Self-Organized TDMA Protocol for Tactical Data Links

Examensarbete utfört i Kommunikationssystem vid Tekniska högskolan i Linköping

av

Wichai Pawgasame
Wuttisak Sa-Ad

LiTH-ISY-EX--11/4527--SE

Linköping 2011
Self-Organized TDMA Protocol for Tactical Data Links

Examensarbete utfört i Kommunikationssystem
vid Tekniska högskolan i Linköping
av

Wichai Pawgasame
Wuttisak Sa-Ad

LiTH-ISY-EX--11/4527--SE

Handledare: Yi Wu
ISY, Linköpings Universitet
Olof Kjellberg
Saab

Examinator: Lasse Alfredsson
ISY, Linköpings Universitet

Linköpings, 14 November, 2011
# Abstract

A Tactical Data Link (TDL) system has been deployed in many military missions as a winning strategy. The performance of a TDL system is governed by the MAC protocol. The MAC protocol that is able to provide more flexibility and high quality of services is more desirable. However, most MAC protocols implemented in current TDL systems are based on a preprogramming TDMA protocol, in which a time slot schedule is fixed. This thesis presents the new self-organized TDMA protocol based on the existing self-organized slot assignment algorithms and the practical military scenarios as the alternative solution to the current preprogramming TDMA protocol. The self-organized TDMA protocol presented in this thesis is based on the Node Activation Polling Access (NAPA), Virtual Slot (VSL), and message based slot assignment algorithms. To evaluate the performance of the designed self-organized TDMA protocol over the preprogramming TDMA protocol, the simulation models for both protocols were implemented and simulated with NS-2 under the specific study scenarios. The results show that the self-organized TDMA protocol offers more flexibility and higher performance than the preprogramming TDMA protocol. In addition, the aspects of stability and security for the self-organized TDMA protocol were discussed. The overall conclusion is that the self-organized TDMA protocol could be a viable alternative for a future TDL system.
Abstract

A Tactical Data Link (TDL) system has been deployed in many military missions as a winning strategy. The performance of a TDL system is governed by the MAC protocol. The MAC protocol that is able to provide more flexibility and high quality of services is more desirable. However, most MAC protocols implemented in current TDL systems are based on a preprogramming TDMA protocol, in which a time slot schedule is fixed. This thesis presents the new self-organized TDMA protocol based on the existing self-organized slot assignment algorithms and the practical military scenarios as the alternative solution to the current preprogramming TDMA protocol. The self-organized TDMA protocol presented in this thesis is based on the Node Activation Polling Access (NAPA), Virtual Slot (VSLot), and message based slot assignment algorithms. To evaluate the performance of the designed self-organized TDMA protocol over the preprogramming TDMA protocol, the simulation models for both protocols were implemented and simulated with NS-2 under the specific study scenarios. The results show that the self-organized TDMA protocol offers more flexibility and higher performance than the preprogramming TDMA protocol. In addition, the aspects of stability and security for the self-organized TDMA protocol were discussed. The overall conclusion is that the self-organized TDMA protocol could be a viable alternative for a future TDL system.
Acknowledgements

We are heartily thankful to our supervisor at Saab, Olof Kjellberg, whose encouragement, guidance and support from the beginning to the final level of the thesis enabled us to develop an understanding of the subject. We are grateful to our supervisors at Linköping University, Lasse Alfredsson and Yi Wu, for their helpful comments and advices on the report.

Lastly, we offer our regards and blessings to all of those who supported us in any respect during the completion of the thesis.

Wichai Pawgasame
Wuttisak Sa-Ad
# Table of Contents

List of Figures ........................................................................................................ vi
List of Tables ........................................................................................................... ix
List of Acronyms ...................................................................................................... x

1. Introduction ........................................................................................................ 1
   1.1 Background .................................................................................................. 1
   1.2 Problem Definition ..................................................................................... 2
   1.3 Scope of the thesis ..................................................................................... 2
   1.4 Working Method ......................................................................................... 3
   1.5 Report Outline ........................................................................................... 3

2. Background Concept ......................................................................................... 4
   2.1 TDL Systems .............................................................................................. 4
   2.2 TDMA Protocol .......................................................................................... 4
   2.3 Slot Assignment Protocols.......................................................................... 5
      2.3.1 Centralized Slot Assignment Protocol .............................................. 5
      2.3.2 Self-Organized Slot Assignment Protocol ....................................... 6

3. Study Scenarios and Parameters ...................................................................... 8
   3.1 Study Scenarios .......................................................................................... 8
      3.1.1 Defensive Scenario ............................................................................ 8
         3.1.1.1 Scenario Phase 1 ........................................................................ 8
         3.1.1.2 Scenario Phase 2 ........................................................................ 9
         3.1.1.3 Scenario Phase 3 ........................................................................ 10
         3.1.1.4 Scenario Phase 4 ........................................................................ 10
         3.1.1.5 Scenario Phase 5 ........................................................................ 12
      3.1.2 Offensive Scenario ............................................................................. 12
         3.1.2.1 Scenario Phase 1 ........................................................................ 12
         3.1.2.2 Scenario Phase 2 ........................................................................ 13
         3.1.2.3 Scenario Phase 3 ........................................................................ 13
         3.1.2.4 Scenario Phase 4 ........................................................................ 15
   3.2 Input Parameters ......................................................................................... 15
      3.2.1 Members ............................................................................................. 15
      3.2.2 TDL Messages .................................................................................... 16
   3.3 Radio Specifications .................................................................................... 16
   3.4 Output Parameters ...................................................................................... 17

4. Description of the Designed Protocol ............................................................... 19
   4.1 Time Slot Assignment Information Exchanges ........................................... 19
   4.2 Frame Structure .......................................................................................... 23
   4.3 Control Slot Assignment Protocol ................................................................ 24
   4.4 Data Slot Assignment Protocol .................................................................... 26
## 4.5 Protocol Functionalities

<table>
<thead>
<tr>
<th>Subsection</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.5.1 Initialization</td>
<td>28</td>
</tr>
<tr>
<td>4.5.2 Net Entry</td>
<td>29</td>
</tr>
<tr>
<td>4.5.3 Net Leaving/Re-entry</td>
<td>30</td>
</tr>
<tr>
<td>4.5.4 Transmission Update</td>
<td>32</td>
</tr>
<tr>
<td>4.5.5 Time Slot Conflict Resolution</td>
<td>34</td>
</tr>
</tbody>
</table>

## 5. Simulation of the Designed Protocol

<table>
<thead>
<tr>
<th>Subsection</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1 TCP/IP Model of the Simulated System</td>
<td>35</td>
</tr>
<tr>
<td>5.2 Implementation of the Simulated System</td>
<td>37</td>
</tr>
<tr>
<td>5.2.1 Application Layer</td>
<td>37</td>
</tr>
<tr>
<td>5.2.2 Transport Layer</td>
<td>39</td>
</tr>
<tr>
<td>5.2.3 Network Layer</td>
<td>40</td>
</tr>
<tr>
<td>5.2.4 Data Link Layer</td>
<td>40</td>
</tr>
<tr>
<td>5.2.5 Physical Layer and Channel Model</td>
<td>44</td>
</tr>
</tbody>
</table>

