseline protocols under a wide range of traffic and channel conditions.

The second contribution assumes a two-level hierarchical network in which nodes are grouped into clusters and the communication occurs within each cluster and across clusters. The goal is to maximize the schedulability of intra- and inter-cluster periodic and aperiodic messages with deadline guarantees. In this context, we propose an Adaptive protocol that maximizes the channel utilization by enabling parallel transmissions in a collision-free manner, in conjunction with the use of the slot-exchange technique to actively combat the erroneous channel conditions. Through simulation studies, we show that the proposed Adaptive protocol achieves significant improvement in packet loss performance compared to the baseline protocols that exploit complete parallelism and full exchange, for a wide range of channel conditions.

The future work includes: (i) Formulation of the MAC scheduling problem to a n-level hierarchical network and developing novel scheduling algorithms (ii) Extending the scheduling problem to account for node mobility and developing mobility-aware MAC protocols.

CHAPTER 1. Introduction

The term industrial traffic refers to the transfer of messages in applications such as industrial automation, process control, communication systems in automobiles etc. These complex systems have their functionality split into different sub-systems that communicate with each other giving rise to distributed process control systems. These devices in each of these sub-systems form a hierarchical structure as shown in Figure 1.2. These hierarchical levels have dissimilar message flows, in terms of required response times, amount of information to be transferred, required reliability and message rates (how frequently messages must be transferred). It is known that time constraints are more stringent as we go down in the automation hierarchy.

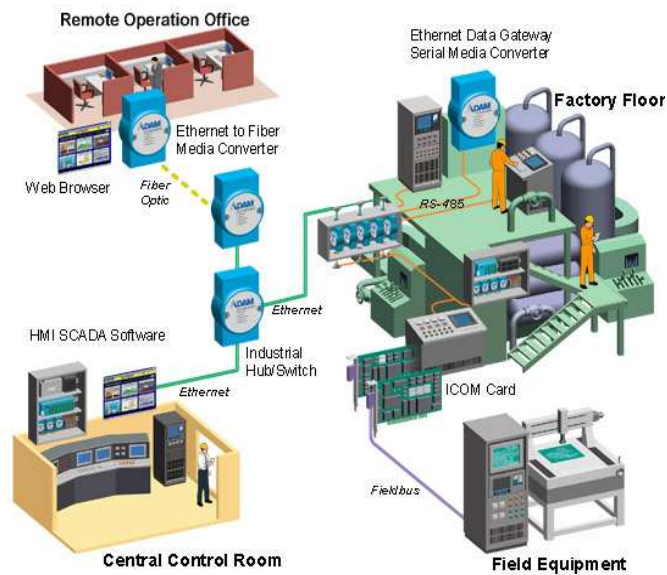


Figure 1.1 A typical Industrial automation plant

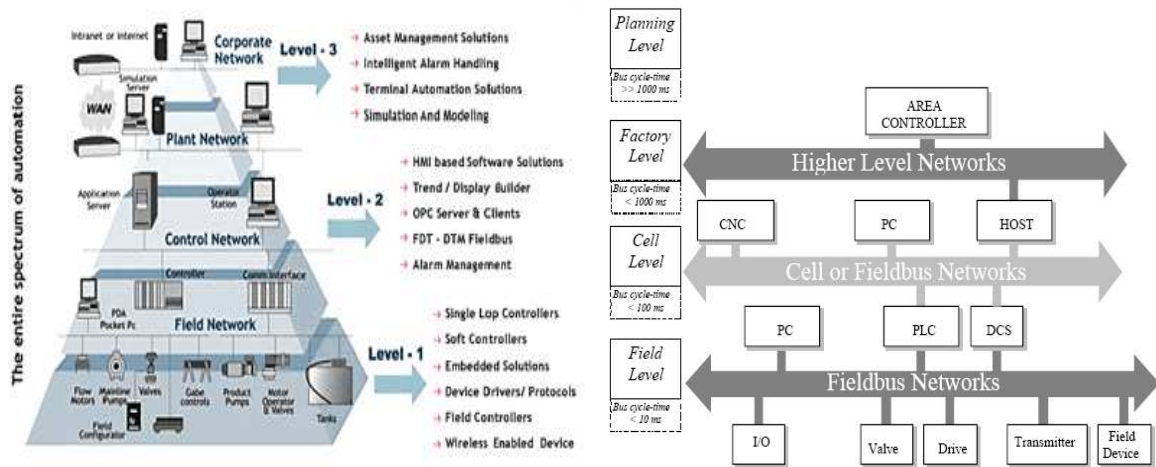


Figure 1.2 Communication Networks within the Industrial Automation Hierarchy

1.1 Application landscape of Industrial communication

Industrial applications can be classified into Non real-time, Soft real-time and Hard real-time [38] based on the nature of nature of timeliness guarantees demanded by the application as shown in Figure 1.3. Representative non real-time applications include remote control, sensor monitoring, system configuring and information exchange. These applications strive for good average-case performance and can tolerate message delays and slow response times. Soft real-time applications like event registration and media applications can tolerate lateness but may respond with decreased service quality. On the other hand, hard real-time industrial applications require a guarantee that all processing and communication be completed within a given time constraint every time. A late response may result in catastrophic consequences and thus, timeliness is a primary measure of correctness for such systems. Common examples of hard real-time industrial applications include communication occurring at the lowest level between field devices such as sensors, actuators and controllers in control loops.

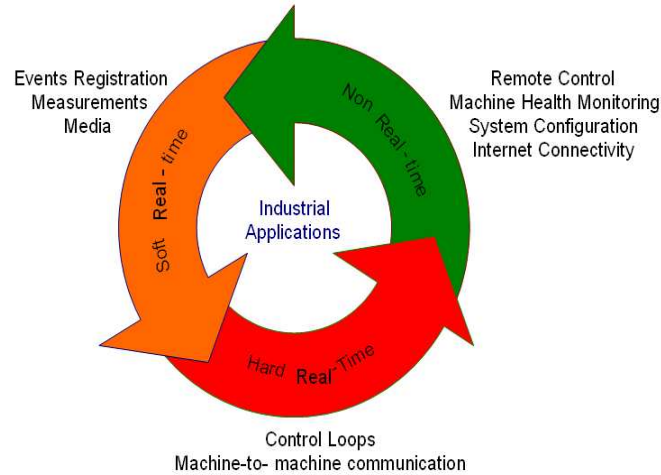


Figure 1.3 Classification of Industrial Applications

1.2 Field Level Communication Message Characteristics and Requirements

Within the industrial communication systems, fieldbus networks are specially intended for the communication between process controllers, sensors and actuators, at the lower levels of the factory automation hierarchy called the field level. In this work, we focus on this field level of communication which must be performed under stringent hard real-time and reliability constraints since missing a deadline can be disastrous. Figure 1.4 shows a factory floor setup where several field devices which are grouped based on their functionality are integrated together through the controller of each sub-system. At the field level, limited amounts of data (few hundreds of bytes) is transferred that broadly follow two message patterns: cyclic data exchange and acyclic data handling. The cyclic data exchange accounts for the periodic polling executed by controllers on the field devices in order to transmit process values, set points etc and the acyclic data handling function allows for the exchange of critical data signaling some exceptional or erroneous condition in the system such as alarms [41]. Both the functions need to be performed within tight timing bounds.

Industrial networks differ significantly from traditional LANs due to special requirements of their applications like the need for hard timing and bandwidth guarantees and supporting

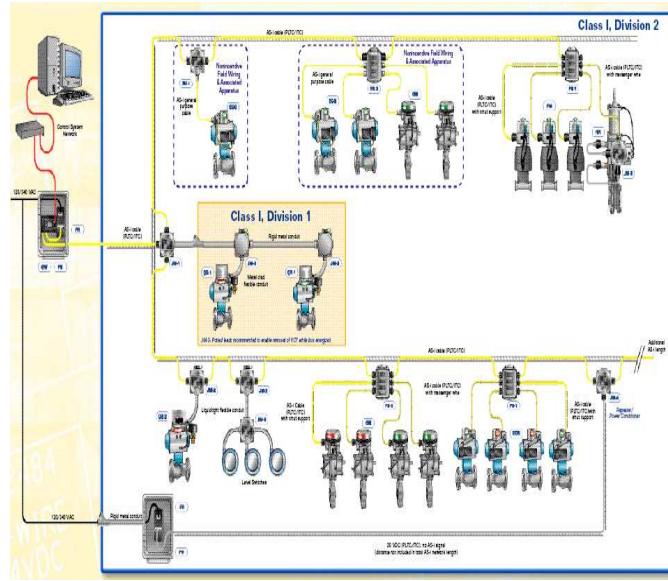


Figure 1.4 Clusters of communication at the field level

priorities. Predictable inter-task communication is extremely critical in such industrial real-time systems because unpredictable delays in the delivery of messages can affect the completion time of the tasks participating in message communication, resulting in deadline misses and eventually performance losses, halts/resets of manufacturing pipelines or defects in products. Some of the application requirements include bounded worst case end to end delay, minimal packet loss and guaranteed maximal jitter.

For guaranteeing this low level of string real-time requirements, fieldbuses are able to support time-critical communication between sensors, actuators, programmable logic controllers and operator workstations. These networks are traditionally based on wired technology and a deterministic Medium access control. The protocols used to handle such high periodicity cyclic and acyclic events are basically of two types Producer-consumer and Master-slave. Producer-consumer protocols are designed for the exchange of a set of identified variables. For each variable, only one producer node exists whereas several nodes may consume it. Fieldbuses using such type of protocols are, for example, WorldFIP [4] and the original IEC fieldbus project, that has been standardized as a technical specification. Master-slave of which typical examples are PROFIBUS-DP and Interbus are based on a point-to-point data exchange implemented

by the action of master devices which poll the slaves by means of suitable communication services. PROFIBUS is a standardized and popular fieldbus system that is designed to meet hard real-time and reliability requirements in harsh communication and electrical conditions. The PROFIBUS uses a token-passing protocol on top of a broadcast medium on the MAC level. In PROFIBUS, stations are organized in a logical ring and communication between distinct stations uses explicit station addresses. Alarm messages are exchanged asynchronously between stations. In such a setting, error detection and re-transmission provide a key mechanism to improve transmission reliability.

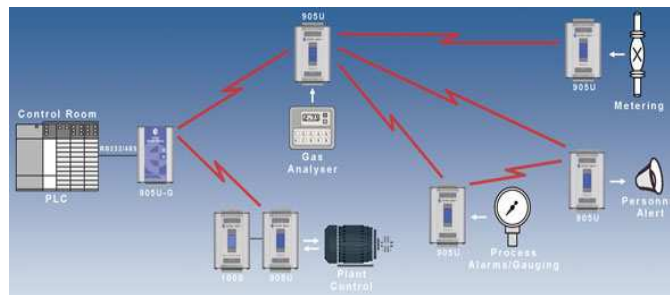


Figure 1.5 Process Control Application with periodic messages and aperiodic alarms

Table 1.1 Industrial message Characteristics and Application requirements

Message Characteristics	Application requirements
Periodic Traffic with deadlines	Meet deadlines
Acyclic packets(alarms) with bounded latency	Predictable and reliable
Short packets(order of kilobytes)	Guaranteed packet delivery and delivery times
	Prioritize messages

1.3 Motivation for Wireless Technology

The current wired infrastructure used for industrial communication is plagued by problems of limited mobility, high deployment and maintenance cost and constrains the viability of any smart real-time system. Wireline implementations are not cost effective for large temporary production lines and may not be feasible for systems including rotating or high mobility machinery such as measurement and control of moving objects. The wireless evolution offers

numerous benefits for industrial applications, where wired solutions have prohibitive problems in terms of cost and feasibility. Hence wireless technology becomes an attractive option providing several advantages such as cabling cost reduction, installation of equipments in hazardous areas (in presence of aggressive chemicals capable of damaging cables), ease of operation and flexibility to perform rapid plan reconfigurations. Another important benefit is the potential that wireless technology offers for truly mobile stations that help in early fault diagnosis [34] and unmanned factory automation facilities using robots and automatic guided vehicles for increased and precision production. These well recognized benefits coupled with the growing popularity of wireless communication in numerous fields including home and office environments has led to its increased dependability, performance improvement and cost reduction. Thus, because of the widespread standardization of the technology, it is very likely in the near future, there will be a proliferation of wireless implementations of factory communication systems. This has motivated the strong research into the use of wireless medium for real-time industrial applications.

1.4 Challenges of Wireless Technology

In spite of having such clear benefits, wireless technology has its own drawbacks arising due to the unreliable characteristics of the wireless medium which makes it, in its current state, unsuitable for supporting real-time communication. Wireless links are more error prone than their wired counterparts due to noisy channel conditions that vary with time. Due to heavy obstruction, the wireless medium of industrial environments are known to suffer more serious large-scale path loss and fading than other indoor environments [44]. Occurrences of outages lasting for several seconds, during which no packet transmission over a channel is not uncommon and there exists a large variability in the distributions of length of error bursts and error free periods [46] making the channel behavior unpredictable. In addition, distance dependent path loss and co-/adjacent channel interference influence the channel. Hence the wave propagation environment (number of propagation paths, their respective losses) and its time varying nature (moving people, moving machines and metal surfaces) play a dominant role

in constituting channel characteristics [50]. The high error rates over wireless links occur due to different phenomena such as multipath fading, path loss, electromagnetic noise, interfering components from machines, metallic objects, civil engineering structures in the factory floor and man-made interference from remote controls, ovens etc that cause bit errors(single or multiple bits in a packets data part change their value) and packet losses(failure to acquire bit synchronization during a packets preamble) that tend to occur in bursts [49]. Hence, measures to substantially improve transmission reliability overcoming the above mentioned challenges need to be developed so that real-time and reliability requirements demanded by the industrial applications can be guaranteed.

In what follows, we include a description providing a detailed insight into the specifics of the PROFIBUS protocol that is instrumental in understanding why the existing wired protocol currently being used for providing real-time guarantees cannot be directly applied to the wireless setup. We then outline other related work in this area and show how none of the MAC protocols proactively adapt to the erroneous channel condition. Using this as the motivation, we develop a new and robust MAC protocol that enables maximum channel utilization while coping with the strong variations in the channel conditions of the industrial environment that significantly improves the success rate of messages over the wireless medium for providing real-time guarantees.

1.5 Existing wired protocol PROFIBUS

The PROFIBUS is a well known, standardized and widely used fieldbus system. The PROFIBUS standard covers the application layer, the MAC and data link layer and the PHY layer. On the PHY, often RS485 is employed and the MAC layer implements a token passing protocol on top of a broadcast medium.

1.5.1 Link layer services

The PROFIBUS MAC- and link-layer protocol offers its services via the link-layer interface. There is one unacknowledged and three acknowledged services, in two of the acknowledged

services, the acknowledgement may carry some data. The acknowledged services are semi-reliable i.e. a configurable number of retransmissions is performed, before the protocol gives up. The services are invoked by the upper layers using appropriate request primitives the link-layer interface offers. Once the outcome of a request is known (for the acknowledged services: an ACK was received or the number of trials is exhausted) then the link-layer protocol generates a confirm primitive for the upper layers, indicating the success of the request. Each request is marked with one of two possible priority values. High priority requests are typically used for safety-critical and urgent messages (e.g. alarms), while low priority requests are used for everything else, including cyclic data transmission and file transfers.

1.5.2 MAC and Link-layer protocol

The combined MAC and link layer is called fieldbus data link (FDL) layer. On the MAC level, two approaches are combined: a request/answer type of protocol for performing data exchange, and a token-passing protocol on top of a broadcast channel for arbitrating the right to initiate data or management transmissions as shown in Figure 1.6. There are two types of stations: the active stations are capable of participating in the token passing protocol, the passive stations are not. An active station is only allowed to transmit data, if it owns the token. The token is passed along a logical ring, which is formed by ascending station addresses. Station addresses are in the range of 0 to 126. Only ring members get the token and are allowed to initiate data transmissions. Every ring member carries out certain procedures for ring maintenance: detection and repair of lost tokens, including new stations into the ring, asking remote stations for their status, etc. Upon every token arrival, a station computes its token holding time according to a modified version of the timed-token protocol: a station subtracts its measured token rotation time from the configured target token rotation time T_{TTRT} . If the difference is greater than zero, then the station is allowed to initiate data transfers. However, even in the case of late tokens (difference smaller than zero) it is allowed to handle at least one high priority service request. If there is afterwards still token holding time available, the station handles the high priority requests first, and then proceeds with low priority request. A service

request is processed according to a variant of the alternating bit protocol (ABP), with one separate protocol instance per target station. The number of retransmissions is bounded; the bound is a fixed parameter *max_retry*. When the token holding time expires, the token-owner is required to pass the token to the next active station.



Fig. 5: PROFIBUS configuration with active masters and slaves

Figure 1.6 PROFIBUS protocol

Passing the token from station a to station b involves transmitting a token frame. After a has sent this frame, it listens for a short time (called slot time T_{SL}) on the medium, whether there is some activity (a infers from this that b has accepted the token). If there is no activity, the token frame is retransmitted. After three unsuccessful trials, a determines b's logical successor in the ring (say, c) and tries to pass the token to c, and b is lost from the ring. It is a's responsibility to re-include b later on, but this may take some time (depending on certain protocol parameters and the current load).

New stations are included by stations already present in the ring: a ring member polls approximately every $gap_factor * T_{TTRT}$ seconds the address space between its own address and that of its logical ring successor. If a new station is found, it is included into the ring.

1.6 Problems with PROFIBUS over Wireless

There exist several problems in applying the PROFIBUS MAC protocol directly over the wireless medium. The concept of permanently relying on token frames for the purpose of ring maintenance is vulnerable to channel errors [7]. Specifically, the membership of a station in

the logical ring is harmed by the loss of token frames. Ring membership is important since only ring members are allowed to transmit data. Multiple subsequent loss of token frames can throw stations out of the logical ring or even destroy it totally. If a station is thrown out of the ring, it has to be re-included which is time-consuming and during the re-inclusion process, the lost station has no chance to initiate any transmission, however time-critical its data might be. These problems are serious and harm the real-time requirement of the application. Hence for the creation of a wireless fieldbus system, alternate MAC and link-layer protocols need to be investigated.

1.7 Inadequacies of existing Wireless standards

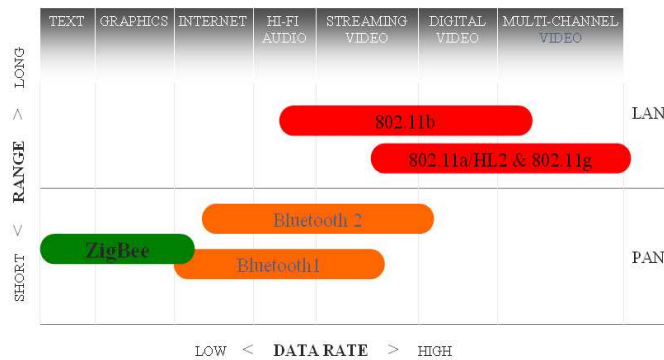


Figure 1.7 Existing wireless protocols

There exists a number of different standards for wireless communication which are shown in Figure 1.7. Among the most important ones, we find the wireless LAN IEEE 802.11 as the oldest and most mature standard. The original version of the standard IEEE 802.11 specifies two raw data rates of 1 and 2 megabits per second (Mbit/s) to be transmitted via infrared (IR) signals or by either Frequency hopping or Direct-sequence spread spectrum in the Industrial Scientific Medical frequency band at 2.4 GHz. The standard was initially designed for best-effort data traffic but a set of Quality of Service extension to the base standard for LAN applications called the IEEE 802.11e exists that enhances the IEEE 802.11 CSMA/CA MAC protocol. The IEEE 802.11e standard is considered of critical importance for delay-

sensitive soft real-time applications, such as Voice over Wireless IP and Streaming Multimedia. The IEEE 802.11 standard and its extensions are not suitable for hard real-time wireless industrial network since its media access control method is contention based that exhibits unstable performance under heavy traffic and produces unbounded delay distributions [40]. The Bluetooth standard is intended to be a low-complexity and low-cost wireless solution for short-range wireless communication and is evolving towards the support of soft real-time applications like multimedia. It is designed for cable replacement and ad hoc connection of different consumer devices and does not include real-time deadline oriented support for applications. The ZigBee specification is designed for a suite of high level communication protocols using small, low-power digital radios based on the IEEE 802.15.4 standard for wireless personal area networks (WPANs). ZigBee, is targeted at RF applications that require a low data rate, long battery life, and secure networking. ZigBee's current focus is to define a general-purpose, inexpensive, self-organizing, mesh network that can be used for industrial control, embedded sensing, medical data collection, smoke and intruder warning, building automation, home automation etc. ZigBee protocols are intended for use in embedded applications requiring low data rates and low power consumption. Because of the low data rates and inadequate real-time support, the Zigbee protocol is not suitable for real-time industrial applications. These findings are summarized in Figure 1.8.

1.8 Motivation for current work

Since wireless networks are substantially different from their wired counterparts with respect to the channel conditions, technologies developed for wired networks cannot be directly adopted. In most wired network models for real-time systems, the communication links are assumed to have a fixed capacity over time. This assumption may be invalid in wireless environments, where link capacities can be temporarily degraded due to fading, attenuation, and path blockage [50]. We also see that wired protocols such as the PROFIBUS generally used for communication in an industrial setup cannot be directly used over the wireless medium. In addition, existing wireless standards such as IEEE 802.11, IEEE 802.15.1 (Bluetooth) and IEEE

Feature	802.11	Bluetooth	Zigbee
Frequency band of operation	802.11a – 5 GHz ISM 802.11b – 2.45 GHz ISM	Unlicensed 2.4 GHz	2.4 GHz ISM 866 MHz ISM – Europe 915 MHz ISM – N. America
Data rate	802.11a – 1-2 Mbps 802.11b – 11 Mbps 802.11g – 54 Mbps	1 Mbps	2.4 GHz – 250 Mbps 915 MHz – 40 Kbps 866 MHz – 20 Kbps
Range	30m at 11 Mbps 90m at 1Mbps With extended antenna 8km	Power dependent (10 cm, 10 m, 100m)	10-20 m
Central Coordinator	Access Point	Master	None
Interference from other devices	--	Avoided using frequency hopping	Dynamic channel selection possible
Optimized for	Multimedia, TCP/IP and high data rate applications	Cable replacement technology for portable and fixed electronic devices.	Low power low cost networking in residential and industrial environment. Optimized for time critical applications
Energy Consumption	High	Low (Large packets over small networks)	Least (Small packets over large networks)
Voice support/Security	Yes/Yes	Yes/Yes	No/Yes
Type of Network / Channel Access	Mobile / CSMA/CA and polling	Mobile & Static / Polling	Mostly static with infrequently used devices / CSMA and slotted CSMA/CA
Real-Time deadlines	?	?	?