## 6. Testing and Results

<table>
<thead>
<tr>
<th>Subsection</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.1 Test Programs</td>
<td>46</td>
</tr>
<tr>
<td>6.2 Performance of the Self-Organized TDMA Protocol</td>
<td>46</td>
</tr>
<tr>
<td>6.2.1 Net Entry Time</td>
<td>47</td>
</tr>
<tr>
<td>6.2.2 Net Leaving Reallocation Time</td>
<td>49</td>
</tr>
<tr>
<td>6.2.3 Transmission Update Time</td>
<td>50</td>
</tr>
<tr>
<td>6.2.4 Time Slot Conflict Resolve Time</td>
<td>51</td>
</tr>
<tr>
<td>6.2.5 Maximum Achievable System Throughput, Channel Efficiency, and Maximum Channel Utilization</td>
<td>53</td>
</tr>
<tr>
<td>6.3 Performance of the Self-Organized TDMA Protocol in the Study Scenarios</td>
<td>59</td>
</tr>
<tr>
<td>6.3.1 Packet Delay</td>
<td>61</td>
</tr>
<tr>
<td>6.3.2 Instantaneous System Throughput</td>
<td>64</td>
</tr>
<tr>
<td>6.3.3 Message Update Rate</td>
<td>69</td>
</tr>
<tr>
<td>6.3.4 Time Slot Utilization</td>
<td>72</td>
</tr>
</tbody>
</table>

## 7. Discussions

<table>
<thead>
<tr>
<th>Subsection</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.1 Performance of the Self-Organized TDMA protocol</td>
<td>77</td>
</tr>
<tr>
<td>7.1.1 Net Entry</td>
<td>77</td>
</tr>
<tr>
<td>7.1.2 Net Leaving</td>
<td>78</td>
</tr>
<tr>
<td>7.1.3 Transmission Update</td>
<td>79</td>
</tr>
<tr>
<td>7.1.4 Time Slot Conflict Resolution</td>
<td>80</td>
</tr>
<tr>
<td>7.1.5 Maximum Achievable System Throughput, Channel Efficiency, and Maximum Channel Utilization</td>
<td>80</td>
</tr>
<tr>
<td>7.2 Performance of the Self-Organized TDMA Protocol in the Study Scenarios</td>
<td>81</td>
</tr>
<tr>
<td>7.3 Strengths and Weaknesses of the Self-Organized TDMA Protocol</td>
<td>83</td>
</tr>
</tbody>
</table>
List of Figures

Figure 1-1: A typical tactical data link system..............................................................................1
Figure 1-2: Thesis’s working method..................................................................................................3
Figure 2-1: A TDMA structure............................................................................................................5
Figure 2-2: Centralized slot assignment protocols................................................................................6
Figure 3-1: Initialization phase of three ground stations........................................................................9
Figure 3-2: Hostiles’ information is forwarded to each ground station..............................................9
Figure 3-3: Aircrafts start up inside the existing network.................................................................10
Figure 3-4: Aircrafts move into the overlap network region...........................................................11
Figure 3-5: An aircraft is leaving the network....................................................................................11
Figure 3-6: An aircraft detects a hostile target and need to send target information to other members.........................................................................................................................12
Figure 3-7: Twelve aircrafts start up and take off...............................................................................13
Figure 3-8: Twelve aircrafts move to the meeting point.................................................................14
Figure 3-9: Twelve aircrafts move to the enemy zone......................................................................14
Figure 3-10: Aircrafts detects hostile targets then send target information...................................15
Figure 3-11: A time slot structure......................................................................................................17
Figure 4-1: One-hop and two-hop neighbors..................................................................................20
Figure 4-2: A collision due to a hidden node problem.....................................................................21
Figure 4-3: Neighbor updating algorithm.......................................................................................22
Figure 4-4: The self-organized TDMA frame structure...................................................................23
Figure 4-5: Control slot assignment algorithm................................................................................25
Figure 4-6: Data slot assignment algorithm........................................................................................27
Figure 4-7: Initialization process .......................................................................................................29
Figure 4-8: Net entry process............................................................................................................31
Figure 4-9: Net leaving process.........................................................................................................32
Figure 4-10: Transmission update process......................................................................................33
Figure 5-1: TCP/IP model of the simulated TDL system.................................................................36
Figure 5-2: TDL message format.......................................................................................................38
Figure 5-3: Updating control message format..................................................................................38
Figure 5-4: A datagram structure......................................................................................................39
Figure 5-5: A self-organized TDMA’s MAC frame structure...........................................................42
Figure 5-6: Preprogramming TDMA’s time slot allocation according to the scenarios..................43
Figure 5-7: A preprogramming TDMA’s MAC Frame structure.....................................................43
Figure 6-1a: Average net entry time of all nodes in logarithm scale versus entry interval.............47
Figure 6-1b: Average net entry time of all nodes in the linear scale between the net entry intervals of 10 seconds and 60 seconds.................................................................48
Figure 6-2: Average net entry time versus the number of existing nodes......................................48
Figure 6-3: Average transmission update time versus update interval.........................................50
Figure 6-4: Average transmission update time versus number of existing nodes......................51
Figure A-1: NAMA scheduling algorithm ................................................................. 92
Figure A-2: Operation of NAPA ............................................................................. 94
Figure A-3: NAMA with VSLOT scheduling .............................................................. 95
List of Tables

Table 3-1: TDL radio specification ................................................................. 16
Table 5-1: Message types and the corresponding bits ........................................ 38
Table 5-2: MAC frame types and the corresponding bits ..................................... 42
Table 6-1: Net leaving reallocation time ........................................................ 49
Table 6-2: Time slot conflict resolve time versus number of conflicted time slots in a frame for the first time slot conflict scenario ........................................................................ 52
Table 6-3: Time slots conflict resolve time versus number of conflicted time slots in a frame for the second test scenario ........................................................................ 53
Table 6-4: Nodes’ activities in the first scenario of measuring the maximum achievable system throughput .................................................................................................................. 57
Table 6-5: Nodes’ activities in the second scenario of measuring maximum achievable system throughput .................................................................................................................. 58
Table 6-6: Nodes’ activities in the simulation according to the defensive scenario ...... 60
Table 6-8: Average packet delay in both study scenarios .................................... 64
# List of Acronyms

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSMA/CA</td>
<td>Carrier Sense Multiple Access with Collision Avoidance</td>
</tr>
<tr>
<td>IP</td>
<td>Internet Protocol</td>
</tr>
<tr>
<td>LLC</td>
<td>Logical Link Control</td>
</tr>
<tr>
<td>MAC</td>
<td>Media Access Control</td>
</tr>
<tr>
<td>NAMA</td>
<td>Node Activation Multiple Access</td>
</tr>
<tr>
<td>NAPA</td>
<td>Node Activation Polling Access</td>
</tr>
<tr>
<td>NOAH</td>
<td>No Ad-Hoc Routing Protocol</td>
</tr>
<tr>
<td>RAP</td>
<td>Recognized Air Picture</td>
</tr>
<tr>
<td>TDL</td>
<td>Tactical Data Link</td>
</tr>
<tr>
<td>TDMA</td>
<td>Time Division Multiple Access</td>
</tr>
<tr>
<td>UDP</td>
<td>User Datagram Protocol</td>
</tr>
<tr>
<td>USAP</td>
<td>Unifying Slot Assignment Protocol</td>
</tr>
<tr>
<td>VSLOT</td>
<td>Virtual Slot</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

1.1 Background

Tactical data link (TDL) systems provide means for rapid exchanges of tactical digital information between air, land, sea, and command center units as illustrated in Figure 1-1. A TDL system is the key component in providing situation awareness in a modern warfare and interoperability between different systems [1]. Nowadays, many TDL systems have been implemented and operated in many military missions throughout the world as a winning strategy.