Figure 1.8 Comparison between IEEE 802.11, Zigbee and BT protocols

802.15.4 (Zigbee) are primarily designed to meet non real-time or soft real-time application requirements and provide inadequate guarantees for hard real-time industrial applications.

Several levels of solutions have been proposed that addresses the problem of providing real-time guarantees over wireless networks. Some of these have been summarized in Figure 1.9. We elaborate on each of these techniques and other related works in this area in Chapter 2 and explain how existing works do not enable full channel utilization and inadequately adapt to the bursty channel error conditions. Hence, there arises a need to design and develop special MAC protocols and techniques which take both the channel characteristics and the hard real-time requirements of the messages into account while maximizing the channel utilization.

Hence, in this work, we develop a real-time MAC protocol for dependable real-time communication over an unreliable wireless communication channel. Since the existing protocols are not designed for a lossy link, the MAC scheme that we develop takes explicitly into account the bursty error characteristics of the wireless channel. The protocol is designed to exploit both the spatial and temporal diversity characteristics of the channel while maximizing the channel utilization by performing simultaneous parallel transmission whenever possible. By

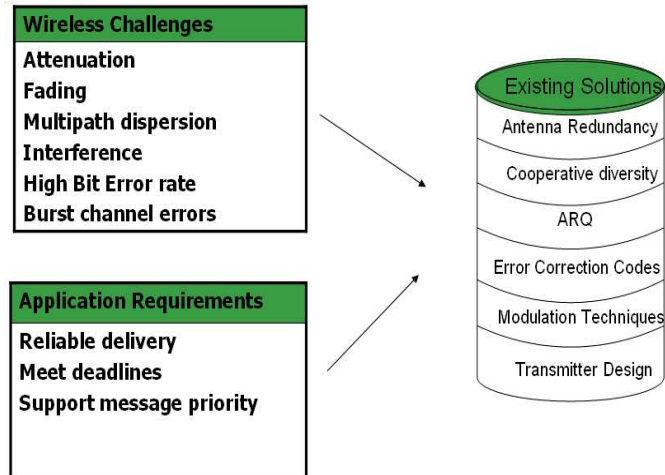


Figure 1.9 Gamut of solutions providing real-time guarantees over wireless networks

means of simulations, we show that these protocol characteristics improve the number of messages that are successfully delivered to the destination within the deadline and helps meeting the real-time and reliability requirements of the application.

CHAPTER 2. REVIEW OF LITERATURE

We group the related work available in literature which focus on achieving predictable and guaranteed delivery of real-time messages over an industrial network into different categories as illustrated in the Figure 2.1. Some works grouped under the wired real-time category focus on improving the performance of protocols over a wired industrial network while works in the wireless domain study the feasibility of achieving real-time performances over a wireless network. There is also a significant amount of literature on achieving real-time guarantees on a network consisting of existing industrial wired networks and new wireless infrastructure that belong to the Hybrid wired/wireless category.

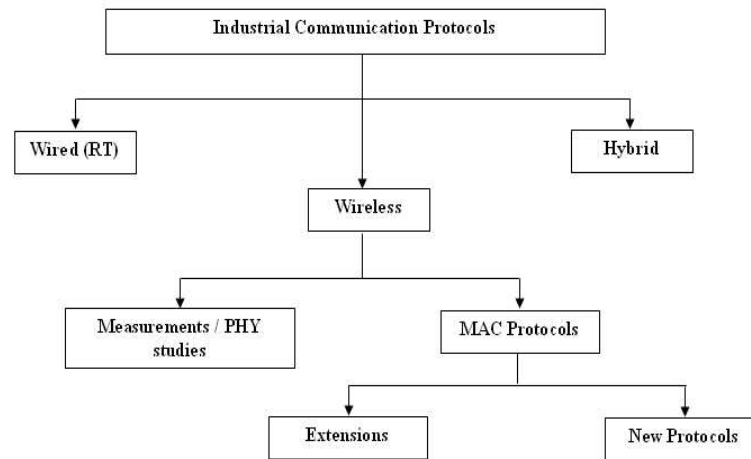


Figure 2.1 Classification of References

In the wired real time category, there exists several variants of master-slave protocols [11, 42, 4, 10, 1, 2] that are currently being used for providing hard real-time guarantees over wired networks in an industrial environments. Other works in this group are directed towards assigning priorities and improving delivery times by the use of wired protocols like Ethernet

[12, 25] for industrial communication.

In the wireless domain, a number of measurement studies [50, 15, 16, 33] have been performed to study the capability of the wireless channel to support real-time communications and these reveal the time-variable and high error rates of the wireless channel. In [50], the authors study the statistical properties of the bit error patterns delivered by a wireless link in an industrial environment. The result of their findings and several other mentioned above indicate the variability of the wireless link due to frequently changing environmental conditions like moving people, portal crane activity, moving parts of a machine etc. Some qualitative results include : time varying behavior, burstiness properties, and orders of magnitude packet losses, large variability of error free burst length distributions for both packet losses and bit errors and the tendency of packet losses and bit errors to occur in bursts. The author establishes that popular Gilbert Elliot model is a useful tool for simulating bit errors on a wireless link which we use in the present work.

Several works in literature have focused on propositions for a MAC protocol for achieving predictable and guaranteed message delivery over a wireless industrial network. While some propose new techniques over the wireless network for improved performance, other explore the applicability of existing standardized protocols such as 802.11 and BT or extend the wired real-time protocols such as PROFIBUS for the wireless environment. In the later category, works in [31, 49, 48] which propose and analyze the use of master-slave based polling protocols over a wireless network are significant. In [31], the authors explore the use of IEEE 802.11 for industrial communication by analyzing the possibility of implementing protocols based on master-slave architecture of traditional field buses on a IEEE 802.11 PHY. They propose a prototype of a Master-slave protocol using the IEEE 802.11 communication primitives that uses a simple polling scheme for the exchange of cyclic data and which consider three different techniques for handling acyclic requests. In a similar approach, in [40], the authors present a four-layer architecture based on the NDIS specification and a virtual polling algorithm to circumvent the unstable performance of the IEEE 802.11b wireless LAN for real-time wireless industrial networks. In [49], the adaptive-intervals MAC protocol has been proposed that uses

a polling-based approach combined with group testing feature for improving the delay in low load conditions. However, in [7, 22], the authors show the serious stability issues due to the loss of token frames leading to unsatisfying real-time performance arising from running the existing PROFIBUS MAC and link-layer token-passing master-slave protocols over an 802.11 DSSS PHY. Works such as [39] where the authors propose schemes to support combined periodic and aperiodic traffic over Bluetooth networks, [44] where the authors propose the use of DSSS CDMA technology to build Industrial Control WLAN for enhanced robustness, and [32, 35, 30] which propose extensions for a wireless fieldbus network also fall into the category of using existing standardized protocols for real-time communication over wireless networks.

The use of existing wireless technology such as IEEE 802.11, BT and their variants for industrial networks has been extensively researched. The conditions under which 802.11g offers suitable performance to support time-critical data traffic has been analyzed in [34]. In [21], an analytical model for the evaluation of IEEE 802.11e has been designed. In [14], an approach to use the PHY and DLL of existing WLAN and WPAN networks with the application layer of the wired fieldbus has been outlined. A frequency hopping wireless system for sensor/actuator communication for real-time factory-automation that uses an optimized TDMA protocol has been proposed in [13]. Notable works in the BT area are [29] in which a slave-to-slave communication architecture configuration has been proposed to improve the protocol performance. However, the acceptability of a slave-to-slave communication approach is still uncertain. In [43, 24], several practical experiments for the performance evaluation of BT have been performed and the authors conclude that adaptive error correction protocols will increase the applicability of BT. Other works in this area [37, 3] measure the mutual interference caused by other co-located systems and a probabilistic upper-bound on packet error probability has been derived. In [36, 23], architectures for wireless industrial monitoring systems but which offer no real-time guarantees have been proposed. The drawback of all of these works is that none of them include active techniques to combat the variations in channel conditions while providing real-time guarantees.

Several new schemes are being proposed to improve the reliability over wireless links to

combat the issues faced by standardized wireless protocols and to overcome the high error characteristics of the wireless channel. In [47], three modifications to the ARQ protocol are introduced usage of additional error correction codes during re-transmission, transmission of multiple copies of the same packet (multicopy ARQ) during the first transmission attempt and antenna redundancy. In [46], the author introduce the concept of antenna redundancy that uses alternate spatial channels for re-transmission with the use of multiple antennas. However, the ARQ schemes proposed don't work well at high error rates and Antenna redundancy requires additional hardware in all communicating devices if any-to-any communication need to be implemented in addition to having a central Base Station setup. In [8] and [45], the authors present the techniques that make use of the wireless channel conditions while making packet dispatching decisions in a wireless LAN. In [8], the authors propose a channel state dependent protocol that uses a priority queue based scheduling mechanism over CSMA/CA to avoid the head of line blocking problem associated with multiple sessions sharing the wireless link to improve the performances of transport layer sessions. However, the traffic considered is best-effort and only a centralized protocol working on a Base Station setup has been proposed. In [45], when the channel is under-utilized, based on the channel state estimation, every packet is given additional slots and the packet scheduler is improved to transmit packet only when the channel condition is good. The efficiency of the proposed scheme depends on how exactly the channel state is determined and on the error characteristics of the real wireless channel. Also, accurate estimation techniques that predict the exact future channel state is unfeasible. In [18], the authors take a coding theory approach of combining different coding and decoding methods with the ARQ scheme. They employ an incremental redundancy retransmission scheme in conjunction with concatenated coding. A similar approach is taken in [27] for a weakly hard (m,k) based system. However, if packets over a channel fail continuously, employing coding strategies may not be effective since In these scenarios, most of the erroneous data bits occur within a single data packet [6] that makes compensation by the use of forward correcting codes difficult.

Another direction in which research has been carried out for the initial migrating of existing

wired fieldbus devices into the wireless domain is the design of hybrid-wired wireless protocols that focus on architectures and protocols for communication between the two domains using repeaters, bridges and routers have been proposed in [9,13,14,21,31,33]. Hybrid wired/wireless systems have also been considered in [5, 28, 19] which we do not elaborate on these literatures in our current work.

While many of the cited works offer probabilistic real-time performances, worst-case channel conditions have been inadequately analyzed and none of them dynamically adapt to the channel conditions as well as maximize channel utilization by performing parallel transmissions that can offer significant performance gains for real-time data transmission in an industrial network. This forms the basis of our work. The rest of the thesis is organized as follows. In Chapter 3, we propose a framework for combined scheduling of periodic and aperiodic messages over a wireless environment. We propose a MAC-layer exchange protocol that overcome the bursty channel error conditions by exploiting the spatial and temporal diversity characteristics of the wireless medium and maximizes the number of real-time messages meeting their deadlines. In Chapter 4, we extend the framework to a 2-level hierarchical industrial network setup and show the trade-offs between using the exchange protocol and enabling parallel transmissions. In this context, we develop a distributed adaptive protocol that performs exchanges as well as enables parallel transmissions based on the channel conditions while maintaining a collision-free environment. We show by simulations the significant performance gains that this scheme offers in improving the success rate of real-time periodic and aperiodic messages. In Chapter 4, we conclude presenting directions for future work in this area.

CHAPTER 3. Real-Time MAC protocols for a Single-level Homogeneous Wireless Network

We study the problem of effectively scheduling real-time messages, both periodic and aperiodic, over a single-hop wireless medium in an industrial setup. In this context, we present a novel MAC protocol that would dynamically adapt to the erroneous channel conditions thereby minimizing the number of deadline misses. To corroborate the performance enhancements of our proposition, we simulate the setup and evaluate the protocol for different channel and load conditions.

3.1 System Model

3.1.1 Network Model

We study a single-hop industrial environment consisting predominantly of periodic message with occasional aperiodic messages/alarms being generated due to faulty or abnormal outcome of some process which require higher priority. Both messages need to be provided with guarantees on delivery times. The communication medium is wireless characterized by high bit error rate due to phenomenon like noise, attenuation, fading and interference. We assume that messages destination is a node in the single hop.

3.1.2 Channel Model

The bursty error characteristics of the wireless environment in a typical industrial setup can be captured by the Discrete-Time Gilbert-Elliot Channel Model [50, 17, 20]. Time in the super-frame is divided into slots and the model works with slotted time where the state transitions happen at the end of each slot. The state space of the Gilbert-Elliot model contains

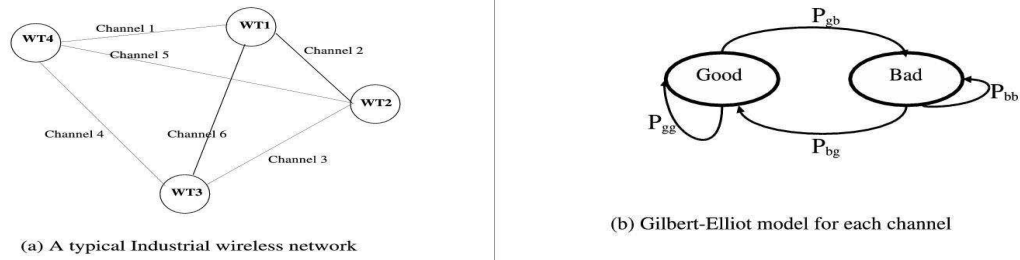


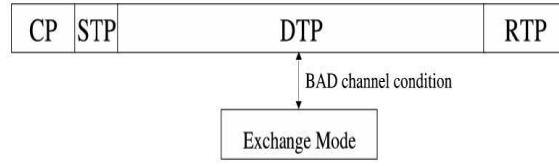
Figure 3.1 Channel model

the following two states: GOOD and BAD. When in the GOOD state, no bit errors occur in the data unit sent in the corresponding slot. Hence the transmission succeeds when done in an exclusive manner. On the other hand, when the channel is in BAD state certain bit errors occur in the received data unit and the data transmission is considered erroneous. Figure 3.1.(b) shows the state diagram along with the transition probabilities. We assume that each channel between a given source-destination pair is statistically independent. In fig 3.1.(a), each solid line between two wireless terminals represent an independent channel over which the Gilbert-Elliot channel model is applied.

3.2 Basic Framework

The medium is shared by all the wireless terminals and transmissions follow a super-frame structure that repeats itself. The super-frame is divided into slots and each message would occupy several slots . In a slot, a sender is able to transmit a unit of the message and receive the corresponding an acknowledgment. Even if a few bits of the unit of message that occupy a slot are corrupted, the destination does not send an ACK confirming that the channel is in bad state and the unit is marked for re-transmission. For simplicity, we do not consider any coding schemes to correct transmission errors.

The basic framework consists of a centralized scheduler that collects all the messages available in the system before every super-frame. The scheduler then prepares a schedule that is followed by all nodes in the system. To facilitate such an approach, every super-frame is divided into 4 phases: the Control phase (CP), schedule transmission phase (STP), data transmission



CP: Control Phase

STP: Schedule Transmission Phase

DTP: Data Transmission Phase

RTP: Re-Transmission Phase

Figure 3.2 Super Frame with different phases

phase (DTP) and re-transmission phase (RTP), each of fixed duration as shown in figure 3.2. The CP and RTP operate in the contention mode while the DTP functions in a contention free mode. We assume that all the messages that need to be transmitted during the data phase become ready at the beginning of this phase and every message needs to complete before the end of the super-frame.

- **Control Phase(CP):** In the control phase, all the messages in the system are sent to the central scheduler which performs an admission test and constructs a non-overlapping transmission schedule for the admitted messages. The admission test works iteratively on the set of messages received by the central scheduler and checks if the super-frame has enough free slots to accommodate the next message and its recurring instances (for periodic messages only) before its deadline. Consider a periodic message of size M_i occupying N_i slots of the super-frame . Let N_{data} denotes the number of slots of the data transmission phase, $N_{admitted}$ denote the number of slots of the super-frame occupied by already admitted messages; $N_{transfer}$ denote the number of transfer slots and N_{exchg} denote the number of exchange slots (more details about the usage of these slots would be explain in Section 3.3). The admission test checks if

$$N_i \leq N_{data} - N_{admitted} - N_{transfer} - N_{exchg} \quad (3.1)$$

If the above condition is satisfied, the message is admitted to the system and the scheduler reserves N_i slots exclusively for the message; else the message is rejected. However,

aperiodic messages are always admitted into the system by removing an instance of the periodic message, since they require higher priority.

- **Schedule Transmission Phase(STP):** In the Schedule transmission phase, the central scheduler broadcasts the above constructed schedule to all the terminals in the network.
- **Data Transmission Phase(DTP):** In this phase, each wireless terminal begins its transmission in its scheduled slot. In spite of allocating enough time slots in an exclusive manner, not all messages will reach the destination without errors because of the erroneous channel condition. Therefore, some messages might miss the deadline. The number of deadline misses will depend on the exact data transmission protocol. We present two basic schemes here which would be used for transmitting messages in this phase. The main differentiation between the two schemes arise in the mechanism they employ for taking advantage of any unutilized slots available in case of an underloaded channel. However, our main contribution is the Exchange protocol which we present in Section 3.3.

In Time Division Multiplexing with Variable number of Retransmissions (**TDMVR**), in case of an underloaded channel, all the unutilized slots towards the end of the super frame are used for re-transmission. However, in Time Division Multiplexing with Constant number of Retransmissions (**TDMCR**), the schedule is formed in such a way that all the unutilized slots are equally distributed between the transmitting terminals. Hence each transmitting terminal is allotted additional channel resources to enable successful transmission of the message. Although these scheme enable full utilization of the channel in case of few messages being in the system by increasing the attempts available for existing message transmissions, they do not adapt to the bursty error conditions of the channel. The exchange protocol presented in the next section adapts to the channel conditions thereby decreasing the number of deadline misses and increasing the effective system utilization.

- **Re-Transmission phase(RTP):** All wireless terminals which could not successfully

transmit all their messages during the DTP in the first attempt contend for channel access (CSMA) and employ a backoff algorithm on collision. Approximately 10 % of the slots in DTP is allocated for Re-transmission due to the channel error conditions typical of the wireless environment. At the end of the superframe, the slots that were unable to be successfully transmitted are declared as deadline misses.

3.3 Exchange Protocol

We now present the Exchange protocol that comes into effect during the DTP. The exchange protocol dynamically adapts to adverse channel conditions and enables effective scheduling of real-time messages in addition to preserving the schedulability guarantees provided to existing messages. Schedulability guarantee implies the fact that when a message is admitted into the system, it is given a certain number of slots (as is occupied by the message) exclusively for data transmission. The scheme caches on two characteristic features - spatial and temporal diversity of the wireless channel; temporal diversity signifies the fact that when a channel is the bad state, it would eventually move to the good state and spatial diversity indicates the condition that if one channel is in bad state, it is possible that a different channel would be in good state.

3.3.1 Basic Idea and Illustrative Example

During the DTP, each wireless terminal begins its transmission in its scheduled slot. When a channel between a source destination pair is bad, transmissions begin to fail. During this state, the Exchange protocol is used that works around the occurrences of error bursts. The primary intuition behind the scheme is to postpone the transmission on a channel in a bad state to a later time and schedule transmissions on a channel in a good state with the hope that the channel in the bad state would change into good state in the meantime. This protocol forms its basis on the wireless channel characteristic of correlated packet losses i.e. on a channel which is erroneous, a single packet loss would be followed by back-to-back packet losses. Hence the exchange protocol takes advantage of this characteristic feature to perform efficient scheduling

of real-time messages.

The following illustrative example demonstrates the exchange protocol mechanism:

Consider a simple network with three wireless terminals shown in figure 3.3.(a). Let the channels between them be C12, C23 and C31 and let the messages that need to be transmitted be: 12, 23 and 13 where the first number indicates the source and second number indicates the destination. Figure 3.3.(b) shows the channel condition variation with time. The shaded slots indicate that the channel is in bad state. The original schedule given by the central scheduler is shown in figure 3.3.(c) and the schedule of the basic schemes is given in figure 3.3.(d) which would lead to 6 slots being unsuccessful.

In the exchange protocol, once a terminal (exchange initiator) notices that its channel to the destination is in bad state it exchanges its slots (as many as possible) with a different terminal(exchanged sender). As a result of the exchange, the exchange initiator performs its transmissions in the slots of the exchanged sender and vice versa. This basic idea is depicted in figure 3.3.(e), where the exchange initiator, node 2 exchanges its 6 slots with the exchanged sender 1. The final schedule due to the exchange is shown in figure 3.3.(f) where only 1 slot is unsuccessful.

Several different heuristics can be applied for a choice of the exchanged sender; based on channel correlation, estimation of the burst length and priority. In this paper, for simplicity, we use the closest node id terminal which has a message to transmit for exchange.

3.3.2 Protocol Details

As highlighted in the illustrative example above, the basic idea of the exchange protocol is to avoid transmissions on a channel in the bad state by passing control to a different transmitter-receiver pair whose channel is in good state. In order to enable this preserving the schedulability guarantee, the exchange protocol incurs some control overhead.