A TDL system is governed by a data link protocol and a physical link technology enabling digital data to be transferred from one source to other destinations through a communication channel. One such channel is a secured radio channel, in which all stations in the network use the same radio channel to distribute and obtain information. To enable the sharing of a channel in a TDL system among members, an efficient medium access protocol (MAC) must be implemented. One approach is to use the time division multiple access (TDMA) protocol, where each station transmits or receives at given time slots. Currently many operating TDL systems are using MAC protocols based on a static preprogramming TDMA protocol. Sometimes a master node is placed into a TDL system to provide some flexibility.
1.2 Problem Definition

Saab is interested in investigating the possibility of replacing the current MAC protocol based on the preprogramming TDMA protocol with a new MAC protocol based on a self-organized TDMA protocol. This thesis will present and investigate a new self-organized TDMA protocol that is based on practical military scenarios and existing self-organized slot assignment algorithms. The following statement describes the problem definition of this thesis.

“Establish a self-organized TDMA protocol that is suitable for the tactical data link and compare its performance to that of the preprogramming TDMA protocol in some practical military scenarios using the relevant performance parameters by performing simulations in the software environment.”

1.3 Scope of This Thesis

This thesis shall present and investigate a new self-organized TDMA protocol based on some practical military scenarios and existing self-organized slot assignment algorithms. To fulfill the purpose of this thesis, the following tasks are needed to be done.

- Any related works, technical papers, and reports shall be studied in order to draw a conclusion whether or not a self-organized TDMA protocol is possible to be implemented and simulated.
- Relevant scenarios shall be defined in order to be used as the basis for designing and simulating a new self-organized TDMA protocol.
- Input parameters to the defined scenarios and the radio specification for the TDL system shall be defined as the basis for designing and simulating a new self-organized TDMA protocol.
- Output parameters of a new self-organized TDMA protocol shall be formulated in order to measure and compare a self-organized TDMA protocol’s performance with a preprogramming TDMA protocol’s performance.
- A new self-organized TDMA protocol based on studied self-organized slot assignment algorithms shall be defined, and a preprogramming TDMA protocol with a time slot schedule based on the relevant scenarios shall also be defined as a comparison.
- The new self-organized TDMA protocol and the preprogramming TDMA protocol shall be implemented and simulated in the simulation tool.
- Performances of both protocols shall be measured and compared in terms of the output parameters defined previously.
- The performances of both protocols shall be discussed, and a conclusion shall be drawn based on the simulation results.
1.4 Working Methods

According to the scope of this thesis, the working method is illustrated as Figure 1-2. The process starts from the idea and ends up with the documentation.

Figure 1-2: Thesis’s working method

1.5 Report Outline

The contents of this thesis report are outlined as follows.

Chapter 2 gives the brief descriptions of a TDL system, a TDMA protocol and slot assignment protocols.

Chapter 3 describes the study scenarios and parameters.

Chapter 4 describes the designed self-organized TDMA protocol.

Chapter 5 describes the implementation details of the simulated protocols.

Chapter 6 presents the results from the simulations.

Chapter 7 discusses the performance of the designed self-organized TDMA protocol in the simulations and other aspects of the designed self-organized TDMA protocol.

Chapter 8 concludes the findings obtained from this thesis and gives some recommendation for further studies and investigations.
Chapter 2

Background Concepts

2.1 TDL Systems

Tactical Data Link (TDL) Systems provide means for exchanges of the tactical digital information between air, land, sea, and command center units. A TDL system contains all layers in the TCP/IP model in order to provide tactical information. However, TDL systems are essentially governed by the protocols in the data link layer and the physical layer.

A data link layer contains two sublayers, a Logical Link Control (LLC) sublayer and a Media Access Control (MAC) sublayer. A LLC sublayer is responsible for functional and procedural means to transfer data between network entities and to detect and possibly correct errors that may occur in a physical layer. Some TDL systems exclude a LLC sublayer from their data link layer specifications, and use an existing LLC sublayer. As for the TDL systems simulated in this thesis, the existing LLC standard is used.

A MAC sublayer provides a channel access control mechanism in a TDL system. There are many MAC protocols applicable to a TDL system. TDMA based protocols are among the most popular MAC protocols for a TDL system. In this thesis, the new MAC protocol based on a TDMA network was defined, in which each member should be able to adjust a slot assignment dynamically when there are any changes in the network.

A physical layer defines the means of transmitting raw bits rather than logical data packets over a physical link connecting network nodes. A physical interface may be a wire, a radio, a satellite, etc. depending on the environment and the type of operation. The TDL system presented in this thesis considers a UHF radio as a physical interface. According to [2], the UHF band is defined from 300 MHz to 3 GHz. The coverage range of the radio depends on the radio’s capability and the environment. In this thesis, only the covering range of a radio is considered. The detailed implementations of the physical layer, e.g. modulation and channel coding schemes, are excluded.

2.2 TDMA Protocol

TDMA (Time Division Multiple Access) is a channel access method for sharing a network medium. TDMA allows several users to share a frequency channel by letting users to transmit signals on different time slots. A TDMA mechanism is illustrated in Figure 2-1. Transmissions will be successful if each member in the same frequency channel transmits data in his own assigned time slots. Otherwise, transmissions will be collapsed.
Figure 2-1: A TDMA frame structure [3]

TDMA is a time-division multiplexing, where there are multiple transmitters connected to one receiver. Each transmitter will transmit on different time slots to avoid collisions with other transmitters. In a mobile network, this becomes difficult because each mobile can move while transmitting. Hence, data may arrive outside the assigned time slots and collisions would occur. Guard periods as illustrated in Figure 2-1 are introduced to prevent this problem. If data arrives within a guard period, a collision would be avoided.

TDMA time slots may be statically assigned or dynamically assigned. When time slots are fixed and cannot be changed or reallocated during a mission, it is called fixed TDMA or static TDMA. Thus, a mission, which is designed to implement a static TDMA, must preplan time slots in advance. A dynamic TDMA is a protocol in which its time slots can be changed or reallocated dynamically, depending on demands from members in the same communication network. A dynamic TDMA may be implemented with a master station to dynamically distribute time slots information. The alternative solution is to implement a self-organized slot assignment protocol.

2.3 Slot Assignment Protocols

There are two types of basic time slot assignment protocol for assigning time slots to each member in the same communication network, i.e. a centralized slot assignment protocol and a self-organized slot assignment protocol. Both protocols are described as follows.

2.3.1 Centralized Slot Assignment Protocol

A centralized slot assignment protocol can be implemented by two methods. The first method requires a preprogramming of a slot assignment on each station. Time slots are fixed for each station during a mission. In such situation, time slots are waste for the member that has nothing
to send on the assigned time slots, while other members that want to transmit high priority packets, do not have enough time slots to transmit. Another problem is that a pre-assigned time slot system cannot adapt its time slot assignment to net entries or leavings of members. Time slots are preprogrammed and fixed, which there is no way to configure a time slot schedule to support a new member during a mission. In addition, when one of members is out of a network range or its radio equipment is turned off, its assigned slots are left unused. Hence, the time slots assigned to the leaving station are left unused and cannot be distributed to other members.

The alternative to a preprogramming centralized slot assignment is to use a master station to dynamically distribute time slot assignment information to all members, when there is a request for a time slot reallocation. However, a master station must always be present in a network during a mission. When a master station loses its communication with other members, the system behaves like a preprogramming centralized slot assignment system.

![Centralized Slot Assignment](image)

**Figure 2-2: Centralized slot assignment protocols**

### 2.3.2 Self-Organized Slot Assignment Protocol

A self-organized slot assignment protocol is more flexible than a centralized slot assignment protocol. In a self-organized slot assignment protocol, each member can dynamically change its time slot assignment according to the messages’ required bandwidth and priority without a presence of a master station. In addition, a self-organized slot assignment protocol should allow each member to adapt its time slot assignment such that a time slot utilization is optimized, when there are any changes in members in a network (entering or leaving).

There have been many researches on a self-organized slot assignment protocol. One research in [4] proposes a self-organized slot assignment protocol called Node Activation Multiple Access (NAMA) protocol. The detailed description of this protocol is given in Appendix A.1. In this protocol, each node is assigned with a random seed value. Random seed values of nodes are exchanged within two-hop communication distance. Hence, all nodes within two-hop communication distance acknowledge random seed values of each other. To determine a transmitting node on a given time slot, each node calculates the hash values of all other nodes within two-hop distance from their random seeds and the slot number. A node with the highest hash value can transmit on a selected time slot.
The NAMA protocol has the problem that time slots cannot be used by other nodes when a winning node has nothing to transmit. The research in [5] is proposed to solve this problem of the NAMA protocol. This improved protocol is called Node Activation Polling Access (NAPA) protocol. The detailed description of this protocol is given in Appendix A.2. The NAPA protocol uses the same mechanism to determine a transmitting node of a selected time slot. If a winning node has nothing to transmit on a selected time slot, it will transmit a polling message instead of remaining silent. A polling message contains a list of polled nodes, which indicates nodes that have a chance to transmit in a selected time slot. The polled nodes are listed according to their hash values. A node that receives a polling message and has something to transmit, checks its position in the polled list. If it is in the polled list and no other nodes that are more than one hop away are in higher positions in the polled list, it will set a back-off time according to its position in the polled list. A polled node will be able to transmit, if it has not detected any transmission before a back-off time is ended. This scheme reduces the chance that time slots are left unused.

Both NAMA and NAPA protocols encounter the problem of a hidden entry node. This problem is discussed in [6]. The problem of a hidden entry node occurs, when an entry node is hidden from other existing nodes that are two hops away. In order for an entry node to be recognized in a network, it must use some time slots to send messages to notify existing members about its existence. These time slots must not be used by any existing nodes in order for an entry node to send a notification message without a collision. One solution is to dedicate some time slots specifically for the net entry processes. However, these dedicated net entry slots will be left unused when there is no new node entering the network. The research in [6] proposes the use of a virtual slot (VSLOT) to handle a net entry without dedicating some time slots as a net entry slots. In this algorithm, there is a virtual node that acts like a node. When this virtual node wins a selected time slot, no real node will transmit. This is a chance for a new node to enter a network. In contrast, if a virtual node does not win the selected time slot, existing nodes in a network can use a selected time slot to transmit. Hence, there is no need to dedicate time slots as net entry slots.

The NAMA and NAPA algorithms do not allocate time slots based on the demand but rather they allocate time slot based on the seed values of the surrounding neighbors. They are suitable to handle contention of a selected time slot, when the knowledge of transmissions on other nodes is not available. Their mechanisms are more suitable for transmissions of the control information, in which transmissions of other nodes are not known in advance. In military missions, the number of required time slots should be dynamically allocated based on a known demand in order to satisfy the quality of service requirement. A data slot assignment algorithm that is based on a known demand is more suitable. The research in [7] implies that dynamically allocating the time slots based on a TDL’s message type will improve the bandwidth utilization. This is because a node with a high priority message should be able to transmit with less delay and a node only reserves time slots as much as it needs.
Chapter 3

Study Scenarios and Parameters

3.1 Study Scenarios

The study scenarios were defined with the advices from Saab to visualize the practical usages of a TDL system in different military airborne missions. The self-organized TDMA protocol is designed such that it can fulfill the requirements of these scenarios. Two study scenarios were defined, i.e. the defensive scenario and the offensive scenario.

The defensive scenario visualizes the air defensive situation, where enemy’s aircrafts are invading into our territory and the responsive action must be done to defense the territory. Aircrafts will be called for taking off and intercepting enemy’s aircrafts. A TDL system will come into handy, in which it can provide enemy information and target information to our aircrafts in the network.

The offensive scenario visualizes the air offensive situation, where our aircrafts are ordered to take off and perform an operation in an enemy’s territory. A TDL system can provide and distribute target information to aircrafts in the network.

The following sections will describe each scenario in different phases of the operations.

3.1.1 Defensive Scenario

The defensive scenario consists of three ground stations and four aircrafts which located in different bases.

3.1.1.1 Scenario Phase 1

In a typical system, there would be several ground stations broadcasting air picture information to members that are within their coverage areas. These ground stations usually situates in different places beyond a radio coverage range of each other to support operation in different areas. In this scenario, there are three ground stations at different locations as illustrated Figure 3-1. Three ground stations will operate in the same network with the same frequency channel or same frequency hopping pattern. Even though they are operated in the same channel, they are beyond the radio coverage range of each other as illustrated in Figure 3-1. Hence, they will not disturb each other. In this phase, each ground station will periodically broadcasting the Recognized Air Picture (RAP) message to provide air picture information to all members within its coverage area.
3.1.1.2 Scenario Phase 2

At this phase, the Early Warning Radar (EWR) connected to three ground stations via the ground network infrastructure detects the hostile aircrafts. The hostile aircraft information is then forwarded to all ground stations as illustrated in Figure 3-2. The hostile aircraft information is broadcasted in the TDL system via the RAP messages by the ground stations. Four aircrafts located in the different bases as illustrated in Figure 3-3 are ordered to takeoff. In this phase, each aircraft attempts to enter the existing TDL network to receive information about the hostile aircrafts.

Figure 3-2: Hostiles’ information is forwarded to each ground station
3.1.1.3 Scenario Phase 3

In this phase, all four aircrafts already took off and are operating in the network. Each aircraft periodically broadcasts the position report messages to other members within its radio range and receives the RAP messages from nearby ground stations. Then, all aircrafts move toward the ground station numbered 1 and travel through the overlap network region where the radio coverage areas of two ground stations are overlapped as illustrated in Figure 3-4. Collisions may be occurred when a self-organized TDMA protocol is used, because the ground station numbered 0 and 1 may not acknowledge each other and they may transmit to aircrafts in the overlap area at the same time. A self-organized TDMA protocol should be able to resolve the collisions that might occur.

3.1.1.4 Scenario Phase 4

In this phase, all four aircrafts are operating inside the same network. One of the aircrafts is leaving the radio coverage area of other members as illustrated in Figure 3-5. All members in the network should acknowledge the leaving member and reallocate time slots such that the time slots reserved by the leaving node can be reused.
Figure 3-4: Aircrafts move into the overlap network region

Figure 3-5: An aircraft is leaving the network
3.1.1.5 Scenario Phase 5

The rest of aircrafts move forward into the battle zone. One of aircrafts’ radars detects hostile targets. It needs to broadcast the target’s track message in order to notify other members about the detected targets, which require more time slots and the new message update rate to fulfill the required quality of service of the track message. Hence, the time slot reallocation is required. All members should acknowledge this reallocation and reallocate its time slot assignment accordingly. The situation is illustrated in Figure 3-6.