Consider an exchange initiator (terminal 2 in figure 3.3.(e)) which detects its channel is in the bad state (slot 5) during data transmission. It would initiate the exchange protocol and try to exchange its remaining slots with some exchanged sender. In order to communicate to

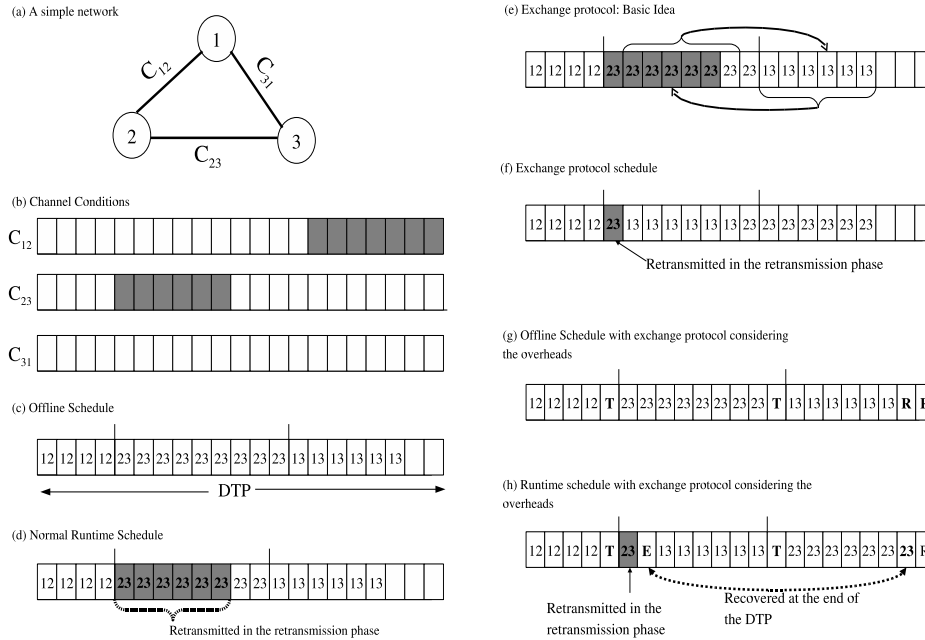


Figure 3.3 Illustrative Example

the chosen exchanged sender (terminal 1 in 3.3.(e)) of its intension to exchange its remaining slots, a slot called the exchange slot (slot 7 indicated as E in 3.3.(h)) is used in which a two way handshake is performed. The exchange initiator sends an exchange request (N_{req}) along with the maximum slots it want to exchange which is typically till the end of its data transmission phase and the exchanged sender replies with an ACK that denotes the actual number of slots it has available for exchange ($N_{available}$). In the example, $N_{exchg} = N_{available} = 6$. Both the exchange request and exchange ACK are assumed to be small enough and can be accommodated in one single slot. Four cases now arise.

- If $N_{req} = N_{available}$ (refer to the example in 3.3), then the exchanged sender takes over the current transmission for N_{req} slots (slots 8-13 in figure 3.3.(h)) and the exchange initiator transmits during the scheduled transmission time of the exchanged sender for N_{req} slots (slots 15-20 in figure 3.3.(h)).
- If $N_{req} < N_{available}$, the exchanged sender takes over the current transmission and transmits to its receiver for N_{req} slots. During the scheduled transmission of the exchanged

sender, the exchanged sender transmits for $N_{available} - N_{req}$ slots and the exchange initiator transmits for N_{req} slots.

- If $N_{req} > N_{available}$, the exchanged sender takes over the current transmission for $N_{available}$ slots after which the exchange initiator tries to transmit to its destination. If the channel is still erroneous, it may try to perform another exchange. During the scheduled transmission of the exchanged sender, the exchange initiator transmits for $N_{available}$ slots.
- If $N_{available} \leq 1$ (this condition arises if the exchanged sender has already bartered all its slots), the exchange initiator looks for a different exchanged sender to exchange with. When $N_{available} = 1$, the exchange is not performed to maintain the message boundaries.

Since for every exchange initiated, an exchange slot is being consumed, the number of exchanges that can be performed is limited to N_{exchg} in every super-frame. The scheduler broadcasts the N_{exchg} value to all nodes during the STP. To compensate for the exchange slot (to maintain the schedulability guarantee) which are being used by the exchange initiator from the scheduled slots that it has been allocated for transmission, N_{exchg} number of slots are reserved at the end of the super-frame (indicated by R in figure 3.3.(g)). From this pool of reserved slots, every exchange initiator exclusively gets a slot for every exchange it has performed. To enable these functions, an exchange counter is maintained that denotes the number of exchanges that has been performed in the super-frame until the current time. This exchange counter is passed on between the transmitting terminals by means of the transfer slot (indicated by T in figure 3.3.g) occurring at the end of every message transmission. So at the beginning of the transmission, each node knows how many more exchanges can be performed. Each time an exchange is performed, the exchange counter is decremented by the exchange initiator and the value of the exchange counter is passed onto the exchange sender in the exchange slot. In this way, the exchanged sender knows how many more exchanges it can perform during the exchange period. After its exchange period, it passes on the value of the exchange counter to the next transmitting station in the transfer slot. If the exchange counter becomes 0, no more exchanges are performed. The exchange and transfer slots are transmitted at high power level

so that they are received correctly. If any of the transfer or exchange slots are completed the exchange counter is reset to zero and the transmissions proceed as per the offline schedule.

Let N_{ctr} denote the current value of the exchange counter, when a node uses up an exchange slot for performing exchange, it decrements the exchange counter to $N_{ctr} - 1$ and $N_{exchg} - N_{ctr}$ th slot is used by this exchange initiator from the reserved slots. In the above example, assume that $N_{exchg} = 2$. So node 2 has the exchange counter of 2 before performing the exchange. During exchange, it decrements the exchange counter to 1 and uses the $(2-1) = 1$ st slot from among the slots reserved for exchange (slot 21 in the example) since it is the first terminal performing the exchange. Note that when an exchange initiator performs an exchange, it is limited to its message boundary and it does not spill over into other transmissions.

Hence, by using the transfer slots and the exchange slots, the exchange counter is maintained in a distributed manner. This enables limiting the number of exchanges in every super-frame and thus enables controlling the number of actual slots available for data transmission. In addition, it allows for reclaiming the slots used up for exchange in an exclusive manner; thus preserving the actual number of slots allotted to each node for performing data transmission. Thus the protocol preserves the schedulability guarantee given for messages at the time of admission and effectively uses channel resources.

The timing diagrams shown in the figure 3.4 explain the exact transmissions that take place for the above example during the working of the Exchange protocol. The timing diagram of the offline schedule is indicated in fig 3.4.(a) which highlights a part of the DTP. As per the schedule broadcasted by the scheduler, Terminal 2 transmits during slots 6-13 followed by a transfer slot. In slots 15-20, terminal 1 transmits to its receiver, terminal 3. Slots 21 and 22 are reserved for exchange since the exchange counter value is 2.

Figure 3.4.(b) indicates the schedule that is followed during the DTP. When the actual data transmission happens, in slot 6, terminal 2 realizes that its channel with terminal 3 is noisy. Hence it initiates an exchange with terminal 1 in slot 7 and exchanges its next 6 slots. Terminal 2 decrements its exchange counter from 2 to 1 and passes it to terminal 1 in this exchange slot. Since C21 is in good state during slot 7, the exchange proceeds successfully.

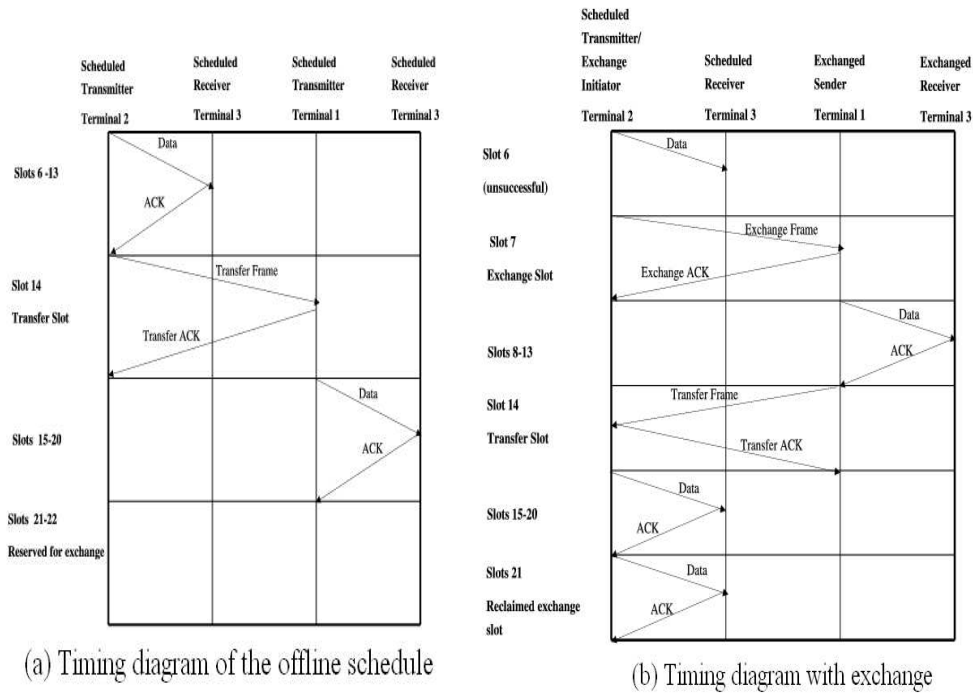


Figure 3.4 Timing diagram

From slot 8, terminal 1 begins transmitting to its receiver, terminal 3 and since C13 is in good state, the transmissions proceed successfully. Slot 14 is a transfer slot where terminal 1 passes back the control to terminal 2 with the exchange counter remaining at 1 since terminal 1 did not need to perform any exchange during the exchange period. In slots 15-20, terminal 2 continues its transmissions to terminal 3 which it had earlier deferred. Since the channel state has turned good, the transmissions proceed successfully. To compensate for the exchange slot, terminal 2 uses the first slot amongst the slots reserved for exchange i.e. slot 21 where it transmits an additional data unit to terminal 3. Hence on the whole, only a single data unit corresponding to the channel being noisy in slot 6 needs to be re-transmitted.

During the execution of the above exchange protocol each terminal can be in one of the following states: Idle, normal transmit mode, exchange mode and exchanged transmit mode. The transition between the different states is illustrated in figure 3.5. We use the boolean variables ME , DT_x , ET_x , ER_x , ET and EE to represent different conditions and frame transmissions leading to state transitions. The corresponding definitions are shown in the figure

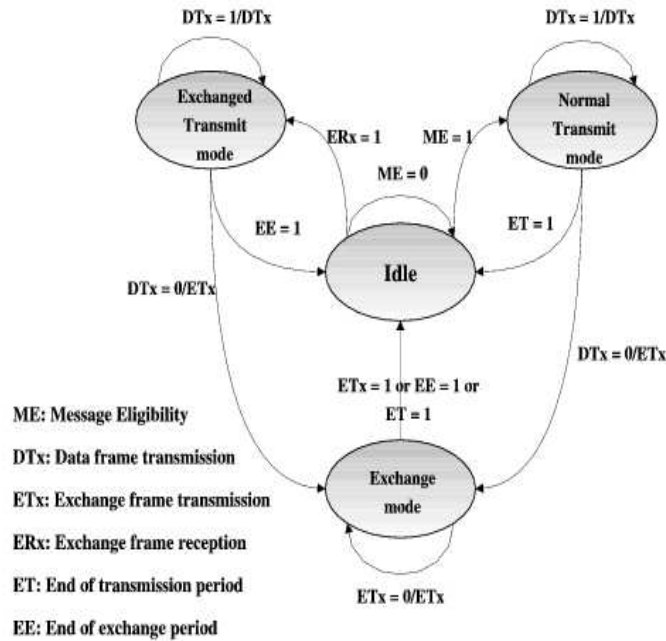


Figure 3.5 State Transition diagram

3.5. A value of zero for each boolean variable indicates that the corresponding condition/frame-transmission is false/unsuccessful while a value of one indicates true/successful. When the message of a wireless terminal becomes eligible and its slot to transmit occurs as per the schedule transmitted ($ME = 1$), the station moves from idle state to the normal transmit mode. It then begins to send the data packets to its receiver (DT_x). Once all the slots allocated to the terminal are used up ($ET = 1$), the transmission ends and the terminal moves back to the idle state. While transmitting to the destination, if the channel becomes erroneous ($DT_x = 0$), the wireless terminal moves to the exchange mode and plays the role of an exchange initiator by transmitting an exchange frame (ET_x) to a different sender (exchanged sender). The exchanged sender would take over the transmission when the exchange frame is received successfully ($ET_x = 1$). The exchange initiator now moves to the idle state. However, if the exchange frame is erroneous ($ET_x = 0$), the exchange initiator tries to exchange with a different exchanged sender until the scheduled end of its transmission phase occurs ($ET = 1$) or the exchange frame transmitted is successful ($ET_x = 1$). While in the idle state, a node can be the exchanged sender for a different wireless terminal if it receives the exchange frame

($ER_x = 1$). In such a case, the wireless terminal (now an exchanged sender) would transition into the exchanged transmit mode and transmit messages to its receiver until the end of the exchange period ($EE = 1$). If however, the transmission is erroneous, it would transition to the exchange mode and a similar process of finding a prospective exchanged sender would ensue. In the exchange mode, if the end of exchange period occurs ($EE = 1$), the exchanged sender transitions to idle state.

3.4 Simulation studies

We simulated a single hop wireless network with 10 machines over a 1Mbps channel with periodic messages of size 1050 bits and aperiodic of size 450 bits which are the typical message sizes in an industrial communication network. Each slot has a time duration equal to the transmission time of 150 bits. We simulated the different channel conditions using the Gilbert-Elliot model for different values of the model parameters. In our simulation studies we compared the performance of the above proposed protocols. The performance metric for all our simulation studies is the loss rate defined as the ratio of number of deadline violated to the number of messages admitted. We studied the effect of the following parameters on the performance of different protocols:

- Bad state probability (P_{bb}): Given that the channel is in a bad condition, P_{bb} represents the probability that the channel remains in the bad state.
- Good state probability (P_{gg}): Given that the channel is in a good condition, P_{gg} represents the probability that the channel remains in the good state.
- System load (S_l): This is the ratio of number of slots admitted to the total number of data transmission slots in the data transmission phase.
- Number of exchange slots (N_e): This denotes the number of exchanges that can be performed in a given super-frame.
- Message size (M_s): Number of slots required for the complete transmission of a message.

- Number of messages (N_m): Total number of messages per super-frame is given by N_m .

In the next subsection, we present our results along with the discussions.

3.4.1 Effect of the bad state probability (P_{bb})

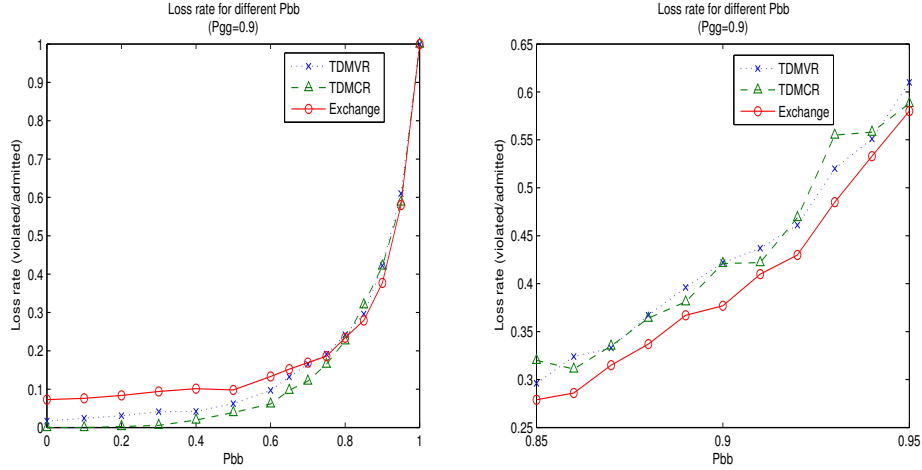


Figure 3.6 (a) Effect of P_{bb} (b) Effect of high P_{bb}

Figure 3.6.(a) compares the loss rates incurred by the above three protocols by varying P_{bb} . The other parameters are assumed as follows: $P_{gg} = 0.9$, $S_l = 1$, $N_e = 11$, $M_s = 7$, $N_m = 10$. The graph shows the effect of the erroneous channel burst length and has two distinct regions of interest corresponding to ($P_{bb} < 0.8$ (small burst region) and $P_{bb} \geq 0.8$ (large burst region)). In the small burst region, with low values of P_{bb} the channel quickly switches to the good state and the benefits of the exchange protocol are not very significant. In fact the overhead due to the exchange scheme overshadows the benefits of the protocol. With such small bad state bursts it appears that a simple retransmission mechanism would perform well. On the other hand, in the large burst region (shown enlarged in figure 3.6.(b)) which depicts the typical industrial environment and which is the major focus of this paper, the exchange protocol performs better than the basic schemes due to the fact that the exchange protocol *exchanges* the slots of a bad channel at the beginning of the bad burst with a good channel which is not noisy. At, $P_{bb} = 0.9$, the exchange protocol gives an improvement of 10.6% over TDMCR and 10.5 % over TDMCR.

Interestingly, towards the end of the large burst region where $P_{bb} \geq 0.96$, exchange protocols behave similar to the basic protocols. This is due to the fact that at very large values of P_{bb} the channel experiences significantly long bursts that deferred transmissions also encounter the erroneous channel condition

3.4.2 Effect of the good state probability (P_{gg})

Figure 3.7.(a) compares the loss rates incurred by the three protocols by varying P_{gg} . The other parameters are assumed as follows: $P_{bb} = 0.9$, $S_l = 1$, $N_e = 11$, $M_s = 7$, $N_m = 10$. The graph has two distinct regions of interest corresponding to $P_{gg} < 0.8$ and $P_{gg} \geq 0.8$. At low values of P_{gg} the channel quickly switches to the bad state and hence experiences frequent bad state bursts whose size is depicted by the P_{bb} value. This results in an exchange being performed from a bad channel to another channel that also moves into bad state frequently; hence the benefits of the exchange protocol are not very significant. Therefore, in such a case, a simple retransmission mechanism would perform better than the exchange protocol in terms of the loss rate.

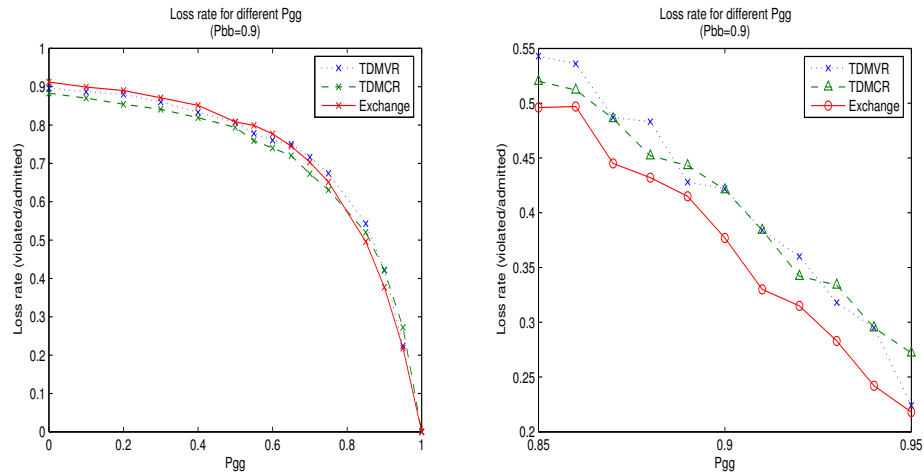


Figure 3.7 (a) Effect of P_{gg} (b) Effect of high P_{gg}

On the other hand, at high P_{gg} (shown enlarged in figure 3.7.(b)), which is the typical scenario in an industrial environment, the exchange protocol performs better than the basic schemes. This is due to the fact at high P_{gg} , the channels are in good state for a longer

time and for a erroneous channel,the exchange protocol *exchanges* its slots with a good channel. At, $P_{gg} = 0.91$, the exchange protocol gives an improvement of 14% over the basic schemes. Therefore, at very large values of P_{gg} the exchange protocol performs better than the others and at $P_{gg} = 1$, all the schemes show similar results.

3.4.3 Effect of the system load (S_l)

Figure 3.8 compares the loss rates incurred by the above three protocols by varying the S_l (varying N_m and N_e proportionally). The other parameters are chosen as follows: $P_{bb} = 0.9$, $P_{gg} = 0.9$, $M_s = 7$. With the increasing system load, for all the schemes the super frame becomes more and more occupied. Consequently more and more messages get affected by the bad channel state and hence an increase in the loss rate is observed. At low values of S_l , the exchange protocol behaves similar to the distributed scheme. This is because there are too few transmissions to perform any exchanges at low S_l . On the other hand, at higher values of S_l the exchange protocol makes use of the abundant messages that are present for performing effective exchanges and tends to perform better than the basic schemes.

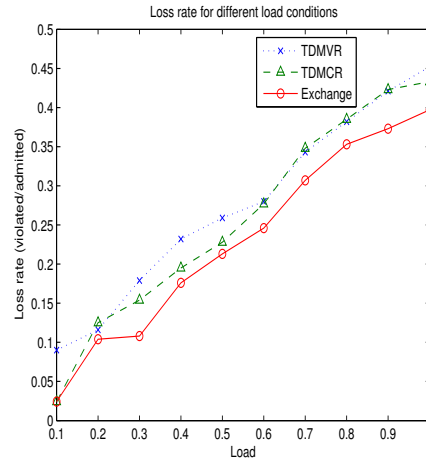


Figure 3.8 Effect of the system load

3.4.4 Effect of the Number of exchange slots (N_e)

In this part of our simulation study, we study the effect of the N_e by varying the message sizes and number of messages per super-frame. The value of N_e represents the number of slots reserved for performing exchanges. We have chosen $P_{gg} = P_{bb} = 0.9$ for these simulations.