![Figure 3-6: An aircraft detects a hostile target and need to send target information to other members](image)

3.1.2 Offensive Scenario

The offensive scenario consists of twelve aircrafts which located in different bases. No ground station is presented.

3.1.2.1 Scenario Phase 1

Twelve aircrafts located in four different bases as illustrated in Figure 3-7 are ordered to take off. In this phase, each aircraft performs the initialization phase and attempts to enter the existing TDL network to share position information with the other aircrafts within the same base. They are considered as operating in the same network, but the distance keeps them apart from communicating with other aircrafts in different bases.
3.1.2.2 Scenario Phase 2

All aircrafts are moving to the meeting point before entering the enemy territory as illustrated in Figure 3-8. In this phase, aircrafts that initially communicate within their group are joined into the larger network. In this phase, each aircraft broadcasts its position to other aircrafts in the larger network.

3.1.2.3 Scenario Phase 3

All twelve aircrafts, which are operating the same area, are moving into the enemy zone as illustrated in Figure 3-9. Each aircraft periodically broadcasts the position report messages to exchange position information with other members.
Figure 3-8: Twelve aircrafts move to the meeting point

Figure 3-9: Twelve aircrafts move to the enemy zone
3.1.2.4 Scenario Phase 4

Aircrafts detect hostile targets and they need to broadcast the target’s track information in order to notify other members about the detected target, which require more time slots and a new message update rate to fulfill the required quality of service of the track message. Hence, a time slot reallocation is required. All members should acknowledge this reallocation and reallocate its time slot assignment accordingly. As aircrafts are going deeper into the enemy’s territory, more targets are detected and more aircrafts are able to detect target. Hence, more time slots are required. In this phase, the TDL system may have some degradation. The situation is illustrated in Figure 3-10.

![Figure 3-10: Aircrafts detects hostile targets then send target information](image)

3.2 Input Parameters

The input parameters define specifications and assumptions of the scenario such that the scenario can be simulated as a practical TDL system in an airborne mission. The input parameters can be divided into the two categories, i.e. members and TDL messages.

3.2.1 Members

The TDL system can support three ground stations operating in different areas. They are separated beyond the radio coverage range, but there exists areas where the radio coverage areas are overlapped.

The maximum number of aircrafts that can operate in the same network is 14 aircrafts. All members communicate in the line-of-sight communication within the given radio range.
3.2.2 TDL Messages

The TDL messages are the information exchanged in the TDL system. The TDL messages are broadcasted to all members in the same network within the given radio range. There are three types of TDL messages in this scenario.

- **Recognized Air Picture (RAP)**: RAP is broadcasted by a ground station. A RAP message can hold 50 tracks. A track contains the information about position, speed, heading, and identification of a vehicle being detected by ground radars. Each track is 100 bytes. The maximum size of a RAP message is 5000 bytes, and it requires the update rate of one message in every 10 seconds. A RAP message has the highest priority, which means that it has the highest priority to reserve time slots.

- **Track**: This message is target track information sent by a member when its radar detects one or more targets. Each target is 100 bytes. One member is capable of sending 10 targets in one track message. The update rate of the track messages is one message in every 2 seconds. The maximum size of a track message is 1000 bytes. A track message has the second priority.

- **Position report**: This message is transmitted regularly at the update rate of one message in every 2 seconds to report the aircraft’s own position to other members. A position report message requires 50 bytes. A position report has the lowest priority.

Each TDL message is transmitted one at a time. Member can switch from one message to other message any time during a mission, but cannot transmit two different types of messages at the same time.

3.3 Radio Specifications

The TDL system requires a radio equipment to be installed on each member in order to transmit and receive the TDL messages. All members are assumed to use the same radio equipment with the same coverage range. The radio equipment supports the simple TDMA protocol, where the time slot assignment can be controlled either by the preprogramming or self-organized methods. The specification of the radio equipment, which is based on the requirement by Saab, is given in Table 3-1.

<table>
<thead>
<tr>
<th>Specifications</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frequency band</td>
<td>UHF (300 MHz -3 GHz)</td>
</tr>
<tr>
<td>Data rate</td>
<td>50 Kbits/s</td>
</tr>
<tr>
<td>Time slot duration</td>
<td>50 milliseconds</td>
</tr>
<tr>
<td>Guard time</td>
<td>2 milliseconds</td>
</tr>
<tr>
<td>Radio Range</td>
<td>300 km (LOS)</td>
</tr>
</tbody>
</table>

Table 3-1: TDL radio specification
According to the radio specification, the time slot duration available for a data transmission is 48 milliseconds as illustrated in Figure 3-7. Hence, the amount of data that can be hold in one time slot is

\[
48 \text{ ms} \times \frac{k\text{bits}}{s} = 2400 \text{ bits} = 300 \text{ bytes}
\]

![Figure 3-11: A time slot structure](image)

### 3.4 Output Parameters

The output parameters are measures of the MAC protocols’ performances. The output parameters are defined as following.

- **Packet Delay**: This is the time from packet arriving at the MAC sublayer to the time it being sent. This parameter can be measured from both self-organized TDMA protocol and preprogramming TDMA protocol.

- **Maximum Achievable System Throughput**: This is the maximum possible amount of information excluding the MAC frame headers sent from all members per unit time. This parameter can be measured from both self-organized TDMA protocol and preprogramming TDMA protocol.

- **Channel Efficiency**: This is the percentage of the maximum achievable system throughput to the channel capacity. This parameter can be measured from both self-organized TDMA protocol and preprogramming TDMA protocol.

- **Maximum Channel Utilization**: This is the percentage of the maximum possible rate of total transmitted bits including the MAC frame headers to the channel capacity as a number of members in the net. This parameter can be measured from both self-organized TDMA protocol and preprogramming TDMA protocol.

- **Instantaneous System Throughput**: This is the rate of total transmitted bits excluding the MAC frame headers at the specific time instance. This parameter can be measured from both self-organized TDMA protocol and preprogramming TDMA protocol.

- **Message Update Rate**: This is the time between two consecutive TDL messages of the same type and the same sender. This parameter can be measured from both self-organized TDMA protocol and preprogramming TDMA protocol. This parameter can also be used as a measure of the quality of service.

- **Time Slot Utilization**: This is the percentage of the time slots actually used for transmitting data to the time slots reserved. This parameter can be measured from both self-organized TDMA protocol and preprogramming TDMA protocol.
- **Net Entry Time:** This is the time that a member spends before getting recognized in the network. This parameter is specifically measured for a self-organized TDMA protocol.

- **Net Leaving Time:** This is the time that existing members in the net use to determine the leaving members and reallocate the time slots. This parameter is specifically measured for a self-organized TDMA protocol.

- **Transmission Update Time:** This is the time that a member uses to reallocate the time slots after changing from one type of message to another type of message. This parameter is specifically measured for a self-organized TDMA protocol.

- **Time Slot Conflict Resolve Time:** This is the time that a member recovers from time slot conflicts, once they occur. This parameter is specifically measured for a self-organized TDMA protocol.
Chapter 4

Description of the Designed Protocol

This chapter describes the designed self-organized TDMA protocol for this thesis. The self-organize TDMA protocol proposed in this thesis was designed based on the NAPA, VSLOT, and message type based data slot assignment protocol described in Section 2.3.2. The designed protocol can be described with the information exchanges required for establishing a self-organized mechanism, the time slot frame structure, the control slot assignment algorithm, the data slot assignment algorithm, and the protocol functionalities.

The necessary information that is required for establishing a self-organized mechanism is included in every packet exchanged in the network. This required information is described in Section 4.1.

The time slot frame structure was particularly designed for the self-organized TDMA protocol presented in this thesis. A time slot frame contains control slots and data slots in order to support both control transmissions and data transmissions. The description of the designed time slot frame structure is given in Section 4.2.

The control slot assignment algorithm was designed based on the combination of the existing self-organized slot assignment algorithms, i.e. the NAPA and VSLOT algorithms. The control assignment algorithm was implemented to handle transmissions of control information that are needed in the adaptation process of the self-organize TDMA protocol. The NAPA algorithm is chosen over the NAMA algorithm because the NAPA algorithm can reduce the chance of unused time slots. The data slot assignment algorithm was designed to handle transmissions of TDL messages. It is based on the message type based data slot assignment protocol. The description of the control slot assignment algorithm and the data slot assignment algorithm are given in Section 4.3 and Section 4.4 respectively.

The protocol functionalities were implemented based on the study scenarios previously defined in Chapter 3. They were designed such that the TDL system with the self-organized TDMA protocol can correctly operate according to the different phases of the study scenarios.

4.1 Time Slot Assignment Information Exchanges

In the self-organized TDMA protocol, there is no predefined time slot assigned for each member and there is no master station to distribute time slot assignment information. However, each member should be able to dynamically reserve time slots using the information exchanged within its own network. In order to reserve time slots, each member should be aware of the existence of its neighbors within one-hop and two-hop distance, and should not reserve same time slots as its one-hop and two-hop neighbors. Neighbors within one hop are surrounding members within a
node’s radio coverage range. Neighbors within two hops are members that are beyond a node’s radio coverage range but are inside the radio coverage range of a node’s one-hop neighbors. Figure 4-1 illustrates a node with the one-hop and two-hop neighbors.

Figure 4-1: One-hop and two-hop neighbors

If a node reserves the same time slot as its one-hop neighbors, a collision will definitely occur when they are transmitting in the same time slot. Even though two-hop neighbors are beyond a node’s radio coverage range, a collision may occur. This is illustrated in Figure 4-2. Node B is a one-hop neighbor of both Node A and Node C, while Node A and Node C are two hops away. If Node A and Node C are not aware of each other, they can reserve the same time slots and collisions will occur at Node B. This problem is known as a hidden node problem. Thus, each node should be aware of its one-hop and two-hop neighbors in order to prevent collisions.

In order to exchange the information about neighbors in the network, each node includes a sender’s ID, a sender’s message type, a sender’s seed value, one-hop neighbors’ IDs, one-hop neighbors’ seed values, and one-hop neighbors’ message types in every transmitted packet. Each node updates the list of its one-hop and two-hop neighbors with the algorithm illustrated in Figure 4-3, when it receives a packet. Each node constructs a neighbor table containing neighbors’ IDs, seed values, message types, and number of hops. When a node receives a packet, it checks for a sender’s ID and adds a sender to its neighbor table as a one-hop neighbor. A receiving node also checks whether a transmission from a sender’s one-hop neighbor has been received in the current time frame. If a sender’s one-hop neighbor has ever transmitted to the
receiving node in the current time frame, it will be added to the receiving node’s neighbor table as a one-hop neighbor. Otherwise, it will be added to the neighbor table as a two-hop neighbor.

Figure 4-2: A collision due to a hidden node problem

The need of including seed values in every transmitted packet is to enable the control slot assignment protocol, in which it will use the seed values of the neighbors within two hops to determine a winner of a selected time slot. Inclusion of message types in every transmitted packet is useful for the data slot assignment protocol. Each message type has the specific number of required time slots and priority in order to satisfy the required update rate. Once message type information of each neighbor within two hops is known, a node can use the data slot assignment algorithm to allocate time slots without any conflicts. In addition, the required quality of service shall be satisfied when allocating time slots according to the message type [7].
Figure 4-3: Neighbor updating algorithm
4.2 Frame Structure

Because there is no predefined time slots assigned for each node, a new node that is going to enter the network should notify and be aware of existing nodes before entering the network. When existing nodes are aware of a new node, it can include a new node as its neighbor and reallocate time slots according to new neighbor’s information. In contrast, a new node can allocate time slots according to the neighbor information of existing network. In order to notify existing nodes about a new entry, a new node needs dedicated time slots to send new entry information. The purpose of these dedicated time slot is different from the time slots reserved for the data transmission. These dedicated time slots will be mentioned as control slots, and the time slots reserved for the data transmission will be mentioned as data slots.

In addition to the net entry information, control slots are also served to transmitting the transmission update information and the acknowledge message. In the study scenarios, a node in the network will update its transmission characteristic when it changes its message type. A node that receives a net entry request or an updating request also needs to send acknowledge message in order to provide a handshake mechanism. Control slots are also used for transmitting these update information and acknowledge information.

In this design, time slots are grouped into frames. Each frame contains 200 time slots, which is equivalent to 10 seconds. A frame is also divides into 10 blocks. Each block contains 20 time slots, which is equivalent to 1 second. Two types of time slots are defined which are the control slots and data slots. Control slots are used to transmit the control information about the net entry and the transmission update, while data slots are used to transmit the TDL messages. Each block contains one control slot and 19 data slots as illustrated in Figure 4-4. Hence, there are 10 control slots and 190 data slots in each time frame.

![Figure 4-4: The self-organized TDMA frame structure](image)
4.3 Control Slot Assignment Algorithm

According to Section 4.2, there are 10 control slots in each time frame. All members in the network will compete for each control slot. The algorithm to select winner of each control slot is based on the Node Activation Polling Access (NAPA) [5] and virtual slot (VSLOT) [6] algorithms. The description of the NAPA and virtual slot algorithms are included in Appendix A.2 and A.3 respectively. The NAPA algorithm is implemented because it can reduce the chance of unused control slots when there are high demands of accessing the control slots to send the control information. The VSLOT algorithm is used because it allows a new node to share control slots with existing nodes for net entry processes.

In this control slot assignment algorithm, each member is assigned with a random seed value, in which it will be changed in every new frame. Virtual nodes are also specified for the network. The seed values of virtual nodes in the network are predetermined and are unchanged. Hence, all members in the network will have the same set of virtual nodes’ seed values. To select a winner of a control slot, each member calculates hashes of its one-hop and two-hop neighbors including hashes of virtual nodes from their seed values. A winner is either a node or a virtual node whose hash is the highest hash. If a winning node has control information to send, it will transmit in the control slot. Otherwise, it will send a polling message that includes the three runner ups of the control slot in the poll list. These three runner ups are nodes whose hashes are the second, the third, and the fourth highest values respectively. A node receiving this polling message and having control information to send checks if it is in the poll list. If so, the node set the back off time with respect to the position in the list. The node waits for the back off time period to end. If it has not detects any transmission during the back off period, it will send its control information. Otherwise, it will remain silent for the current control slot. If a virtual node wins the control slot, all nodes in the network will listen for the net entry information. Hence, it is an opportunity for a new node to enter the network. The net entry algorithm will be described in more details in Section 4.5.1.