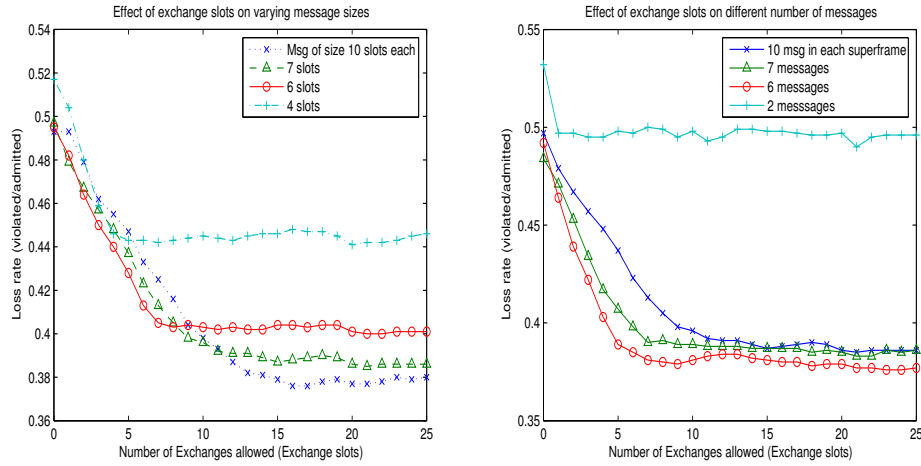


Figure 3.9 (a) Effect of N_e for different M_s (b) Effect of N_e for different N_m

- Effect of the message size (M_s) Figure 3.9.(a) shows the effect of the N_e for different M_s values keeping N_m fixed at 10 . With the increasing N_e , the loss rate for all message sizes decrease due to the benefits of the exchange protocol. However, after a point, N_e becomes more than the maximum number of exchanges that need to be performed and hence the loss rate saturates beyond that point. The saturation point depends on the message size, number of messages and channel parameters. For large message sizes, the saturation point is higher (12 for message of size 10 while it is 8 for message of size 4) since more exchanges can be performed.
- Effect of the number of messages (N_m) Figure 3.9.(b) shows the effect of the N_e for different M_m values keeping M_s fixed at 7 . With increasing N_e , the loss rate for all N_m values decrease due to the benefits of the exchange protocol. However, after a point, N_e becomes more than the maximum number of exchanges that need to be performed and

hence the loss rate saturates beyond that point. As in the previous case, the saturation point is higher for large number of messages (2 for $N_m=2$ while it is 12 for $N_m=10$) since the number of exchanges that can be performed is more when the number of messages increase.

3.5 Contributions and Future work

In this paper, we propose a framework for real-time message scheduling in a wireless industrial network. The contributions of this work are as follows:

- A framework for combined scheduling of periodic and aperiodic real-time messages
- A MAC protocol, called the Exchange protocol, that dynamically adapts to the channel conditions during run-time improving the success rate of real-time messages

The salient features of the Exchange protocol are as follows:

- Takes the bursty error characteristics of the wireless links explicitly into account.
- Exploits the spatial diversity characteristics of the wireless channel which states that when a channel is currently in the bad state, it is likely that there is a channel in the good state on which message transmissions can succeed
- Exploits the temporal diversity characteristics of the wireless channel with states that when a channel is currently in the bad state, it is likely that it will move to the good state in some time
- Adapts to the channel condition in a distributed manner
- Ensures collision-free transmissions
- Maintains the schedulability guarantees given to the messages
- Provides better loss rate compared to baseline protocols

As parts of our future work, we plan to extend the protocol to a multi-hop environment and improve the protocol through channel estimation techniques.

CHAPTER 4. Real-Time MAC protocols for a Two-level Heterogeneous Wireless Network

In the previous chapter, we proposed the exchange protocol that works on an offline schedule of messages in a single hop network and dynamically adapts to the erroneous wireless channel characteristics at runtime such that the number of messages that meet their deadline is maximized. In this paper, we extend the protocol to a hierarchical network with multiple hops to minimize the periodic message deadline misses and maximize the schedulability of aperiodic messages by enabling as many parallel transmissions as possible.

4.1 System Model

4.1.1 Network Model

We consider a industrial setup where all the devices in an industrial environment form a hierarchical network as shown in Figure 4.1. Each device in the network is comprised of a communication sub-system. We categorize the devices into different levels based on their functionality and communication range. Devices like sensors which have very basic measurement sensing functionality and short-range communication form Level 1 of the network. The Level 1 devices, also called the intra-cell nodes may communicate with other Level 1 devices in their range directly using intra-cell messages. The messages a_1 through a_{16} in Figure 4.1 represent such communication. All devices in this range are considered to be part of a cell. To communicate with other devices out of their range, there exists a controller, also called an inter-cell node, in each cell which is a more powerful node with a better communication range than the Level 1 devices. In the figure, nodes N_1 through N_8 which are located at the center of each cell are the inter-cell nodes. These inter-cell nodes may have their own

sensing or computation sub-system apart from having a communication sub-system. In the figure, message communication between N_1 and N_2 occurs by passing message m_1 which is an inter-cell message. Typically, the inter-cell nodes in each cell collect information pertaining to the cell and exchange it with the inter-cell nodes in other cells. Messages m_1, m_2, \dots, m_8 being exchanged between the inter-cell nodes N_1, N_2, \dots, N_8 form the Level 2 layer of communication. Based on the information received, each of these inter-cell nodes may do certain local processing and take decision which are dissipated to the intra-cell nodes in that cell for actuation. If for a intra-cell node, the messages destination is a intra-cell node in a different cell, it is first transmitted to the inter-cell node of the source cell which in-turn transmits the message to the inter-cell node of the destination cell. The destination inter-cell node then transmits the message to the respective destination intra-cell node in its cell. These inter-cell nodes can again be grouped which have a Layer 3 node preceding over the activities thus forming a network that is scalable to many levels. In this paper, we restrict ourselves to a hierarchical network comprising of 2 Levels that covers the entire factory floor. We assume that each of the Level 2 devices can communicate with each other either directly or using several multi-hop message. The network shown in Figure 4.1 comprises of both intra-cell(a_i) and inter-cell(m_i and ap_i) messages. The dashed lines indicate periodic messages(m_i and a_i) and the dotted lines indicate aperiodic messages(ap_i). The figure illustrates aperiodic messages among Layer-2 devices only. However, aperiodic messages may arrive dynamically at a device at any level and without loss of generality, all the protocols described in this work can also be applied to aperiodic messages among Layer 1 devices. The scheduling policy needs to schedule as many of the messages at run-time without violating the guarantees provided for periodic messages. We use the system in Figure 4.1 throughout this work to illustrate the working of different protocols.

4.1.2 Message Characteristics

We consider communication between devices in an industrial network where several real-time messages need to be transferred between devices and machines for an entire workflow to be

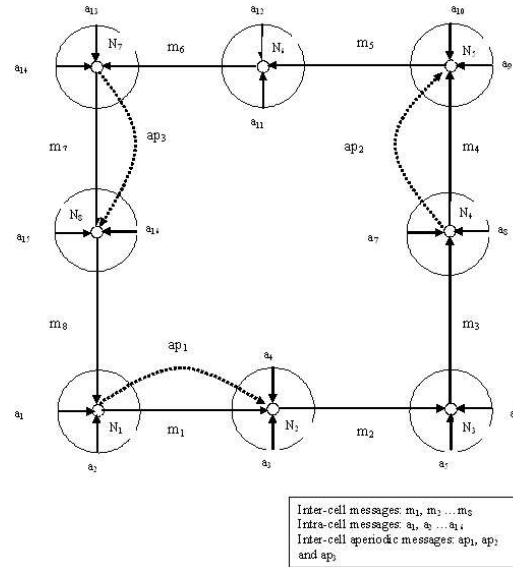


Figure 4.1 Hierarchical Industrial Network Setup

operational. Each message is associated with the deadline and failure to meet the deadlines can be catastrophic. All such real-time messages can be classified as being periodic and aperiodic. Periodic messages are the ones that make the production facility operation and are comprised of messages transferred between different sensing and actuating devices as well as controllers. These periodic messages need to be scheduled such that the number of messages that miss their deadlines is minimized. The aperiodic messages typically are of the form of alarms and denote the mal-functioning or other high priority messages that need immediate attention. The requirement for aperiodic message is that they be transmitted with minimum latency. The inter-cell and intra-cell message communication can be comprised of both periodic and aperiodic messages.

4.1.3 Channel Model

We consider transmissions to follow a superframe structure that repeats itself. All the periodic messages are assumed to repeat every super frame. We work with slotted time where each message is broken down to occupy several slots. Each slot represents the time in which the source can transmit the smallest atomic unit of a message and receive an implicit acknowl-

edgement.

Measurements in a wireless industrial environment[50] indicate that the channel causes errors that occur in bursts which can be captured by the Gilbert Elliot model[17, 20]. This model consists of the good and bad state as indicated by the state diagram in Figure 4.2. When the channel is in the good state, P_{gg} is the probability that the channel continues to remain in the good state and P_{gb} is the probability with which the channel moves in the bad state. In the same way, when the channel is in the bad state, there is a certain probability (P_{bb}) with which the bad state is retained and P_{bg} is the probability of moving into the good state. We assume every message is transmitted on a channel that that follows its own Gilbert Elliot model and that the channel condition changes at the end of every slot. We assume that all transmissions that occur in a slot when a channel is in the bad state is erroneous that is indicated by the lack of acknowledgement for that slot and that slot transmission is considered to have failed.

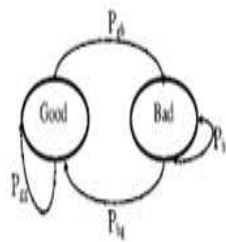


Figure 4.2 Channel Model

4.2 Objective

The objective of this paper is to come up with an efficient channel-adaptive scheduling scheme for combined scheduling of periodic and aperiodic real-time messages in a 2-level hierarchical wireless industrial network. The wireless medium is a broadcast medium but since the transmission ranges of the devices at different levels are different, it is possible that two messages that do not conflict with each other be scheduled in parallel. Hence, we want to maximize the number of messages parallel transmission that can occur while making sure that

no two transmissions that interfere be scheduled at the same time. The wireless channel is known to cause large bursts of packet loss and hence there needs to exist a mechanism that dynamically adapts to the erroneous channel condition to ensure that the number of messages that miss the deadline is minimized. In essence, we want to minimize the deadline misses for periodic messages and increase the schedulability for aperiodic messages while preserving the schedulability guarantees given for periodic messages. We want to achieve this by overcoming the vagaries associated with the channel and enabling maximum utilization of the channel by enabling parallel transmissions.

4.3 Assumptions

We consider a generic message system where both the Layer 1 and Layer 2 devices in the network have periodic messages and dynamically arriving aperiodic messages to transmit. The source and destination of periodic and aperiodic messages are known a priori. The period of periodic messages is known but aperiodic messages arrive dynamically at the nodes. The location of each of the nodes and the communication range for transmitting each message is determined beforehand. We assume that each device transmits using a power level that is just necessary to reach the destination and this level is predetermined. Also, any communication outside of this transmission range can go on in parallel. We also assume that all the messages are of the same size. We use the symbols shown in Table 4.1 in the rest of our work.

4.4 Background Information

4.4.1 Vertex coloring

In the context of this problem, a parallel, conflict-free schedule for a set of messages in a wireless network can be obtained using a vertex coloring approach. The parallel schedule is formed by constructing a contention graph $G = (V, E)$ which represents conflicts among all the transmissions that exist in the network. In this graph, each vertex represents a transmission (both periodic and aperiodic) and an edge exists between two vertices iff the two transmissions (vertices) cannot be scheduled simultaneously. In our case since every data transmission

Table 4.1 Explanation of symbols

Symbol	Explanation
n_s	Number of slots in the superframe
m_s	Message size in slots
S	Total messages in the system
S_m	Set of inter-cell messages
S_a	Set of intra-cell messages
S_p	Set of periodic messages
S_{ap}	Set of aperiodic messages
$S_{l,p}$	Set of locally available periodic messages(at a particular node)
$S_{l,ap}$	Set of locally available aperiodic messages(at a particular node)
P_i or $P(m_i)$	Set of messages scheduled to transmit parallelly with message m_i
E_i or $E(m_i)$	Set of messages with which message m_i can exchange
I_i or $I(m_i)$	Set of messages that interfere with the transmission of message m_i

unit is followed by an Acknowledgement, a conflict occurs if either the data transmission from the source or the ACK from the destination of any message conflicts with any other data transmission or acknowledgement. A conflict may also occur because of a hidden node. These scenarios are considered when constructing the edges for the graph. By coloring the vertices of the graph with minimum colors such that no two adjacent nodes connected by an edge have the same color and grouping similar colors together, we can construct a parallel schedule that ensures that all the message complete within the shortest time. Since the problem of finding a minimum coloring for a graph is NP-hard, we use a heuristic[9] to construct the parallel transmission schedule. The inputs and outputs of any vertex coloring algorithm is illustrated in Algorithm 1.

<p>Input: Set of message S with their source and destination information, network topology</p> <p>Output: Parallel Schedule with λ classes of messages where each class corresponds to a particular color S_λ denotes the set of message in each class belonging to the same color</p>

Algorithm 1: Vertex coloring algorithm

Consider the network in Figure 4.1 comprising of both intra-cell(a_i) and inter-cell(m_i and ap_i) messages. We want to construct a parallel schedule for these set of messages. We define an interference set I_i for each inter-cell message in the network to be the set of messages with

Message	Interference Set I_i	Exchange Set E_i
m1	$I(m1) = \{a1, a2, a3, a4, a5, a6, a15, a16, m2, m3, m7, m8, ap1, ap3\}$	$E(m1) = \{m4, m5, m6, m7, m8, ap2, ap3\}$
m2	$I(m2) = \{a1, a2, a3, a4, a5, a6, a7, a8, m1, m3, m4, m8, ap1, ap2\}$	$E(m2) = \{m1, m4, m5, m6, m7, m8, ap1, ap2, ap3\}$
m3	$I(m3) = \{a3, a4, a5, a6, a7, a8, a9, a10, m1, m2, m4, m5, ap1, ap2\}$	$E(m3) = \{m1, m2, m6, m7, m8, ap1, ap3\}$
m4	$I(m4) = \{a5, a6, a7, a8, a9, a10, a11, a12, m2, m3, m5, m6, ap2\}$	$E(m4) = \{m1, m2, m3, m6, m7, m8, ap1, ap3\}$
m5	$I(m5) = \{a7, a8, a9, a10, a11, a12, a13, a14, m3, m4, m6, m7, ap2, ap3\}$	$E(m5) = \{m1, m2, m3, m4, m8, ap1, ap2\}$
m6	$I(m6) = \{a9, a10, a11, a12, a13, a14, a15, a16, m4, m5, m7, m8, ap2, ap3\}$	$E(m6) = \{m1, m2, m3, m4, m5, m8, ap1, ap2\}$
m7	$I(m7) = \{a1, a2, a11, a12, a13, a14, a15, a16, m1, m5, m6, m8, ap1, ap3\}$	$E(m7) = \{m2, m3, m4, m5, m6, ap2\}$
m8	$I(m8) = \{a1, a2, a3, a4, a13, a14, a15, a16, m1, m2, m6, m7, ap1, ap3\}$	$E(m8) = \{m2, m3, m4, m5, m6, m7, ap2, ap3\}$
ap1	$I(ap1) = \{a1, a2, a3, a4, a5, a6, a15, a16, m1, m2, m3, m7, m8, ap3\}$	$E(ap1) = \{m4, m5, m6, m7, m8, ap2, ap3\}$
ap2	$I(ap2) = \{a5, a6, a7, a8, a9, a10, a11, a12, m2, m4, m3, m5, m6\}$	$E(ap2) = \{m1, m2, m3, m6, m7, m8, ap1, ap3\}$
ap3	$I(ap3) = \{a1, a2, a11, a12, a13, a14, a15, a16, m1, m5, m6, m7, m8, ap1\}$	$E(ap3) = \{m2, m3, m4, m5, m6, ap2\}$

Figure 4.3 Interference and Exchange Sets for inter-cell messages

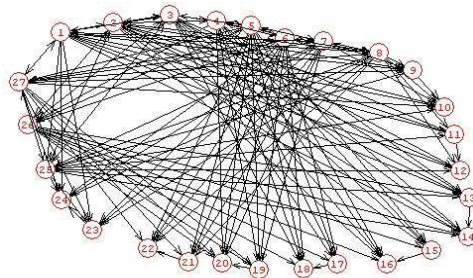


Figure 4.4 Conflict Graph

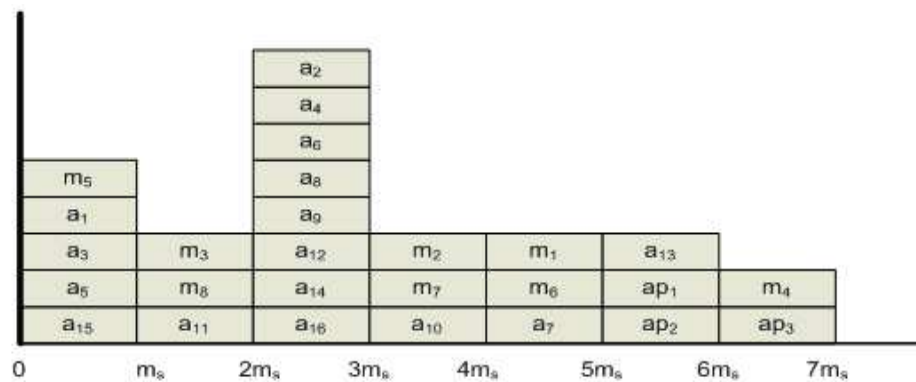


Figure 4.5 Parallel Schedule with all messages in the system

which the transmission of the inter-cell message would conflict. i.e. the inter-cell message cannot be scheduled to transmit in parallel with any other message $\in I_i$. Figure 4.3 gives the interference set for each inter-cell message. The conflict graph for the network is shown in Figure 4.4. Applying the vertex coloring heuristic algorithm and grouping similar colors gives the parallel schedule as shown in Figure 4.5.

In a normal scheduling policy for this network, slots equivalent to the message size are allocated and according to the schedule all the periodic intra-cell and inter-cell messages are transmitted without any collisions. Slots are allotted for each aperiodic message in the same way as is done for periodic messages. If a particular aperiodic message has arrived by the scheduled time, it is transmitted. If the message arrives after the schedule time in the super-frame, the scheduled time in the current super-frame is left idle and the aperiodic message gets transmitted during the scheduled time in the next super-frame. Aperiodic intra-cell messages would be scheduled in a similar way as inter-cell aperiodic messages.

4.4.2 Exchange Protocol

We proposed the exchange scheme in [26] to improve the success rate of messages meeting their deadline in a wireless network. It forms its basis on the bursty wireless channel conditions and the fact that a channel currently in an erroneous state that has its message transmissions failing would move to the good state as time progresses and that another channel might currently be in the good state and could have its transmissions coming through successfully. The main idea is that when a transmission on a channel begins to fail because of the erroneous channel conditions, the message exchanges its currently scheduled time slots with another message that uses a different transmission channel. The original message is scheduled during the scheduled time slots of the exchanged message by which time its channel would have moved to the good state. The protocol incurs some control overheads to perform the exchange and these slots are recovered in a distributed conflict-free manner and in this way, a scheduling scheme that dynamically adapts to the channel conditions at run-time in a collision-free manner for improving the success rate of messages and which preserves the schedulability guarantee given

to the messages is obtained. We now explain the exchange setup that we consider for the hierarchical system model that we consider in this work.

4.4.3 Exchange Setup

We restrict exchanges within every level i.e. inter-cell messages exchange with other inter-cell messages and intra-cell messages with other intra-cell messages in their cell. From Figure 4.3 and from the graph in Figure 4.4, we see that inter-cell messages have more constraints(edges) than the intra-cell level messages and this is true for any network. When a message is being transmitted between two cells, no intra-cell messages can go on in parallel in either of the source or the destination cells. Hence we concentrate on exchanges between inter-cell messages when forming a schedule. When an intra-cell message such as a_1 is scheduled, at run-time, it can exchange with other messages in its cell such as a_2 independently without affecting the transmissions or exchanges occurring between the intra-cell messages in other cells. Hence we initially schedule the inter-cell messages m_1 through m_8 such that they can exchange among each other and the intra-cell messages a_1 through a_{16} are scheduled around the inter-cell messages and they would be able to independently exchange among themselves.

In this paper, we want to extend the exchange scheme and propose a scheduling protocol for a hierarchical(2-level) wireless network that dynamically adapts to the channel conditions using the exchange scheme explained above. The scheduling protocol should maximize the number of messages that can be sent simultaneously in a collision-free manner. The constraint is that no two messages that interfere each other can be transmitted simultaneously and the schedulability guarantee should be preserved in a distributed fashion i.e. if a message was allotted certain slots for transmission in a conflict-free manner in the offline schedule, it should get the same number of data transmission slots during run-time even after adapting to the channel conditions.

The approach we take to solve this problems is as follows: we formulate the conditions of exchange in a parallel message setup and show why a completely parallel offline schedule using the vertex coloring mechanism explained in Section 4.4.1 would not result in a schedule

congenial for exchange. We then explore the mechanism to construct a parallel schedule that provides exchange capability at run-time and show how such a static schedule does not lead to much parallelism in transmissions. Hence, there is an inherent conflict between enabling exchanges and the amount of parallelism that such a schedule can provide. We then go on to explain the dynamic adaptive scheme that we propose to schedule messages in a hierarchical network that achieves parallelism among message transmissions at run-time while giving the room to perform exchange in a distributed fashion on the event of hostile channel conditions.

4.4.4 Conditions for exchange in a parallel message setup

We now explore the conditions that are necessary to enable exchanges given a parallel message schedule. For ease of explanation of the algorithm, we restrict exchanges to a single level i.e. when a message exchanges with another message, the exchanged message does not initiate another exchange while in it transmitting in the original message's slots. This means that no nested exchanges are permitted. We describe both the schedule and protocol considerations that determine the formation of the exchange set and describe the algorithm to construct the exchange set given any parallel transmission schedule.

4.4.4.1 Schedule Considerations

Consider a generic parallel transmission schedule obtained using the vertex coloring approach described in Section 4.4.1. Let P_i denote the set of messages that have been scheduled to transmit parallelly with m_i . Let I_i denote the set of messages that conflict with the transmission of a message m_i . Let E_i denote the set of messages with which m_i can exchange as shown in Figure 4.3. The exchange set E_i was formed considering that the channel for each message is independent and a message m_i can exchange with all messages that do not use the same channel that m_i uses, for data transmission or for performing an exchange. In a parallel message setup, additional constraints are imposed on the ability to perform exchange, We need to ensure that when a message exchanges with another message, the exchanged message is able to transmit in a collision-free manner with all the other parallel transmissions that are

occurring at the current time. Also the deferred message should not interfere with the parallel transmissions that are scheduled during the deferred time. For e.g., for the schedule in Figure 1, if m_1 wants to exchange with m_2 , m_2 should not interfere with messages m_6 and a_7 and m_1 should not interfere with m_7 and a_{10} . In addition, if in a single slot, all the messages exchange, the exchanged messages must be able to co-exists. i.e. if m_1 exchanges with m_2 and m_6 exchanges with another message m_8 , m_2 and m_8 must be able to transmit in a collision-free manner.