The control slot assignment algorithm is illustrated in Figure 4-5. Because a seed value of each node is assigned randomly in every new frame, the seed values of a transmitting node and its one-hop and two-hop neighbors should be included in every packet. A member, who receives a packet, has the most recent seed values of their neighbors within two hops, thus the control slot assignment algorithm can be executed correctly.
Figure 4-5: Control slot assignment algorithm
4.4 Data Slot Assignment Algorithm

TDL messages are transmitted on reserved data slots. Each node reserved data slots for its message transmission based on information about its neighbor within two hops. Each node should employ a unique algorithm based on TDL message types of the node and its neighbors within two hops, such that all nodes within two hops can construct a unique time slot schedule. Hence, collisions in data slots can be avoided. In addition, the required quality of service of each message type will be satisfied [7]. A time slot schedule table is constructed by ordering a node itself and its neighbor within two hops according to the priorities of their messages. If there are multiple nodes whose message priorities are equal, they are ordered by their ID. In this algorithm, the node at the top of the priority list will reserve time slots first, and the next nodes in the list will reserve time slots respectively until time slots are reserved to all nodes in the list. Hence, a high priority node has a higher opportunity to reserve data slots as much as required.

A message type specifies how many time slots are required to obtain the required update rate of a TDL message. To reserve time slots, a node determines the update rate of a current message type. In the study scenarios, there are three message types, which are Recognized Air Picture (RAP), target track, and position report. Their update rates are one message in every 10, 2, and 2 seconds respectively. Then, a node determines the maximum number of bytes required for the current message type. According to Section 3.2.2, RAP requires 5000 bytes, Target track requires 1000 bytes, and Position report requires 50 bytes. The data rate is 50 Kbits/s and one slot can transmit 300 bytes of packet as specified in Section 3.2.3. If each packet has 100 bytes of header, then maximum size of data contained in each packet is 200 bytes. Therefore,

RAP requires $\frac{5000}{200} = 25$ slots in every 10 s.

Target Track requires $\frac{1000}{200} = 5$ slots in every 2 s.

Position Report requires $\frac{50}{200} \approx 1$ slot in every 2 s.

An update rate can be defined in terms of a period of time slot blocks. RAP requires 25 slots in a period of 10 blocks, target track requires 6 slots in a period of 2 blocks, and position report requires 1 slot in a period of 2 blocks. In the simulation, the size of headers will be different, and hence the number of required time slots will be differed from these numbers.
Once the number of time slots required in a period and the number of periods in a frame are known for a current message type, a time slot schedule for a node can be constructed using the algorithm illustrated in Figure 4-6. To guarantee that all nodes within two hops should have at least one data slot to transmit, each node within two hops is allocated with at least one time slot in every 2 seconds and other nodes should not be able to reserve this pre-allocated time slots. To reserve time slots, a frame is divided into periods with the determined number of time slots. Then
a node will find and reserve any free time slots in each period. If there are more free time slots than required time slots in a period, a node will only reserves free time slots as much as required time slots, and then it will move to next period. If there are less free time slots than required time slots in a period, a node will reserve all available free time slots and move to next period. The process continues until all periods in a frame have been searched. Free time slots are defined as time slots that are not reserved by any members within two hops.

In a very congested network, low priority nodes may not be able to reserve enough time slots to obtain the required update rate. However, low priority nodes are guaranteed to have at least one time slot in every 2 seconds. Hence, they are guaranteed to be able to transmit.

4.5 Protocol Functionalities

According to the study scenarios, the functionalities of the self-organized slot assignment protocol can be divided into five functions. These functions are described as follow.

4.5.1 Initialization

The initialization function describes the protocol’s behavior when a node is turning on its radio equipment until its TDL application starts. This is corresponded to Phase 1 in the defensive scenario and the offensive scenario. A node must operate in the certain network. Therefore, the Net ID, in which a node will be operating, should be specified when a TDL radio is turned on. The Net ID specifies the operating frequency and the virtual nodes’ seed values of the network. After a radio is turned on, time slots are synchronized to a master time source and the virtual nodes’ seed values are calculated. A node will be in the idle state, where it neither receives nor transmits any packets. When the TDL application starts, TDL messages will be generated by the application. A TDL message may be divided into several packets depended on the size of message. When the first packet arrives at the MAC sublayer from the upper layer, a node will execute the net entry function to enter the specified net. The process of the initialization function is illustrated in Figure 4-7.
4.5.2 Net Entry

A node will execute the net entry function when it is going to enter the network. This is corresponded to Phase 2 in the defensive scenario and Phase 1 in the offensive scenario. This function is initiated when the MAC sublayer receives the first packet from the upper layer. To enter the desired network, an entry node set the random scanning period to collect the neighbor information in the network. During this scanning period, an entry node listens to any transmissions in the network for the random number of time slots. When an entry node detects some transmissions in the network, it can extract neighbor information included in the packet header and be able to construct the neighbor table by using the algorithm described in Section 4.1. If an entry node cannot detect any transmission during the scanning period, an entry node is considered as the first node in the network. In this situation, an entry node can allocate time slots and start a transmission instantaneously. If an entry node detects some transmissions during the scanning period, an entry node will try to send a net entry message on a control slot that is won by a virtual node.

An entry node uses the control slot assignment algorithm to find the winner of a control slot by excluding itself from the algorithm. This is because existing members in the network are not yet aware of an entry node. If a virtual node wins a control slot, an entry node will send a net entry message on that control slot. Otherwise, a node will wait for another control slot that is won by a virtual node to send a net entry message.
A net entry message will be broadcasted in the network by an entry node. Any existing node in the network that receives this net entry message will send a net entry ACK message in the next available control slot according to the control slot assignment algorithm. A net entry message contains an entry node’s ID, seed, and message type, while a net entry ACK message forwards this information to one-hop neighbors of a net entry ACK sender. The node, who sends a net entry ACK message, and its one-hop neighbors, who receive a net entry ACK message, will update their neighbor tables and reallocate their time slot schedules, in which an entry node are included in their neighbor tables. An entry node waits for a net entry ACK message after transmitting a net entry message, and it will allocate its time slot schedule after receiving a net entry ACK message. In the case that a net entry message has not reached any node in the network, or there are some collisions between net entry requests of multiple entry nodes, an entry node will not receive a net entry ACK message. To prevent an entry node from waiting for a net entry ACK message indefinitely, an entry node set a waiting time for a net entry ACK message, and it will restart the net entry function again when this waiting time is ended.

The Net Entry Function is illustrated in Figure 4-8.

4.5.3 Net Leaving/Re-entry

The net leaving function is executed, when there are one or more members leaving the network. This is corresponded to the defensive scenario and the offensive scenario, when there is any node moving away from the other nodes. The nodes that are still operating in the network will reallocate time slots, such that the time slots reserved by the leaving node can be reused. In order to detect which node is leaving, each node constructs two tables; one table records one-hop neighbors and the other table records two-hop neighbors in each time frame. At the end of each time frame, a node examines each neighbor whether it has been recorded in either one-hop neighbor table or two-hop neighbor table. If there is no record of the neighbor in both tables, that neighbor is considered as leaving the network and it will be removed from the neighbor table. A one-hop neighbor table and a two-hop neighbor table will be reset in every new time frame. In addition, a node will reallocate time slots at the start of every new frame with an updated neighbor table. Thus, data slots reserved by leaving nodes will be reused by other nodes in a new time frame. The net leaving function is illustrated in Figure 4-9.