4.4.4.2 Protocol Considerations

The exchange protocol initiates an exchange by having the source of a message transmit an exchange frame to the source of the prospective message to exchange with in the exchange set. If this message is successful, the exchanged message begins its transmission. Consider that m_i wants to exchange with m_k . Then the channel from the source of m_i to the source of m_k should not interfere with any of the parallel message transmissions or the exchanged messages of those parallel messages or the channel between the source of the parallel message transmissions and the exchanged message transmissions. This condition is necessary since exchanges can be initiated at any time during the course of data transmissions. In addition, a prospective message for exchange is a message that does not use the same channel as the original message for data transmission or for exchange initiation. The later condition means that the if the channel between the source of a message m_i and the source of its prospective exchanged message uses the same channel as between the source and destination of m_i , then such a prospective exchanged message is no longer considered for exchange. These are the conditions imposed by the way the exchange protocol has been designed in [26]. Hence these are also taken into consideration while formulating the conditions of exchange in a parallel message setup. We summarize both the schedule and protocol considerations into the algorithm that constructs an exchange set E_i for every message m_i in the network.

4.4.5 Algorithm to construct Exchange set for a message, m_i given the messages that transmit in parallel with m_i

The algorithm to construct E_i given P_i is described in below:

Step 1: Consider a message m_i in the network. Find a message m_k such that m_k does not interfere with any message in P_i and m_i does not interfere with any message in P_k i.e. find a m_k such that $I(m_k) \cap P_i = \emptyset$ and $I(m_i) \cap P_k = \emptyset$. Add m_k to $E(m_i)$ and m_i to $E(m_k)$. Construct the exchange set for all messages in the network using Step 1.

In Step 1, we try to construct an exchange set for every message, m_i in the network by finding a exchange person, m_k for m_i such that m_k does not interfere with any message scheduled to transmit parallelly with m_i and m_i does not interfere with any message scheduled to transmit parallelly with m_k . If such a m_k is found, we can exchange m_i with m_k . We find all possible m_k s for every m_i .

Step2: For each message m_i and for each message $m_{e,i}$ in $E(m_i)$, check if $m_{e,i}$ uses the same channel as m_i for data transmission or exchange initiation. If they do or if $m_{e,i} \in P_i$, remove $m_{e,i}$ from $E(m_i)$.

This is a protocol consideration that checks if the exchanged message of any message m_i uses the same path as m_i for data transmission or exchange initiation. Since such a path is going to be erroneous since that was the primary reason why the exchange was initiated, we ignore exchanges with such messages. This step also checks to see if the exchanged messages was scheduled to transmit parallelly with the current messages m_i and if it was, it is removed from the set of prospective messages considered for exchange.

Step 3: If the channel between source of m_i and source of every message $m_{e,i} \in E_i$ conflicts with any message $m_{p,i} \in P_i$ or with any message $m_{e,p_i} \in E(m_{p,i})$ or with the channel between source of $m_{p,i}$ and source of m_{e,p_i} , then remove $m_{e,i}$ from E_i .

Step 3 tries to impose conditions based on the protocol considerations on the eligibility of a message to be part of an exchange set. It ensures that the channel used for exchange initiation does not conflict with any of the parallel transmissions or the exchanges of a parallel transmission or the exchange initiation path between a parallel message and its exchanged

message.

Step 4: For every message m_i , for each message $m_{e,i}$ in E_i and $m_{p,i}$ in P_i , if $E(m_{p,i}) \cap I(m_{e,i}) \neq \emptyset$ remove $m_{e,i}$ from E_i .

Step 4 tries to ensure that none of the exchanged message conflict with each other.

Applying the algorithm to the schedule in Figure 4.5, we see that all the exchange sets are empty and the parallel schedule in Figure 4.5 for the network does not permit any exchanges between the inter-cell messages.

4.4.6 Graph Transformation to develop parallel schedule permitting exchanges

We now describe the technique to construct a parallel schedule for a network given the number of exchanges that each message can perform, ignoring the specifics of the exchange protocol(Steps 2 and 3 of the algorithm above) i.e. given E_i for each message in the network, we want to find out P_i .

We use a conflict graph and the vertex coloring approach explained in Section 4.4.1 to construct a parallel schedule. However, apart from adding edges to the conflict graph when two message transmissions conflict, we include extra edges so that the exchanges specified in E_i for each message is permitted. For each message m_i , we add edges in the conflict graph from m_i to all other messages that conflict with m_i which signifies that m_i should not be scheduled simultaneously with any of these messages. In addition, we also add edges from m_i to all messages that conflict with $E(m_i)$ that signifies the fact that no message that conflicts with any message in $E(m_i)$ should be scheduled to transmit parallelly with m_i . On coloring the graph, we obtain a parallel schedule permitting exchanges.

By transforming the graph as mentioned before, we ensure that Step 1 and Step 4 of the algorithms are captured by the additional edges that are added to each message. We show how two exchanged messages would not conflict with each other(step 4) is captured.

Applying the above mentioned technique to the messages in the network illustrated in Figure 4.1, we obtain the schedule shown in Figure 4.6 for the exchanges shown in the Table 4.2. Several heuristics can be considered for forming the exchange set for each message. The

Table 4.2 Exchange Set for each message for the determination of parallel schedule

Message	Exchange Message
m_1	m_4
m_2	m_8
m_3	m_7
m_4	m_1
m_5	m_4
m_6	m_5
m_7	m_3
m_8	m_2
ap_1	ap_2, ap_3
ap_2	ap_1, ap_3
ap_3	ap_2

heuristic for forming the exchange set shown in the Table 4.2 was to form an exchange set for each message m_i comprising of all the messages that m_i could exchange with considering the channel constraints imposed in step 2 of the algorithm mentioned in Section 4.4.5 and choosing one in this set, m_k such that $E(m_i)$ contains m_k and $E(m_k)$ contains m_i and adding m_i to $E(m_k)$ and m_k to $E(m_i)$. If no such message is available, any message with which an exchange can be performed is chosen. Another heuristic is to choose an exchange message m_k for each m_i such that amongst all the messages that can exchange with m_i considering the channel constraints imposed in Step 2 of the algorithm 4.4.5, m_k adds minimum number of edges to the conflict graph.

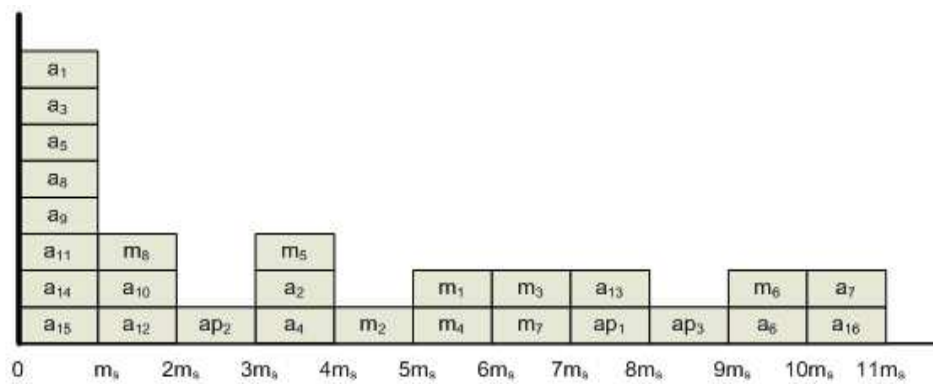


Figure 4.6 Parallel Schedule permitting exchanges of Table 4.2

This produces a schedule which occupies 11 slots as shown in Figure 4.6 which is greater than the original parallel schedule shown in Figure 4.5 by 4 slots. However, the increase in number of slots gives the ability to perform exchanges and in the process, the parallelism achieved for inter-cell messages (one inter-cell message transmitting with another inter-cell message) has been reduced.

If we allow all inter-cell periodic messages to exchange among each other and all inter-cell aperiodic message to exchange amongst each other i.e. if we impose a full exchange policy such that any message can exchange with any other message according to the exchange setup, the parallel schedule obtained by the DSTAUR algorithm occupies 13 slots with very little parallelism among the inter-cell message. This produces a serial schedule for all inter-cell messages with parallelism exploited for only the intra-cell messages as shown in Figure 4.7:

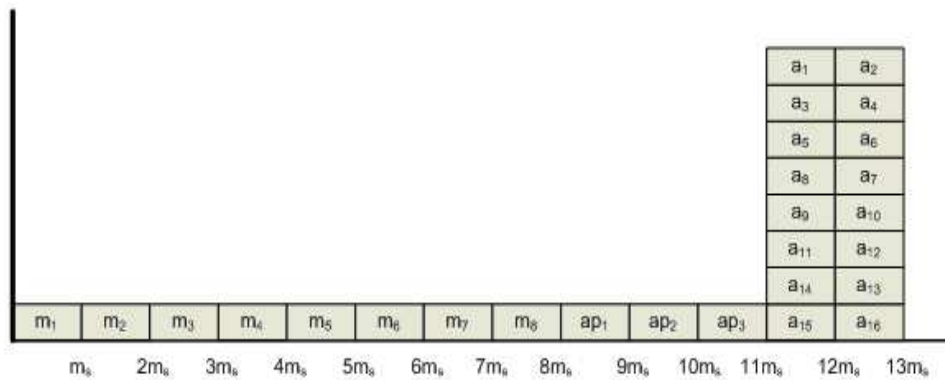


Figure 4.7 Parallel Schedule permitting full exchange

From the above two schedules in Figures 4.6 and 4.7, it is clear that there is an inherent conflict between permitting exchanges and the amount of parallelism that such a schedule would offer. More the number of exchanges, less parallel becomes the schedule since additional constraints are imposed on the messages that can be scheduled in parallel and this leads to more edges in the conflict graph. Even if by the above mentioned graph transformation, we can obtain a parallel schedule for a given set of exchanges that need to be performed, using the protocol used for exchange that we have proposed, there is no easy method to keep track of the number of exchanges that have been performed in a distributed manner. This is crucial

for preserving the schedulability guarantee of the message. Hence, when we use the exchange protocol, we always consider that the inter-cell messages are scheduled serially with a transfer slot between each message to keep track of the number of exchanges that have been performed. Exchanges proceed as has been described in [26]. In case of inter-cell messages, an intra-cell message within each cell can transmit and there exists a transfer slot between each intra-cell transmission in a cell as shown in Figure 4.8. Since these messages exchange with other intra-cell messages in the same cell, transmission in the transfer slot occurring within each cell will not collide and exchanges can be performed within a cell without causing disruptions to the transmissions occurring in other cells.

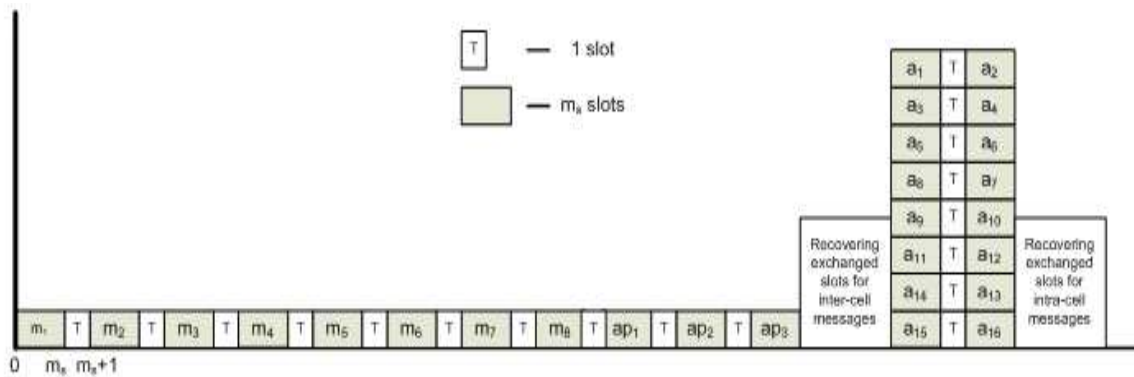


Figure 4.8 Schedule permitting full exchanges with control slots

4.5 Scheduling Algorithms

We propose three scheduling algorithms to schedule periodic and aperiodic messages in a two-level hierarchical wireless network as described in Section 4.1.1. We assume that the source and destination of both the periodic and aperiodic messages that need to be transmitted are known apriori. Each scheduling algorithm works in two stages there exists an offline phase where a collision-free schedule is constructed and distributed to all the nodes in the network. In this schedule, every station that has a message to transmit is given a certain number of collision-free exclusive slots equal to the message length, for data transmission. We term this as the schedulability guarantee for the message. During runtime, the offline schedule is

either followed in its original form or the wireless terminals adapt to the channel conditions in a distributed fashion to improve the loss rate and maximize the schedulability of aperiodic messages in such a way that the schedulability guarantees provided offline is retained. The three scheduling algorithms differ on the basis by which the schedule construction is done offline as well as the run-time policy followed for adapting to the channel conditions.

4.5.1 Framework

The basic framework is shown in Figure 4.9. There exists a control phase, an offline schedule construction phase, schedule transmission phase and data transmission phase for each of the protocols. In the control phase, all the messages in the system are sent to the central scheduler which performs an admission test that provides the set of messages that needs to be scheduled in every super-frame. The scheduler then applies an offline schedule construction algorithm to these set of messages based on the protocol being used and produces an offline schedule that is sent to all the nodes in the system. During run-time, at each of the nodes, an online-scheduling algorithm works on the offline schedule produced to dynamically adapt to the channel conditions while transmitting the messages in a distributed fashion. The offline schedule construction algorithm and the online scheduling algorithm varies for each of the three protocols.

The admission test that is performed during the Control Phase admits messages based on the availability of slots in the super-frame. We use the serial protocol as the base-line to determine the set of messages that need to be transmitted in the super-frame. The same set of messages are scheduled in each super-frame by the parallel and adaptive protocols. In the superframe, certain number of slots are allocated for inter-cell messages and certain slots for intra-cell messages. The admission test checks if the super-frame has enough free slots to accommodate the next message and its recurring instances (for periodic messages only) before its deadline.

Consider a inter-cell message of size M_i occupying N_i slots of the superframe on which the admission test needs to be performed. Let $N_{data,inter-cell}$ be the number of slots reserved for the

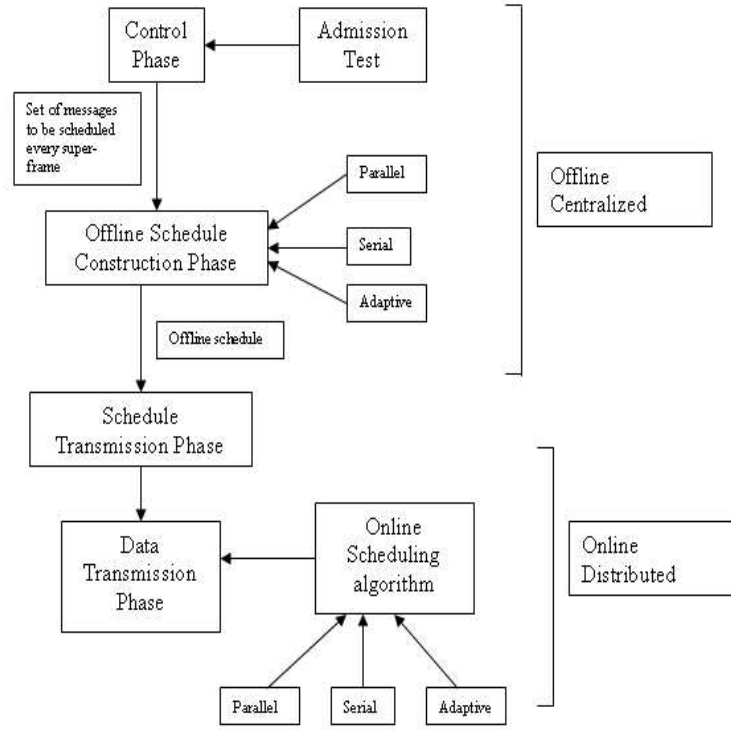


Figure 4.9 Framework

transmission of inter-cell messages in the super-frame. $N_{admitted,inter-cell}$ denote the number of slots of the super-frame occupied by already admitted inter-cell messages; $N_{transfer,inter-cell}$ denote the number of transfer slots and $N_{exchg,inter-cell}$ denote the number of exchange slots for inter-cell messages (more details about the usage of these slots are provided in [26]). In this work, we restrict the number of exchanges to one per inter-cell message transmission. Hence $N_{exchg,inter-cell} = N_{transfer,inter-cell}$ which equals the total number of inter-cell messages that can be accommodated within $N_{data,inter-cell}$. By restricting the number of overhead slots this way, we ensure that the size of the super-frame is fixed across all the protocols and the adaptive protocol can use the offline schedule similar to that used by the serial protocol as will be explained later. The admission test for inter-cell messages checks if

$$N_i \leq N_{data,inter-cell} - N_{admitted,inter-cell} - N_{transfer,inter-cell} - N_{exchg,inter-cell} \quad (4.1)$$

If the above condition is satisfied, the message is admitted to the system and the scheduler reserves N_i slots exclusively for the inter-cell message; else the message is rejected. However,

aperiodic messages are always admitted into the system by removing an instance of the periodic message, since they require higher priority.

Intra-cell messages across cells can be transmitted in parallel. Hence the slots reserved for intra-cell messages in the super-frame can be used simultaneously by intra-cell messages in each of the cells. Consider a intra-cell message of size M_i occupying N_i slots of the superframe on which the admission test needs to be performed. Let $N_{data,intra-cell}$ be the number of slots reserved for the transmission of intra-cell messages in the super-frame. $N_{admitted,intra-cell}$ denote the number of slots of the super-frame occupied by already admitted intra-cell messages of the cell to which current message belongs; $N_{transfer,intra-cell}$ denote the number of transfer slots and $N_{exchg,intra-cell}$ denote the number of exchange slots for exchanges within each cell. In this work, we restrict the number of exchanges to one per intra-cell message in each cell. Hence $N_{exchg,intra-cell} = N_{transfer,intra-cell}$ which equals the total number of intra-cell messages within each cell that can be accommodated within $N_{data,intra-cell}$. The admission test for intra-cell messages checks if

$$N_i \leq N_{data,intra-cell} - N_{admitted,intra-cell} - N_{transfer,intra-cell} - N_{exchg,intra-cell} \quad (4.2)$$

If the above condition is satisfied, the message is admitted to the system and the scheduler reserves N_i slots exclusively for the intra-cell message; else the message is rejected. However, aperiodic messages are always admitted into the system by removing an instance of the periodic message, since they require higher priority.

Once the admission test is performed, the number of messages to be scheduled within each super-frame is known. Based on the protocol used, an offline schedule construction algorithm is applied on these set of messages to arrive at an offline schedule that is distributed to all the nodes in the system in the schedule transmission phase. Using the offline schedule that each station received and based on the online scheduling algorithm, that adapts to the channel conditions, transmissions are performed in a collision-free manner. We now explain each of the protocol in detail.

4.5.2 Parallel Protocol

In this scheduling strategy, a parallel schedule is constructed offline based on the knowledge of all the messages that need to be sent in the network. Each message is given exclusive slots for data transmission and the schedule constructed is completely collision free. This is distributed to all the nodes during network set-up time. During run-time, in each super-frame, every node performs data transmission during their allotted time in the offline schedule.

- Parallel Offline Schedule Construction Algorithm:** There exists two different phases in the offline schedule: one where all messages are scheduled parallelly followed by a phase where a parallel schedule for aperiodic messages is developed. The vertex coloring approach explained in Section 4.4.1 is used for producing a parallel schedule for both the sets of messages without making a distinction between intra-cell and inter-cell messages. This schedule is shown in Figure 4.10 which is same as the schedule in Figure 4.5. However, the aperiodic schedule is followed only if the super-frame has free slots after following the schedule with all messages scheduled in parallel. In addition, since the arrival time of the aperiodic messages is unknown, the scheduled slot for an aperiodic message is used only if an aperiodic message is ready during that time. The pseudocode for the schedule construction algorithm is shown in Algorithm 2. The schedule thus formed is distributed to all the nodes in the network.

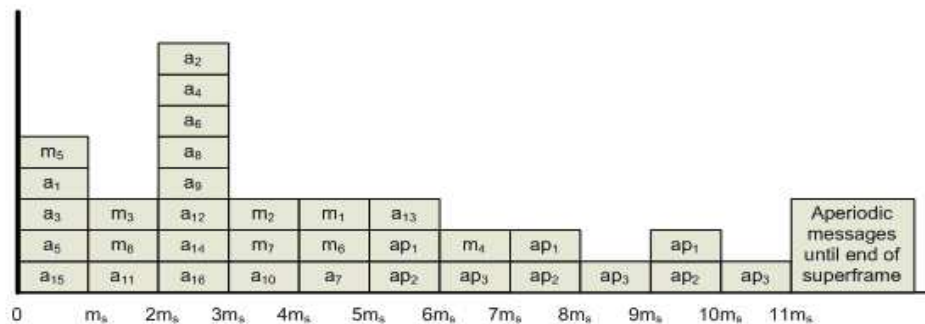


Figure 4.10 Schedule Produced by the Parallel Offline Schedule Construction algorithm

```

Input: Set of message  $S$ , Set of aperiodic messages  $S_{ap}$ 
Output: Parallel Schedule  $PS$  where  $PS_i$  denote the set of messages scheduled for
slot  $i$ 
1 NodeColoring( $S$ );
2 Set  $j = 1, i = 1$ ;
3 while  $j \leq \lambda$  //For each color do
4   | For each msg in  $S_\lambda$ , add  $m_s$  slots of msg to  $PS_i \dots PS_i + m_s$ ;
5   | Set  $i = i + m_s$ ;
6   | Set  $j = j + 1$ ;
7 end
8 while  $i \leq n_s$  do
9   | NodeColoring( $S_{ap}$ );
10  | Set  $j = 1$ ;
11  | while  $j \leq \lambda$  //For each color do
12  |   | For each msg in  $S_\lambda$ , add  $m_s$  slots of msg to  $PS_i \dots PS_i + m_s$ ;
13  |   | Set  $i = i + m_s$ ;
14  |   | Set  $j = j + 1$ ;
15  | end
16 end

```

Algorithm 2: Parallel Offline Schedule Construction algorithm

- **Parallel Online Scheduling Algorithm:** During runtime, the parallel offline schedule is followed by all the nodes and each message is transmitted during the time allocated to it in the schedule. The aperiodic message schedule is then followed if enough slots are left in the superframe. If a aperiodic message is pending, it is scheduled during its allocated slot. Else a periodic message with the same source-destination pair is transmitted. If no such message exists, the aperiodic slot is left idle. The pseudocode for the parallel scheduling algorithm is described in Algorithm 3.

Hence the above mechanism schedules both periodic and aperiodic messages allowing as many parallel messages as possible to be transmitted simultaneously. This leads to a scheme where all the messages complete in the shortest time. However from section 4.4.5, we can see that this mechanism offers little potential for adapting to the channel conditions by performing an exchange. Hence this scheme is not entirely suitable when scheduling messages over a wireless network where the channel conditions deteriorate frequently.

Input: Parallel Schedule PS , Set of locally available periodic messages S_l, p , Set of locally available aperiodic messages S_l, ap

Output: Message Transmission

```

1 Set  $i = 1$ ;
2 while  $i \leq n_s$  do
3   if any msg in  $S_l, p \in PS_i$  then
4     | Transmit msg;
5   end
6   else
7     | if any msg in  $S_l, ap \in PS_i$  and msg is ready then
8       | | Transmit msg;
9     | end
10  end
11  Set  $i = i + 1$  ;
12 end

```

Algorithm 3: Parallel Online Scheduling algorithm

4.5.3 Serial Protocol

The serial protocol tries to ensure a transmission schedule that gives the ability for any source of a message transmission to perform an exchange if the source finds that the data transmission is not successfully being received at the destination because of erroneous channel conditions. Exchanges are restricted to each level as explained in Section 4.4.4.

- **Serial Offline Schedule Construction Algorithm:** The offline schedule produces a completely serial schedule for all inter-cell messages that permit all possible combinations exchanges between them as shown in the Figure 4.11. All intra-cell messages are scheduled with maximum parallelism by accommodating intra-cell messages in each cell parallelly. The number of exchanges is fixed at 1 for each message and as many slots are allocated at the end of the inter-cell and intra-cell message schedules for each message to transmit data exclusively in these slots to make up for the slots that were used as exchange slots for performing an exchange, as explained in Section 4.4.2. Since no parallelism between inter-cell message is exploited in this scheme, the time taken for completing the message transmissions is longer than both the other schemes and hence the superframe length formed by the serial scheduling policy is uses as the basis for the

other schemes too. The offline schedule construction algorithm is shown in Algorithm 4. Once the schedule is formed offline, it is distributed to all the nodes in the system.

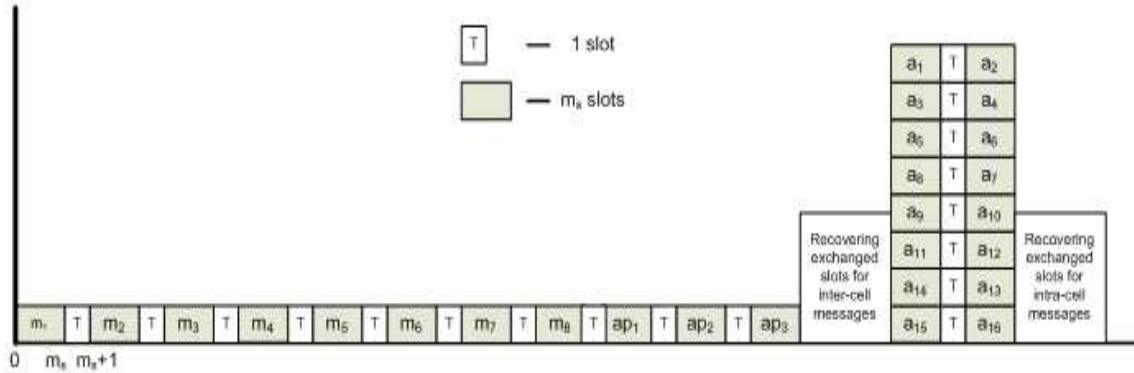


Figure 4.11 Schedule Produced by the Serial Offline Schedule Construction algorithm

- Serial Online Scheduling Algorithm:** Once the schedule formed offline is received by the nodes, they start transmitting their message in the scheduled time slots. However, when any message transmission begins to fail because of erroneous channel conditions, they request an exchange to be performed with another message. Several heuristics may be applied for the choice of a message to exchange with which is out of the scope of this paper. Exchanges are restricted to each level i.e. inter-cell messages exchange with other inter-cell messages and intra-cell messages exchange with other intra-cell messages in the same cell as explain in the Section 4.4.4.

Since the number of exchanges that can be performed at each level is limited and a set of slots equal to this number, called exchange pool, is reserved at the end of each intra cell and inter-cell periodic message schedules. Whenever an exchange is performed, an exchange slot is consumed where a data transmission slot is used for initiating an exchange. In order to preserve the schedulability guarantees, each node that performed the exchange exclusively gets a slot for data transmission from the exchange pool as explained in Section 4.4.2. To facilitate this, a transfer slot is needed at the end of each message for keeping track of the number of exchanges that has been performed in the

Input: Set of inter-cell message S_m , Set of intra-cell messages S_a , Total number of exchanges for inter-cell messages n_e, m , Number of exchanges for intra-cell messages in each cell n_e, a

Output: Serial Schedule SS where SS_i denote the set of messages scheduled for slot i

```

1 Set  $i = 1$ ;
2 for each msg in  $S_m$  do
3   | Add  $m_s$  slots of msg to  $SS_i \dots SS_i + m_s$ ;
4   | Set  $i = i + m_s$ ;
5   | Add 1 Transfer slot to  $SS_i$ ;
6   | Set  $i = i + 1$ ;
7 end
8 Add  $n_e, m$  exchange slots from  $SS_i$  to  $SS_i + n_e, m$ ;
9 Set  $i = i + n_e, m$ ;
10 NodeColoring( $S_a$ );
11 Set  $j = 1$ ;
12 while  $j \leq \lambda$  //For each color do
13   | For each msg in  $S_\lambda$ , add  $m_s$  slots of msg to  $SS_i \dots SS_i + m_s$ ;
14   | Set  $i = i + m_s$ ;
15   | Add 1 Transfer slot to  $SS_i$ ;
16   | Set  $i = i + 1$ ;
17   | Set  $j = j + 1$ ;
18 end
19 Add  $n_e, a$  exchange slots from  $SS_i$  to  $SS_i + n_e, a$ ;
20 Set  $i = i + n_e, a$  //This i forms  $n_s$ ;

```

Algorithm 4: Serial Offline Schedule Construction algorithm

system.

<p>Input: Serial Schedule SS, Set of locally available periodic messages $S_{l,p}$, Set of locally available aperiodic messages $S_{l,ap}$</p> <p>Output: Message Transmission</p> <pre> 1 Set $i = 1$; 2 while $i \leq n_s$ do 3 if any msg in $S_{l,p} \in SS_i$ then 4 Transmit msg; 5 If msg transmission unsuccessful, use exchange protocol to perform exchange and update SS accordingly; 6 end 7 else 8 if any msg in $S_{l,ap} \in SS_i$ and msg is ready then 9 Transmit msg; 10 If msg transmission unsuccessful, use exchange protocol to perform exchange and update SS accordingly; 11 end 12 end 13 else 14 if an exchange request message is received then 15 Use exchange protocol to determine if exchange is possible and update SS accordingly; 16 end 17 end 18 Set $i = i + 1$; 19 end </pre>

Algorithm 5: Serial Online Scheduling algorithm

A similar argument applies for intra-cell messages where each message can exchange with other intra-cell messages in the same cell without affecting the transmissions occurring in other cells. The number of exchanges in each cell is limited is kept the same for all the cells and an exchange pool is present for each cell that occurs at the end of the parallel intra-cell message schedule. This is feasible since the intra-cell transmissions occurring in each cell are independent of each other. There exists a transfer slot at the end of every intra-cell message which is used for keeping track of the exchanges that have been performed in each cell. Every data slot that has been used for an exchange is retrieved from the exchange pool exclusively since the number of exchanges is kept track of in a

distributed fashion as explained in the Section 4.4.2. The pseudocode for the serial online scheduling algorithm is shown in Algorithm 5.

Hence the serial scheduling mechanism allows for maximum exchanges to be performed when messages encounter erroneous channel conditions. When a particular channel condition deteriorates rapidly, an exchange can be performed with another channel which may be in the good state. There is a high chance of an erroneous message being successful when it is postponed to transmit at a later time since the channel might move to the good state by such time. This scheme is very effective for wireless channels with bursty error conditions where when a channel is in the bad state, it continues to be in the bad state for a long time and repeatedly trying to schedule the message over such a channel is wasteful. However, in the process of providing the capability for channel exchange, the parallelism that can be exploited for inter-cell messages is minimized and hence the total time taken for delivery of all the messages increases which is the drawback of this scheme.

4.5.4 Adaptive Protocol

The parallel protocol achieves most parallelism among messages but is unable to perform exchanges when the channel conditions are bad. On the other hand, the serial protocol produces a very pessimistic schedule that assumes worst channel conditions giving the luxury to perform any number of exchanges when the channel conditions are bad but fails to exploit parallelism between inter-cell messages even if none of them need to perform an exchange. The tradeoff between the parallelism and the capability to perform an exchange is bridged by the adaptive mechanism that dynamically adapts to the channel conditions at runtime. The scheduling algorithm moves towards the parallel schedule when the channel is in the good state and the serial schedule when in a bad state.

- **Adaptive Offline Schedule Construction Algorithm:**

The offline schedule that is constructed is similar that constructed for the serial algorithm described above. However, two extra slots are added to every inter-cell message as shown

in Figure 4.12 which are the control overheads incurred by the protocol. These slots enable parallel transmission of inter-cell messages to occur at run-time if the channel conditions are favorable and exchange is not initiated. At the same time, the serial schedule gives the capability to perform an exchange during an unfavorable channel condition. In addition to this schedule, the parallel schedule produced by the parallel protocol is also stored for reference and this parallel transmission schedule is achieved in case an exchange is not initiated. The pseudocode for the adaptive offline schedule construction algorithm is described in Algorithm 6.

Input: Set of inter-cell message S_m , Set of intra-cell messages S_a
Output: Adaptive Schedule AS where AS_i denote the set of messages scheduled for slot i

- 1 Set $i = 1$;
- 2 **for** each msg in S_m **do**
- 3 Add $m_s + 2$ slots of msg to $AS_i \dots AS_i + m_s + 2$;
- 4 Set $i = i + m_s + 2$;
- 5 **end**
- 6 NodeColoring(S_a);
- 7 Set $j = 1$;
- 8 **while** $j \leq \lambda$ //For each color **do**
- 9 For each msg in S_λ , add $m_s + 2$ slots of msg to $AS_i \dots AS_i + m_s + 2$;
- 10 Set $i = i + m_s + 2$;
- 11 Set $j = j + 1$;
- 12 **end**

Algorithm 6: Adaptive Offline Schedule Construction algorithm

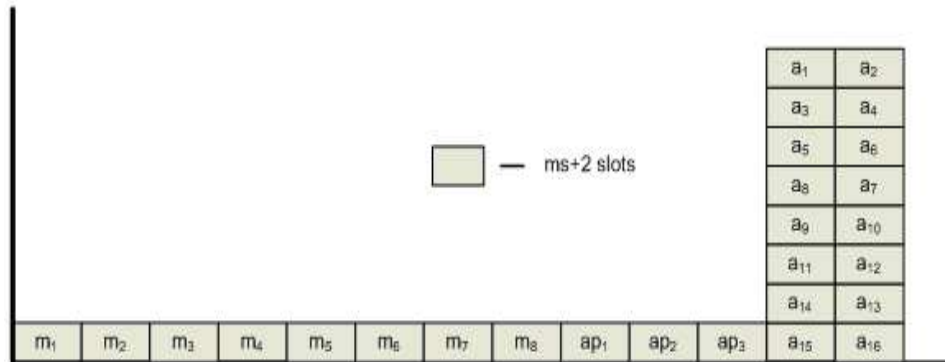


Figure 4.12 Schedule Produced by the Adaptive Offline Schedule Construction algorithm

- **Adaptive Online Scheduling Algorithm:**

During run-time, all the nodes adapt to the channel conditions in the following way. In the first of the control slot of an inter-cell message, the source of the message transmits an unit of data to the destination. If this transmission fails, an exchange is initiated in the next control slot. If the exchange has been successful, then the exchanged message transmits in the third slot. If data transmission in the first control slot is successful, then no exchange is initiated and the data transmission continues through the next two slots and proceeds according to the serial offline schedule. Nodes that are scheduled to transmit in parallel with this inter-cell message according to the parallel schedule listen to the channel for the 3 slot. If they find that no transmission in the vicinity occurs, the start transmitting in parallel with the inter-cell message. If any of the parallel messages observe a different transmission going on within their transmission range, they back-off from transmitting. In this way, a collision is avoided even when an exchange occurs and only those messages that are truly parallel with the exchanged message transmit. If the original inter-cell message proceeds without any exchange, all the parallel transmissions as per the parallel schedule start transmitting at the end of the three control slots and hence parallelism of the inter-cell messages is also achieved. In case of such parallel transmissions successfully completing their transmissions, the parallel messages would have their scheduled slots coming later according to the serial schedule. Whenever such a parallel transmission of an inter-cell periodic message occurs, the source of the transmission has its scheduled slots available for either re-sending the slots that failed in the initial periodic transmission or for sending aperiodic messages. If the initial periodic message transmission was successful, the scheduled transmission time for the periodic message can be used to transmit the aperiodic message in a collision-free manner. In case the initial periodic transmission was unsuccessful, the scheduled transmission time is used initially to complete the successful reception of the periodic message and then aperiodic messages, if any are transmitted by the source to any destination. When an aperiodic transmission occurs, the messages that are parallel to the original scheduled transmission according to the serial schedule wait to

listen to the channel in the control slots. If any of the messages find another transmission in their vicinity proceeding, they realize that either an exchange or aperiodic transmission is in progress and backoff. If any of such parallel message do not hear of any transmission in the vicinity, it starts transmitting without colliding with the original transmission and hence conflicts are avoided in all cases. The pseudocode for the Adaptive Online Scheduling algorithm is shown in Algorithm 7.

Consider an exchange occurring between two messages m_i and m_j . If an exchange occurs, the current message transmits during the scheduled time of the exchanged message. The parallel messages of this exchanged message listen to the channel during the 3rd control slot during which time they may hear the transmission of a different (exchanged) message. If so, they back-off their transmission. If there is some message amongst the set of parallel messages that do not hear of the exchanged messages transmission, they would think that the original message is transmitting and would try to transmit parallelly. If the exchanged message and the parallelly transmitting message are out of range of each other, they would be able to proceed parallelly without any conflicts. However, there might be a case where the source of the message transmission does not hear of the exchanged message transmission and transmit to the destination which is in the vicinity of the exchanged message. In this case, the destination would know of the ongoing exchanged message transmission and not send an ACK since it would conflict with the message. In this case, the parallel message would have no ACK sent to the source. Hence, conflicts of any kind are avoided and exchanged message and a message transmitting without any exchanges all have their schedulability guarantees being met. At the same time, parallelism is also exploited that gives the messages that complete their transmission early to transmit aperiodic messages. In this way, aperiodic schedulability is improved and periodic loss rate is decreased by giving multiple chances for periodic message transmission. Hence, the full capability of the serial schedule to perform exchanges is exploited while ensuring that parallel transmissions are enabled whenever the channel is in a good state. Essentially, this protocol starts off with a pessimistic serial schedule and moves towards the parallel

Input: Adaptive Schedule AS , Set of locally available periodic messages $S_{l,p}$, Set of locally available aperiodic messages $S_{l,ap}$, $P(m_i)$ that gives the set of messages that can transmit in parallel with each inter-cell message m_i

Output: Message Transmission

```

1 Set  $i = 1$ ;
2 while  $i \leq (m_s + 2) * Size(S_m)$  do
3   Set  $j = 1$ ;
4   while  $j \leq m_s + 2$  do
5     if any msg in  $S_{l,p} \in AS_i$  then
6       if  $j = 1$  then
7         Transmit msg;
8         If Transmission unsuccessful, perform exchange in slot  $i + 1$  and if
           exchange successful, barter next  $m_s$  slots with exchanged message;
9       end
10      else
11        Transmit message; //any other transmission slot;
12      end
13    end
14    else
15      if any msg in  $S_{l,ap} \in AS_i$  and msg is ready then
16        Repeat steps 6-12;
17      end
18    end
19    else
20      if any msg in  $S_{l,p}$  or  $S_{l,ap} \in P(m_i)$  where  $m_i \in AS_i$  and  $j = 3$  then
21        If no transmission in the vicinity, it means no conflicting exchange has
           been performed. Update  $AS_{i+1} \dots AS_{i+m_s}$  with msg;
22      end
23    end
24    else
25      if any msg sends request for exchange then
26        If exchange is possible, barter last  $m_s$  slots of message;
27      end
28    end
29    Set  $i = i + 1$ ;
30    Set  $j = j + 1$ ;
31  end
32 end
33 Repeat steps 2-19 from Serial Online Scheduling algorithm;

```

Algorithm 7: Adaptive Online Scheduling algorithm

schedule whenever channel conditions permit.

As for intra-cell messages, they might schedule parallelly with other inter-cell messages. Otherwise they transmit in their scheduled slots along with intra-cell messages in other cells. If they have transmitted parallelly with another inter-cell message, then their scheduled slot is used to transmit an aperiodic message, if any, to any other node in the same cell. Since for every transmission slot, there can only be one message in a cell, no collisions would arise by a transmission to a different destination in the cell. In this way, any aperiodic intra-cell message is also supported by this protocol. Hence to summarize, in this scheme, aperiodic inter-cell messages are supported when a message transmits parallelly with another message and in the process gets its scheduled slots as per the serial schedule available for aperiodic message transmission to any destination. Aperiodic intra-cell message transmission to any destination in the same cell is enabled when a scheduled intra-cell messages transmits parallelly with another inter-cell messages and has its scheduled slots available for aperiodic message transmission.

From the schedule in Figure 4.10, for each of the inter-cell messages m_i , we can determine the set of messages $P(m_i)$ transmitting in parallel with m_i . These are summarized in the Table 4.3.

Table 4.3 P_i that represents the set of messages transmitting in parallel with every message m_i

m_i	$P(m_i)$
m_1	m_6, a_7
m_2	m_7, a_10
m_3	m_8, a_11
m_4	ap_3
m_5	a_1, a_3, a_5, a_15
m_6	m_1, a_7
m_7	m_2, a_10
m_8	m_3, a_11
ap_1	a_13, ap_2
ap_2	a_13, ap_1
ap_3	m_4

Hence when the channel conditions are good and none of the inter-cell messages initiate exchange in the first time slot of their scheduled transmission, the final schedule using the adaptive algorithm that is followed by the system will be as shown in Figure 4.13. Each of the messages in $P(m_i)$ listen to the channel when m_i is transmitting in the first 3 slots. If they hear no message transmission in the vicinity, it means no exchange that is conflicting with any of the message transmission in $P(m_i)$ has been performed and all messages in $P(m_i)$ start transmitting in parallel for the last m_s-1 slots.

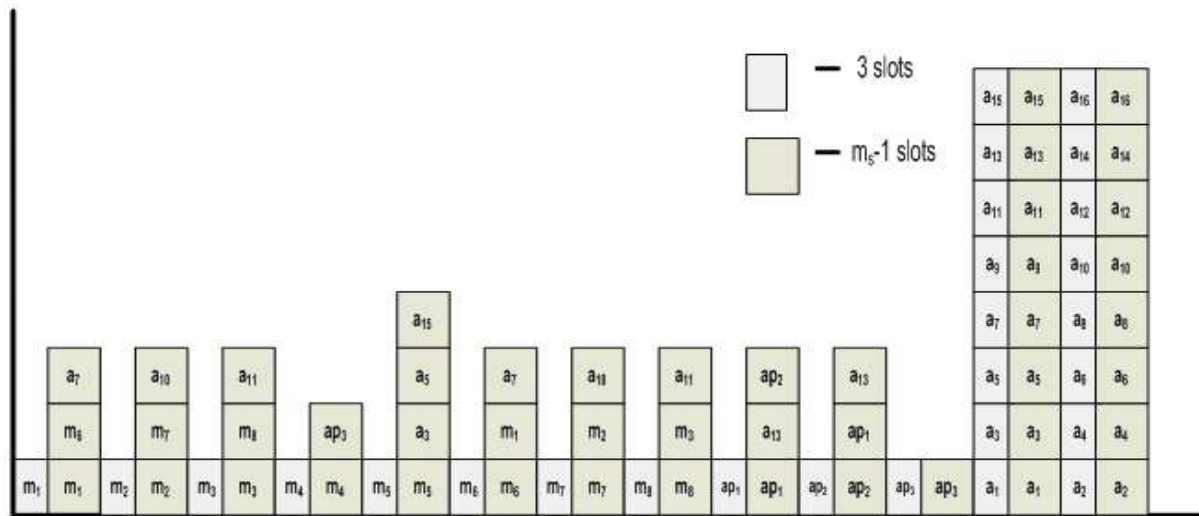


Figure 4.13 Adaptive Run-time schedule followed when no exchange is initiated

Hence for a channel in good state, maximum parallelism is exploited as can be seen from the Figure 4.13. This is the best case schedule produced by the adaptive protocol. In case the channel condition deteriorate and exchange are performed, it maybe possible that only a sub-set of messages in each $P(m_i)$ can transmit in parallel with the message m_i since some of them may collide with the transmission of the exchanged message and would have performed a back-off. Consider the exchanges shown in the Table 4.4 that can be initiated by each message. If all of the inter-cell messages perform exchange in the first slot of their scheduled transmission, only the sub-set of messages can be transmitted with each of the inter-cell message. We denote $P_e(m_i)$ to be the set of messages that transmit in parallel when the message m_i has exchanged

its slots and the exchanged message is transmitting. This is shown in the Table 4.4. The schedule followed will look as shown in Figure 4.14:

Table 4.4 Exchanged messages and messages that can transmit in parallel with the exchanged message

m_i	Exchanged message $E(m_i)$	$P_e(m_i)$
m_1	m_4	-
m_2	m_8	a_{10}
m_3	m_7	-
m_4	m_1	-
m_5	m_4	a_3, a_{15}
m_6	m_5	m_1
m_7	m_3	-
m_8	m_2	a_{11}
ap_1	ap_2	a_{13}
ap_2	ap_1	-
ap_3	-	-

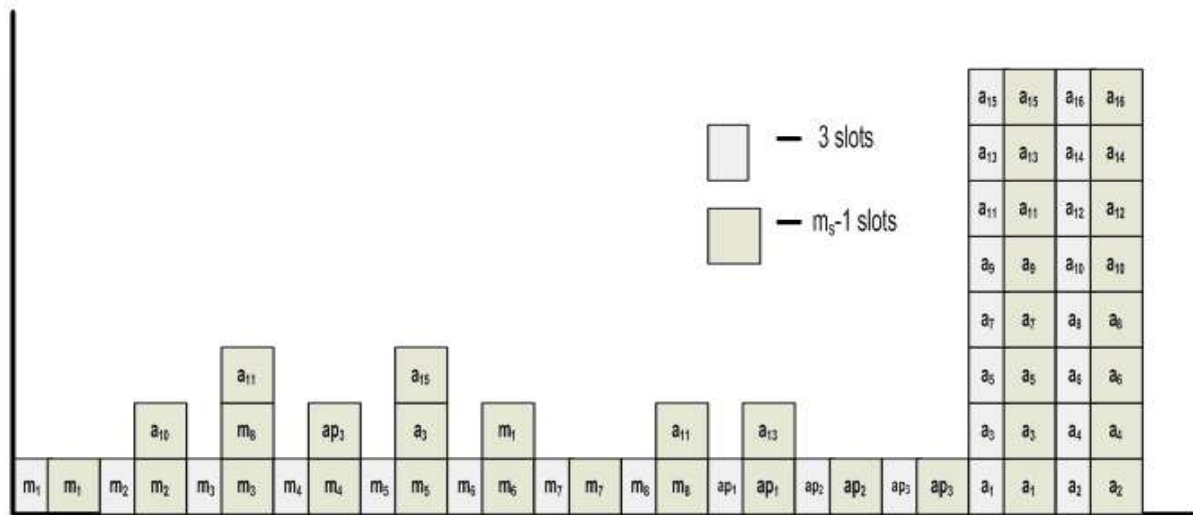


Figure 4.14 Adaptive Run-time schedule followed when exchanges are initiated

Even after performing exchange, the adaptive schedule exploits certain degree of parallelism between the messages. Note that the intra-cell messages can perform exchange the usual way but they have to initiate exchange based on the channel conditions in the first of their scheduled transmission slot. Every intra-cell message performs exchange with other intra-cell messages

in the same cell and exchanges the last ms slots of the message.

The two control slots used by the protocol are equivalent to the serial protocol that enables one exchange to be performed by each message. The two slots account for the transfer and the exchange slot for each message that is used by the serial protocol. This helps accommodation the same messages between the serial and the adaptive protocol in a single super-frame. In the serial protocol, exchange can be initiated at any point in time during the message transmission. Even though the allocation of exchange slot is one for each message in the system, the serial protocol does not enforce this and one message can perform several exchanges based on the channel condition. However, in the adaptive scheme, only one exchange can be performed by each message and an exchange is performed based on the state of the channel in the first slot. Thus the adaptive protocol removes the drawbacks of the serial protocol that produces a trashing effect while performing multiple exchanges when all the channel condition is bad that is not fruitful. Also, observing the channel condition in the first slot in the first slot provides for a good estimation of the channel conditions during the entire message transmission as can be seen from the Simulation studies. In addition, the adaptive protocol used parallel transmissions that increases the number of messages meeting their deadline and provides for better channel utilization.

Hence, from the explanation of the three scheduling strategies, it is clear that the algorithm adopted for the offline schedule construction has an impact on the run-time channel adaptation techniques that can be used. A completely parallel offline schedule yields no exchanges and hence no channel adaptation at run-time and a completely serial schedule permitting exchanges offers very little parallelism during run-time. The former strategy is followed by the parallel protocol and the latter by the serial protocol. The adaptive algorithm works with the serial schedule produced offline but at run-time exploits as much parallelism as can be obtained based on the channel conditions. Thus the adaptive algorithm removes the weaknesses of both the parallel and serial algorithms during run-time and thus minimizes the loss rate and improves the schedulability of periodic messages as well as aperiodic when the channel conditions permit. We show these performance gains by means of our simulation results . We also try to determine

a clairvoyant algorithm that has complete knowledge of the channel conditions and schedules messages in the best possible manner with this information. We show that this is a NP-hard problem. We propose a heuristic algorithm that schedules both periodic and aperiodic messages given complete knowledge of the channel conditions and message arrival times.

4.5.5 Clairvoyant Protocol

The clairvoyant algorithm has complete knowledge on the messages and channel conditions. With this knowledge, it tries to schedule messages such that the maximum numbers of aperiodic and periodic messages succeed in the system. Consider a particular case of this problem when all the channel conditions are good. The problem now is to schedule a set of aperiodic and periodic messages such that maximum parallelism is exploited and the messages complete within minimum slots. This would reduce to the node-coloring problem that is NP-hard. Hence the clairvoyant problem is also NP-hard. We propose a heuristic solution to the problem. For each slot in the super-frame, the heuristic algorithm obtains the set of messages that need to be transmitted. From this set, it determines the messages whose channels conditions are good in the current slot such that the corresponding message transmission will be successful. Constructing a conflict graph with these set of messages and node coloring the graph produces a parallel transmission schedule. In order to determine the transmissions that need to occur in the current slot, we obtain the color in the schedule which has the maximum set of message in it and transmit those message. By repeating this algorithm for every slot in the super-frame, a schedule where a lot of periodic and aperiodic messages are scheduled is obtained. The pseudocode for the Clairvoyant protocol is shown in Algorithm 8. This heuristic clairvoyant algorithm can form the baseline for the comparison of the three proposed protocols. We however, do not simulate the algorithm in our work.

4.6 Simulation Studies

We simulate the hierarchical network shown in Figure 4.1 comprising of 24 nodes of which eight inter-cell nodes and 14 intra-cell nodes with each inter-cell node having two intra-cell node

Input: Set of messages that need to be scheduled in each slot S_i , Channel conditions

Output: Message Transmission

```

1 Set  $i = i$ ;
2 while  $i \leq n_s$  do
3   From  $S_i$ , determine the set of messages whose channel is in good state,  $S_{ch}, i$ ;
4   NodeColoring( $S_{ch}, i$ );
5   Obtain  $\lambda$  comprising of the maximum number of messages;
6   Transmit messages belonging to this  $\lambda$ ;
7 end

```

Algorithm 8: Heuristic Clairvoyant Algorithm

in its cell. This network is representative of a typical hierarchical network in a factory floor with functionality clustered into cells comprising of a controller(inter-cell node) and several sensors and actuators(intra-cell nodes) sending information to it. This communication pattern is analogous to a PROFIBUS master-slave setup described where several slaves send messages to the master and the masters arranged in a ring communicate with the next masters in the ring. This setup will be used for the initial migration from wired to wireless environments in a factory floor. Each node in the system has a periodic message to transmit and three of the inter-cell nodes also have dynamically arriving aperiodic messages. We fix the size of each message to 10 slots(1500 bits) and the total slots in the superframe to 156. We assume that there are infinite number of aperiodic messages with the inter-cell nodes and all the periodic messages are ready at the beginning of the super-frame. We consider the deadline to be the end of the super-frame. Our objective is to compare the relative performance of the three protocols on this message set and observe the success rate for varying channel conditions. We use a standard DSTAUR node coloring algorithm implementation for generation the parallel schedule of this message set. Increasing or decreasing the number of messages will affect the three protocols in a deterministic manner and hence this message set to study the relative performance was considered to be sufficient. Each of the protocol was simulated for 10,000 super-frame runs.

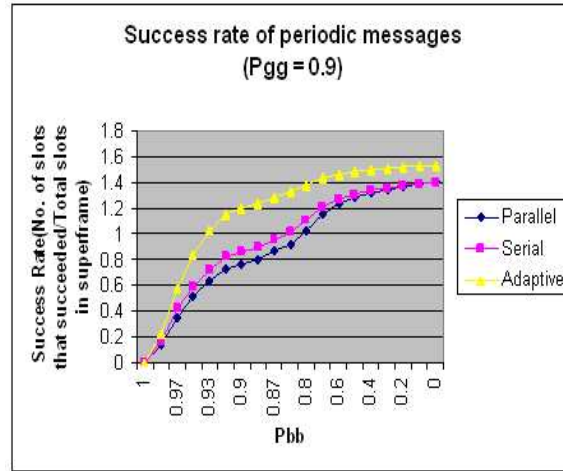


Figure 4.15 Success rate of periodic messages with variation in P_{bb}

4.6.1 Effect of varying P_{bb} on the success rate of periodic messages

A decrease in P_{bb} denotes that the channel moves out of the bad state quickly and all the three schemes show an increase in the success rate with decreasing P_{bb} as shown in Figure 4.15. The success rate is greater than 1 for all the 3 schemes because of the parallel transmissions capability exploited by all the 3 schemes (Even in serial, the intra-cell messages are transmitted in parallel). When the probability of staying in the bad state having once entered it is high (outage condition when $P_{bb} = 1$), none of the slots on any of the channels are able to succeed and hence the performance of all the three schemes remains the same. As the channels conditions become better with decreasing P_{bb} , the serial scheme performs better than the parallel scheme since whenever a scheduled channel is bad, it trades it with a channel in the good state giving rise to the performance gains. The adaptive schemes performance is significantly higher when compared to either the parallel or serial since apart from using the exchange technique, it enables as many parallel transmissions as possible and since each message is dynamically given more slots for data transmission, the success rate increases. As P_{bb} approaches 0, whenever the channel moves into the bad state with a probability of 0.1, it immediately moves back into the good state. In such a scenario, for the parallel scheme, some slots are lost while the channel is in the bad state. During these conditions, the serial scheme tries to perform an exchange that

is not too beneficial. In case of the adaptive scheme, the number of exchanges carried out by the scheme is limited and the loss of slots while in the bad state is offset by the additional slots that each message gets by transmitting parallelly that accounts for the better performance of the adaptive scheme than the serial and parallel schemes. Also each message in the adaptive scheme gets more slots than the actual message size by the nature of the protocol that accounts for this difference. The performance gap narrows as P_{bb} approaches 0 since the success rate is high for all of the schemes since the channel conditions are good and no more performance improvement can be derived by any of the schemes. The success rate of periodic inter-cell and intra-cell messages also show a similar trend as seen from Figure 4.16.

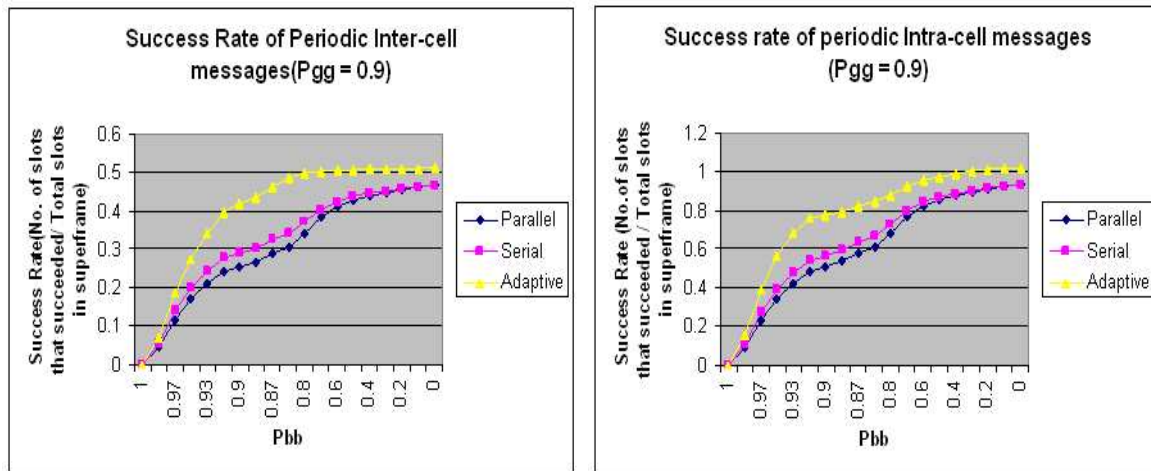


Figure 4.16 Success rate of inter-cell and intra-cell periodic messages with variation in P_{bb}

4.6.2 Effect of varying P_{bb} on the success rate of aperiodic messages

From Figure 4.17 we can observe that as P_{bb} decreases, the success rate improves for all of the schemes since the channel stays in the bad state for shorter periods of time. The success rate of aperiodic messages is greatest in the parallel scheme. When all the periodic messages are scheduled once, the remaining slots are used for the transmission of aperiodic messages and hence more aperiodic messages are scheduled and meet their deadlines. In case of the adaptive scheme, the aperiodic messages are transmitted once in their scheduled time frame and also

exploit any parallel transmissions that may be possible and hence its success rate is better than the serial scheme but poorer than the parallel scheme. In the serial scheme, the aperiodic slots are scheduled only once every superframe which does not lead to significant gains.

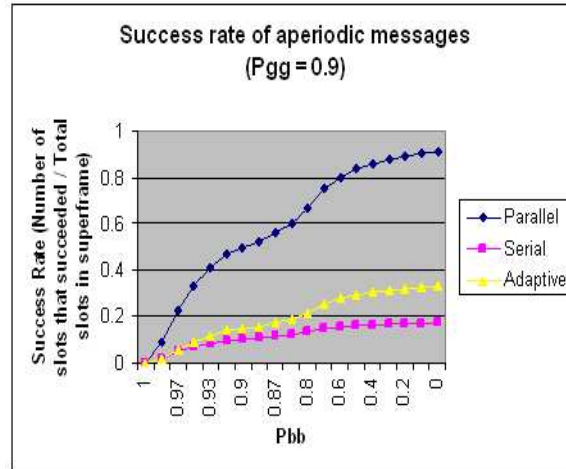


Figure 4.17 Success rate of aperiodic messages with variation in P_{bb}

4.6.3 Effect of varying P_{bb} on the exchange rate of messages

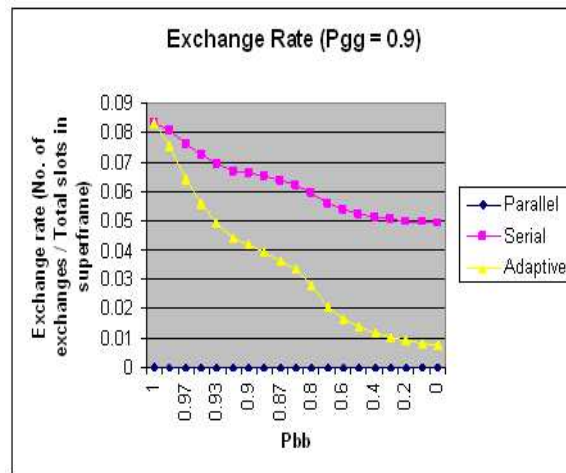


Figure 4.18 Exchange rate of messages with variation in P_{bb}

As P_{bb} decreases, the channel stays in the bad state for lesser time and since an exchange is initiated only when the channel is in bad state, the exchange rate decreases as P_{bb} decreases

as shown in Figure 4.18. For the parallel schemes, no exchanges are performed and hence the exchange rate is 0. When P_{bb} approaches 1, the exchange rate is the same for both the serial and adaptive scheme since we do not allow nested exchanges in the serial scheme. Hence, in the first few scheduled timeslots of every message, an exchange is performed by both the adaptive and the serial scheme and hence the two schemes converge on the exchange rate. The adaptive scheme performs exchanges only if the first time slot of a message transmission is noisy while the serial scheme can perform exchange at any point during the message schedule. As P_{bb} decreases, the probability that the first time slot of any message is unsuccessful decreases and hence the exchange rate of the adaptive scheme is lesser than the serial scheme. This is the trend that we want since we do not want too many exchanges happening when the channel occasionally moves to the bad state while for large periods, it remains in the good state (P_{bb} approaching 0) but the serial scheme does not cache on this and performs exchanges every time the channel moves to the bad state increasing the overhead. The rate of exchanges performed by inter-cell and intra-cell messages also shown a similar trend as can be seen from Figure 4.19. The rate of intra-cell exchanges is more than the inter-cell exchange rate since there are more intra-cell messages in the system.

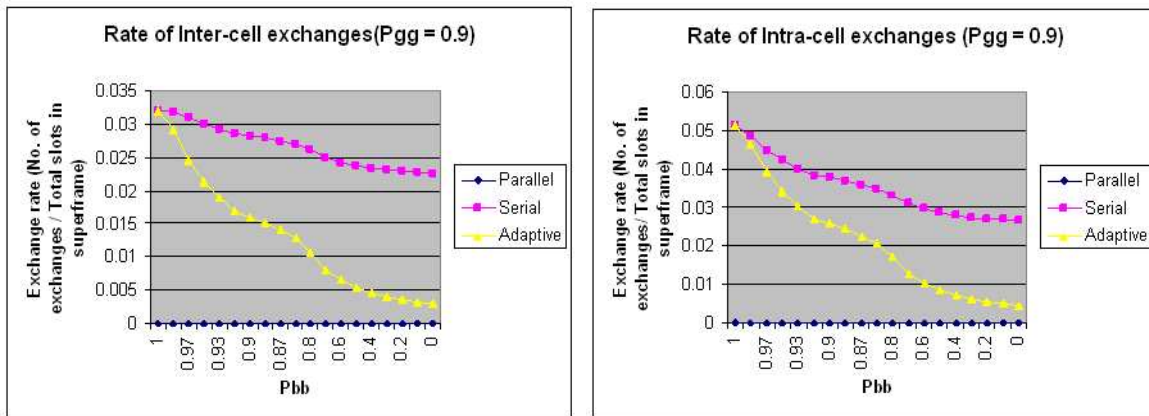


Figure 4.19 Rate of Inter-cell and Intra-cell exchanges with variation in P_{bb}

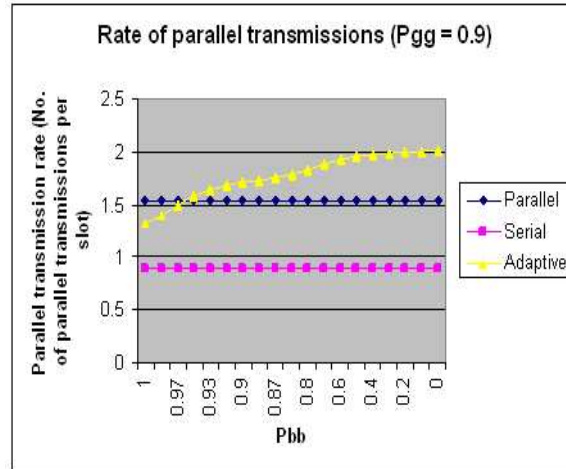


Figure 4.20 Rate of parallel transmissions with variation in P_{bb}

4.6.4 Effect of varying P_{bb} on the rate of parallel transmissions

This graph in Figure 4.20 shows the average number of parallel transmissions per slot. Since the intra-cell messages are transmitted in parallel in the serial scheme, it has a constant parallel transmission rate. The parallel scheme follows a fixed parallel schedule and hence has a non-varying parallel transmission rate. For the adaptive scheme, as P_{bb} decreases, the number of parallel transmissions occurring per slot increases. At low values of P_{bb} , an exchange is performed by inter-cell messages and lesser message transmit in parallel with the original scheduled message during the inter-cell message transmission than the case at higher values of P_{bb} when no exchange is performed and more messages are able to transmit in parallel; hence the increasing trend in parallel transmissions. The rate of parallel transmissions is higher than the parallel scheme since in the parallel scheme, all messages are scheduled parallelly once after which aperiodic inter-cell messages are transmitted that are few in number and lesser parallel transmissions occur while aperiodic messages are transmitted however maximum parallel transmissions is exploited throughout the superframe by the adaptive scheme that accounts for the increased parallel transmission rate. At low values of P_{bb} , the rate of parallel transmissions per slot is lower for the adaptive scheme than the parallel scheme since in the adaptive scheme, lesser parallel transmissions occur per slot as the parallel transmissions occur

for one slot lesser than the parallel scheme and since most of the channels are in bad state, the number of parallel transmissions occurring for inter-cell messages is lower for all message transmissions. As P_{bb} increases, more channels are in good state and hence more parallel transmissions are exploited by the adaptive scheme and its performance exceeds that of the parallel scheme.

4.6.5 Effect of varying P_{gg} on the success rate of periodic messages

As P_{gg} increases, the success rate increases since the channel remains in good state for longer periods of time as shown in Figure 4.21. At low values of P_{gg} , the performance of all the schemes is similar since all the channels are in the bad state and the success rate is poor. As P_{gg} increases, the adaptive scheme outperforms the parallel and serial scheme since exploits both parallel transmissions as well as exchanges when the channel is in bad state. At high values of P_{gg} when all the channels are in the good state, the performance of all of the schemes converges since all of the transmissions succeed in all the schemes. From Figure 4.22, we can see that the success rate of periodic inter-cell and intra-cell messages also follow a similar trend.

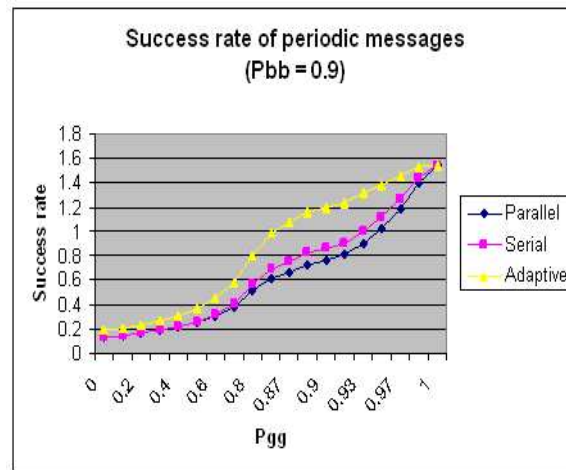


Figure 4.21 Success rate of periodic messages with variation in P_{gg}

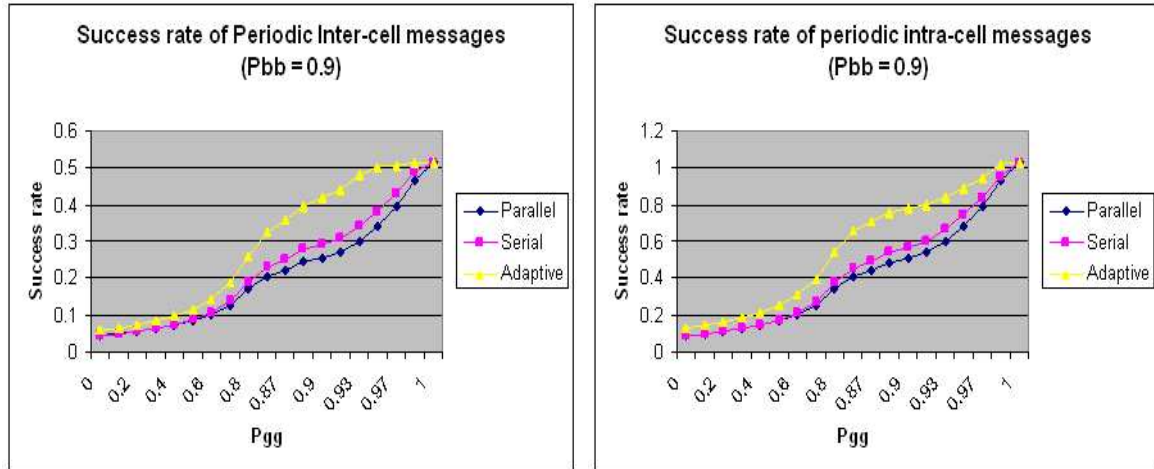


Figure 4.22 Success rate of Inter-cell and Intra-cell periodic messages with variation in P_{gg}

4.6.6 Effect of varying P_{gg} on the success rate of aperiodic messages

As P_{gg} increases, the success rate of aperiodic messages increases in all the schemes as shown in Figure 4.23. Success rate is highest for the parallel scheme since aperiodic messages are given more slots to transmit. The adaptive scheme outperforms the serial scheme at higher values of P_{gg} since it exploits both parallel transmissions of aperiodic messages that increases the success rate.

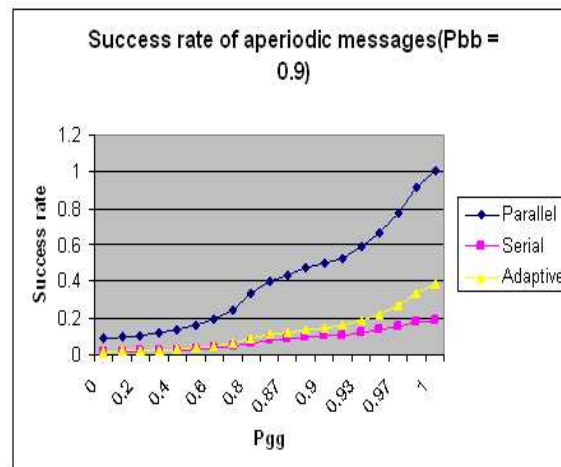


Figure 4.23 Success rate of aperiodic messages with variation in P_{gg}

At low values of P_{gg} , whenever the channel is in good state, it moves quickly back into the bad state and hence neither an exchange nor parallel transmissions are effective and hence the poor performance of both the serial and the adaptive schemes. The parallel schemes performance is better even at low values of P_{gg} since several slots of the message are allowed to be transmitted that far exceeds the message size.

4.6.7 Effect of varying P_{gg} on the exchange rate of messages

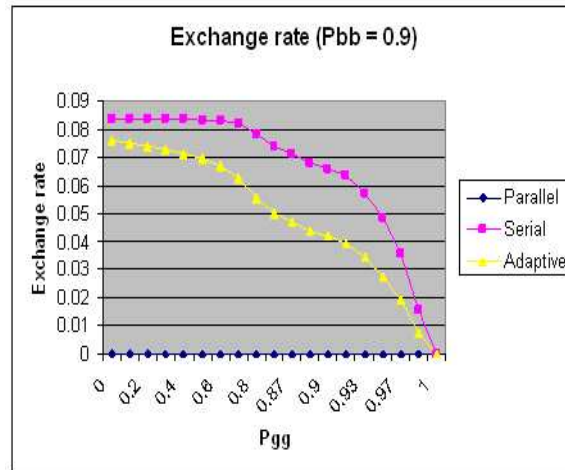


Figure 4.24 Exchange rate of messages with variation in P_{gg}

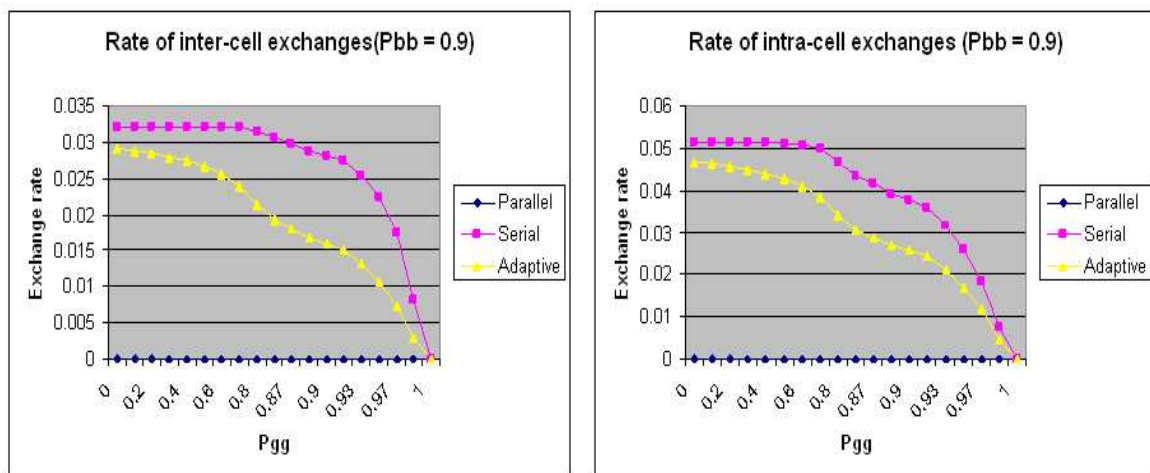


Figure 4.25 Rate of Inter-cell and Intra-cell exchanges with variation in P_{gg}

As P_{gg} increases, the exchange rate decreases as shown in Figure 4.24 since most of the channels are in a good state and no exchange is initiated. The exchange rate is greater for serial than the adaptive scheme since the exchange can be initiated by the serial scheme at any time slot of the scheduled message transmission when the channel is in the bad state; however for the adaptive scheme, the exchange is initiated based on the channel condition observed during the first slot of the message transmission. As P_{gg} approaches 1, both the serial, parallel and adaptive schemes converge since not many exchanges are initiated due to the good channel conditions. The rate of Inter-cell and Intra-cell exchanges also show a similar trend with the rate of Intra-cell exchanges being higher because of the presence of many more Intra-cell than Inter-cell messages in the system.

4.6.8 Effect of varying P_{gg} on the rate of parallel transmissions

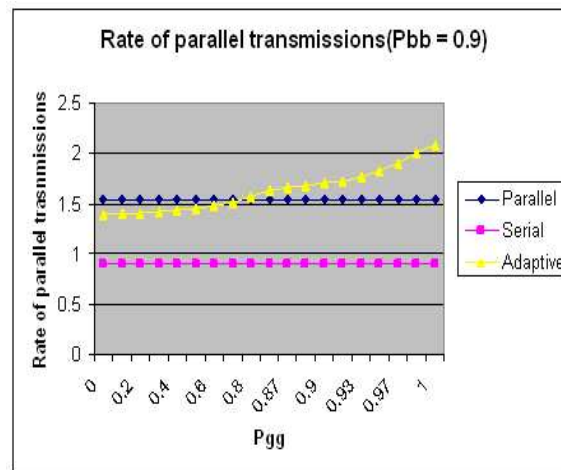


Figure 4.26 Rate of parallel transmissions with variation in P_{gg}

The rate of parallel transmissions is constant for both the parallel and the serial scheme as shown in Figure 4.26. The parallel scheme follows a static parallel transmission schedule while the serial scheme follows a parallel transmission schedule only for intra-cell messages. For the adaptive scheme, the parallel transmission schedule varies based on the state of the channel particularly during the transmission of the inter-cell messages. When a scheduled inter-cell messages transmission is successful, the messages that would have transmitted parallel with

the inter-cell message as per the parallel scheme is used. However, if an exchange is performed, only a subset of these messages would transmit in parallel. Hence as P_{gg} increases, more of the channels are in good state and the entire set of parallel messages are able to transmit that is shown by the increase in the rate of parallel transmissions. At lower values of P_{gg} , more exchanges are performed and the rate of parallel transmissions is even lesser than the parallel scheme.

4.7 Contributions and Future work

In this paper, we proposed a framework for scheduling real-time messages over a hierarchical industrial network and extended the exchange protocol to this network setup. The contributions of this work are as follows:

- Framework for combined scheduling of periodic and aperiodic messages over a 2-hop hierarchical wireless industrial network
- Identifying the trade offs between maximizing the channel utilization using parallel transmissions and using the exchange protocol to adapt to the bursty channel error characteristics
- Formulating the conditions for using the Exchange protocol in a parallel message setup
- Presenting the adaptive protocol that bridges the tradeoff between the use of the exchange protocol and the amount of parallelism that can be exploited in the network
- Comparison with a protocol that achieve maximum parallelism and one which ensures complete exchange

The salient features of the adaptive protocol are as follows:

- Specifically tailored for the communication between field devices like sensors/actuators and controllers which are organized in clusters in an industrial environment

- Switches between exploiting maximum parallelism when the channel conditions are good and enabling exchanges by using the Exchange Protocol when the channel conditions are bad
- Starts with a pessimistic schedule expecting bad channel conditions and moves to an optimistic schedule that exploits parallel transmissions based on the channel conditions
- Adapts to the channel conditions in a distributed manner
- Preserves the schedulability guarantees given to the messages
- Ensures that transmissions proceed in a collision-free environment
- Achieves better success rate for both periodic and aperiodic messages and good channel utilization compared to the two baseline protocols

As part of our future work, we plan to extend the protocol by incorporating techniques that provide guarantees for complete message transmissions. We would also like to extend the adaptive protocol to a mobile environment where the network topology changes with time. Another future direction of research is to explore new heuristic node coloring algorithms that will produce parallel schedules taking into account the constraints of exchange.

CHAPTER 5. Conclusions and Future Work

The use of communication networks in the industry is heavily influenced by the peculiar requirements of the application environment. In particular, if the network is employed at the lowest level of factory communication systems, also called the field level, where fieldbuses are traditionally used, it has to be able to cope with the real-time communication between controllers and sensors/actuators. As an additional constraint, industrial plants are known to represent an hostile environment for communication systems due to the presence of several kinds of electromagnetic noise. Moreover, it has to be considered that, often the environmental conditions are characterized by the presence of dust, vibrations and other adverse factors. In the past, the aforementioned considerations were sufficient to discourage the use of wireless systems for industrial applications. Recently, however, there has been an impressive growth in the deployments of such networks in several fields of application, which has led to its increased dependability, performance improvement and cost reduction. Unfortunately, wireless links are much more error-prone than wired links and the channel conditions vary with time which can harm the reliable transmission of messages and hence the real-time requirements of the application. Therefore, transmission reliability needs to be improved substantially to use wireless media for real-time industrial applications. Additionally, the protocols developed need to take the error characteristics of the wireless links explicitly into account.

Towards this effort, we propose a framework for combined scheduling of periodic and aperiodic messages in a wireless industrial network. We propose the exchange protocol dynamically adapts to the bursty channel error conditions exploiting the spatial and temporal diversity characteristics of the wireless medium and improves the success rate of messages. We extend the exchange protocol to a 2-level hierarchical network support and improve channel utiliza-

tion by means of enabling parallel transmissions. Because of the inherent trade off between the use of the exchange protocol and the ability to maximize parallel transmissions, we propose an adaptive protocol that provides the ability to perform exchanges as well as enable parallel transmissions based on the channel conditions. Simulation studies show that the protocol offers significant performance gains in terms of the success rate of messages compared to baseline protocols.

As part of our future work, we plan to formulate the MAC scheduling problem to a n-level hierarchical network and develop novel scheduling algorithms based on the slot-exchange technique that maximizes channel utilization. In addition, we also intend to extend the scheduling problem to account for node mobility and develop mobility-aware MAC protocols. In this scenario, the conflict graph of the network would vary with time and the parallel and Adaptive protocols proposed in Chapter 4 need to be extended to account for this topology variation with time. Another direction of research is to design energy-aware protocols that are capable of adapting to the channel conditions using the exchange mechanism. In the current work, all of the nodes need to be awake since they might receive a request for exchange. However, many nodes in an industrial environment would be battery-powered and such a technique would reduce the life-span of the nodes. Hence, another direction of research is to develop channel-adaptive MAC scheduling algorithms that combat the erroneous channel conditions as well as utilize minimum energy to extend the lifetime of the network.

BIBLIOGRAPHY

- [1] Digital data communications for measurement and control - fieldbus for use in industrial control systems - parts 3 and 4. *International Electrotechnical Commission Std*, January 2000.
- [2] Final draft international standard: Profile sets for continuous and discrete manufacturing relative to fieldbus use in industrial control systems. *International Electrotechnical Commission Std*, February 2002.
- [3] A.El-Hoiydi and J.D.Decotignie. Soft deadline bounds for two-way transactions in bluetooth piconets under co-channel interference. *Proceedings of the 8th IEEE International Conference on Emerging Technologies and Factory Automation*, June 2001.
- [4] L. Almeyda, E. Tovar, J. A. G. Fonseca, and F. Vasques. Schedulability analysis of real-time traffic in worldfip networks: an integrated approach. *IEEE Transactions on Industrial Electronics*, October 2002.
- [5] M. Alves, E. Tovar, F. Vasques, G. Hammer, and K. Rother. Real-time communications over hybrid wired/wireless profibus-based networks. *Proceedings of 14th Euromicro Conference on Real-Time System*, June 2002.
- [6] A.Vedral and J.F.Wollert. Analysis of error and time behavior of the ieee 802.15.4 phy-layer in an industrial environment. *IEEE International Workshop on Factory Communication Systems*, June 2006.
- [7] A.Willig. Analysis of the profibus token passing protocol over wireless links. *Proceedings of the 2002 IEEE International Symposium on Industrial Electronics*, June 2002.

- [8] P. Bhagwat, P. Bhattacharya, A. Krishna, and S. Tripathi. Enhancing throughput over wireless lans using channel state dependent packet scheduling. *Proceedings of the IEEE INFOCOM*, March 1996.
- [9] D. Brelaz. New methods to color the vertices of a graph. *Communications of the ACM*, April 1979.
- [10] S. Cavalieri, A. D. Stefano, and O. Mirabella. Optimization of acyclic bandwidth allocation exploiting the priority mechanism in the fieldbus data link layer. *IEEE Transactions on Industrial Electronics*, June 1993.
- [11] G. Cena, C. Demartini, and A. Valenzano. On the performances of two popular fieldbuses. *Proceedings of WFCS*, May 1997.
- [12] L. Ciminiera, C. Demartini, and A. Valenzano. Industrial iee 802.3 networks with short delivery time for urgent messages. *IEEE Transactions on Industrial Electronics*, February 1988.
- [13] D.Dzung, C.Apneseth, J.Endresen, and J.Frey. Design and implementation of a real-time wireless sensor/actuator communication system. *10th IEEE Conference on Emerging Technologies and Factory Automation*, September 2005.
- [14] D.ellegrini, F.Miorandi, D.Vitturi, and S.Zanella. On the use of wireless networks at low level of factory automation systems. *IEEE Transactions on Industrial Informatics*, May 2006.
- [15] D. Duchamp and N. Reynolds. Measured performance of wireless lan. *Proceedings of the 17th Conference on Local Computer Networks*, July 1992.
- [16] D. A. Eckhardt and P. Steenkiste. A trace-based evaluation of adaptive error correction for a wireless local area network. *MONET -Mobile Networks and Applications*, January 1999.

- [17] E.O. Elliot. Estimates of error rates for codes on burst-noise channels. *Bell Syst. Tech.*, September 1963.
- [18] E.Uhlemann, T.Aulin, T.Rasmussen, and L.K.Wiberg. Concatenated hybrid arq - a flexible scheme for wireless real-time communication. *Proceedings of the Eighth IEEE Symposium on Real-Time and Embedded Technology and Applications*, February 2002.
- [19] L. Ferreira, M. Alves, and E. Tovar. Hybrid wired/wireless profibus networks supported by bridges/routers. *Proceedings of WFCS*, August 2002.
- [20] E.N. Gilbert. Capacity of a burst-noise channel. *Bell Syst. Tech.*, September 1960.
- [21] H.Zhu and I.Chlamtac. Performance analysis for ieee 802.11e edcf service differentiation. *IEEE International Workshop on Factory Communication Systems*, July 2005.
- [22] J.A.Carvalho, A.S.Carvalho, and P.J.Portugal. Assessment of profibus networks using a fault injection framework. *10th IEEE Conference on Emerging Technologies and Factory Automation*, September 2005.
- [23] J.G.Castano, J.Andreasson, M.Ekstrom, A.Wrzesniewski, H.Ahlblom, and Y.Backlund. Wireless industrial sensor monitoring based on bluetooth. *Proceedings of the IEEE International Conference on Industrial Informatics*, August 2003.
- [24] J.S.Lee. An experiment on performance study of ieee 802.15.4 wireless networks. *10th IEEE Conference on Emerging Technologies and Factory Automation*, September 2005.
- [25] V. D. Kapsalis, V. S. Koubias, and G. Papadopoulos. Implementation of a mac-layer protocol (git-csma/cd) for industrial lans and its experimental performance. *IEEE Transactions on Industrial Electronics*, December 1997.
- [26] B. Kavitha, G.S.Anil Kumar, G. Maniraman, and Z.Wang. A novel real-time mac protocol exploiting spatial and temporal channel diversity in wireless industrial networks. *International Conference on High Performance Computing*, December 2006.

- [27] K.H.Kim and J.Kim. An energy-efficient fec scheme for weakly hard real-time communications in wireless networks. *Proceedings. of the 12th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications*, August 2006.
- [28] S. Lee, K. C. Lee, M. H. Lee, and F. Harashima. Integration of mobile vehicles for automated material handling using profibus and ieee 802.11 networks. *IEEE Transactions on Industrial Electronics*, June 2002.
- [29] L.L.Bello and O.Mirabella. Communication techniques and architectures for bluetooth networks in industrial scenarios. *10th IEEE Conference on Emerging Technologies and Factory Automation*, September 2005.
- [30] L.Rauchhaupt. System and device architecture of a radio based fieldbus the rfieldbus system. *Proceedings of WFCS*, October 2002.
- [31] D. Miorandi and S. Vitturi. Analysis of master-slave protocols for real-time industrial communications over ieee802.11 wlans. *Proceedings of INDIN04*, June 2004.
- [32] P. Morel, A. Croisier, and J.D. Decotignie. Requirements for wireless extensions of a fip fieldbus. *Proceedings of the IEEE International Conference on Emerging Technologies and Factory Automation*, November 1996.
- [33] G. T. Nguyen, R. H. Katz, B. Noble, and M. Satyanarayanan. A trace-based approach for modeling wireless channel behavior. *Proceedings of the Winter Simulation Conference, Coronado*, December 1996.
- [34] N.Krommenacker and V.Lecuire. Building industrial communication systems based on ieee 802.11g wireless technology. *10th IEEE Conference on Emerging Technologies and Factory Automation*, September 2005.
- [35] Online. R-fieldbus. *Bell Syst. Tech*, September 1960.

- [36] P.Bartolomeu, J.Fonseca, and O.Pacheco. Distributed monitoring subsystems based on a bluetooth implementation. *Proceedings of the IEEE Conference on Emerging Technologies and Factory Automation*, September 2003.
- [37] P.S.Neelakanta and H.Dighe. Robust factory wireless communications: a performance appraisal of the bluetooth/spl trade/ and the zigbee/spl trade/ colocated on an industrial floor. *The 29th Annual Conference of the IEEE Industrial Electronics Society*, November 2003.
- [38] P.Wiberg and U.Bilstrup. Wireless technology in industry-applications and user scenarios. *8th IEEE International Conference on Emerging Technologies and Factory Automation*, October 2001.
- [39] K. Sairam, N. Gunasekaran, and S.R. Redd. Bluetooth in wireless communication. *IEEE Communications*, June 2002.
- [40] S.Lee, K.N.Ha, J.H.Park, K.C.Lee, G.S.Byun, and H.K.Lee. Ndis-based virtual polling algorithm for ieee 802.11b for guaranteeing the real-time requirements. *Computer Standards and Interfaces*, September 2007.
- [41] S.Vitturi and D.Miorandi. Hybrid ethernet/ieee 802.11 networks for real-time industrial communications. *10th IEEE Conference on Emerging Technologies and Factory Automation*, September 2005.
- [42] E. Tovar and F. Vasques. Real-time fieldbus communications using profibus networks. *IEEE Transactions on Industrial Electronics*, December 1999.
- [43] U.Bilstrup and P.Wiberg. Bluetooth in industrial environment. *Proceedings of the IEEE International Workshop on Factory Communication Systems*, June 2000.
- [44] Q. Wang, X. Liu, W. Chen, W. He, and M. Caccamo. Building robust wireless lan for industrial control with dsss-cdma cellphone network paradigm. *Proceedings of the IEEE Real-Time Systems Symposium*, December 2005.

- [45] A. Willig. A mac protocol and a scheduling approach as elements of a lower layers architecture in wireless industrial lans. *2nd IEEE International Workshop on Factory Communication Systems*, June 1997.
- [46] A. Willig. Antenna redundancy for increasing transmission reliability in wireless industrial lans. *Proceedings of the 9th IEEE International Conference on Emerging Technologies and Factory Automation*, April 2003.
- [47] A. Willig. Exploiting redundancy concepts to increase transmission reliability in wireless industrial lans. *IEEE Transactions on industrial electronics*, November 2003.
- [48] A. Willig. Polling-based mac protocols for improving realtime performance in a wireless profibus. *IEEE Transactions on Industrial Electronics*, March 2003.
- [49] A. Willig and A. Kopke. The adaptive-intervals mac protocol for a wireless profibus. *Proceedings of the 2002 IEEE International Symposium on Industrial Electronics*, July 2002.
- [50] A. Willig, M. Kubisch, C. Hoene, and A. Wolisz. Measurements of a wireless link in an industrial environment using an iee 802.11-compliant physical layer. *IEEE Transactions on Industrial Electronics*, December 2002.