The net re-entry function is executed, when a node leaves and then re-enters the network again. There is no specific function for the net re-entry process. If a node leaves and re-enters the network before its neighbor table and existing nodes’ neighbor tables are updated, there will be no problem in re-entering the network. A leaving node can use time slots, in which it had used before leaving the network. However, if either a leaving node or existing nodes in the network update their neighbor tables and reallocate time slots after a leaving node is absent, there may be collisions in data slots. This is because a leaving node and existing nodes in the network reallocate time slots with different neighbor information. These collisions should be resolved by
the function described in Section 4.5.5, and a re-entering node should be able to operate in the network again without collisions.

Figure 4-8: Net entry process
4.5.4 Transmission Update

The transmission update function is performed, when a node wants to change its current message type. This is corresponded to the defensive scenario and the offensive scenario, when an aircraft changes its message type during the mission. The number of required time slots and the messages’ update rate will be changed, and hence the time slot schedule will need to be reallocated. The neighbors of an updating node also need to be notified about the change, such that they can reallocate the time slot schedule accordingly. Changing a message type is initiated by the command from a user.
When the application layer of the TDL system receives a message updating command, it will update the current message type, size and message generating rate. An updating node will also generate an interrupt message to notify the MAC sublayer about the change. Once the MAC sublayer of an updating node receives this message, the updating node will generate a transmission update request message to notify its neighbors. A transmission update request message is a control message that notifies one-hop neighbors about the new message type and
size. An updating node will access a control slot according to the control slot assignment algorithm in order to send a transmission update request message. After an updating node transmitted a transmission update request message, the updating node will waits for a transmission update ACK message for some time slots. A transmission update ACK message is transmitted in a control slot by a neighbor, who receives the transmission update message. This neighbor will reallocate time slot according to the transmission update request information after sending the transmission update ACK message. The updating node will reallocate time slots after receiving the transmission update ACK message. Other neighbors will also reallocate time slots accordingly after receiving the transmission update ACK message. If a waiting time is expired before an updating node receives a transmission update ACK message, the updating node will restart the transmission update process. Note that an updating node will transmit TDL messages with an old time slot allocation during transmission update process, but it will transmit TDL messages with a new time slot allocation after receiving a transmission update ACK message. The transmission update function is illustrated in Figure 4-10.

4.5.5 Time Slot Conflict Resolution

A time slot conflict occurs when multiple nodes reserve the same data slot for transmissions. Hence, collisions among transmitted packets will occur, and no packet transmitted on the same slot in the same network will be received. In the study scenarios and this self-organized slot assignment protocol, a time slot conflict may occur in two situations. The first situation is when a node leaves the network, reallocates time slots outside the network, and re-enters the network again. Some data slots may be allocated to multiple nodes, because the leaving node reallocates time slots with different neighbor information from the existing nodes in the network. Hence, collisions may occur in some data slots.

The second situation is when a node move into a hidden node area as described in Section 4.1. In the second situation, two nodes are beyond the radio coverage range of each other, and there exists a hidden node area between these two nodes. These two nodes can reserve the same time slots without causing a collision. If there is the third node moves into this hidden node area, the third node will receive from both nodes in the same data slots and collisions will occur.

A time slot conflict can be resolved automatically by the neighbor update algorithm. Because sender information and neighbor information are included in every packet, nodes in conflict can update their neighbor information if they can receive any data packet or control packet containing information about conflicted nodes in conflict-free slots. As long as not all time slots are in conflicts and there is information about conflicted nodes transmitted in conflict-free the slots, a node can use conflict-free time slots to update its neighbor information. Hence, conflicts can be resolved when a node reallocate time slot with updated neighbor information.
Chapter 5

Simulation of the Designed Protocol

According to [8], Network Simulator 2 (NS-2) can be used to simulate a TDL system. NS-2 is an open source network simulation tool. It includes all layers in a network system according to the TCP/IP model. The main focus of this thesis is the MAC sublayer. However, the MAC sublayer cannot be simulated independently. A TDL system must be broken down into the defined layers according to the TCP/IP model, and each layer must be implemented and simulated in NS-2. With some modifications in each layer of NS-2, a TDL system can be simulated. This chapter gives the implementation details of the simulated TDL system according to the TCP/IP model.

5.1 TCP/IP Model of the Simulated TDL System

The TCP/IP Model of the simulated TDL system is illustrated in Figure 5-1. Each layer has specific purpose in the simulated TDL system and will be described in this section.

The application layer of the simulated TDL system generates tactical information messages to be broadcasted to all members within the radio’s coverage range. Each message has its own specific priority and required update rate. A higher priority message should have higher priority than a lower priority message to reserve time slots for transmission. All messages considered in this thesis are periodic messages that have to be sent periodically. Hence, the required update rates of the TDL messages should be satisfied in order to guarantee the quality of service of the tactical application. In addition to tactical information, the application layer also generates the control messages used in controlling the MAC sublayer.

The transport layer of the simulated TDL system provides an end-to-end service for the tactical data link application. The transport protocol that was implemented and simulated in this TDL system was the User Datagram Protocol (UDP). The reason is that all messages are periodic messages that are sent continually to provide the same information. There is no need to wait for retransmissions of dropped messages that would interrupt message stream. Waiting for a retransmitted message is undesirable because the tactical information is needed to be updated at the desired rate. The UDP transport protocol also partitions a message into small datagrams, and hence each datagram will be processed independently by the lower layers of the TDL system. The datagrams from the same message are tagged with the same sequence number, such that a message can be resembled correctly at the receiver end’s transport layer.

The network layer of the simulated TDL system provides a routing service to deliver information to the desired destination. The network layer processes each datagram from the transport layer, and produces an IP packet. The Internet protocol (IP) was used to implement the network layer in this model due to the fact that most modern tactical radio are IP-based radios, and the IP
protocol is already implemented in NS-2 [9]. Hence, it is more practical and simple to implement and simulate the system based on an IP network.

The Data Link Layer of the simulated TDL system contains two sublayers; the Logical Link Control (LLC) sublayer and the Media Access Control (MAC) sublayer. The LLC sublayer is responsible for functional and procedural means to transfer data between nodes and control error.
in packets. In this model, the LLC sublayer based on NS-2 [9] was used to control packet flows and detect errors.

The MAC sublayer provides a physical addressing and a channel access control mechanism. In this model, either the self-organized TDMA protocol as described in Chapter 4 or the preprogramming TDMA protocol was implemented in the MAC sublayer as a channel access control mechanism. Basically, the MAC sublayer processes a packet obtain from the LLC sublayer and produce a frame that includes the specific MAC header into a packet. This MAC header together with the channel access control algorithm provides the physical addressing and channel access control mechanisms for this TDL System.

The physical layer of the simulated TDL system defines a mean of transmitting raw bits rather than logical data packets over a channel. The physical interface in this tactical data link system was implemented as an UHF radio that provides the transmission range of 300 km according to the specification defined in Section 3.3, and the channel was modeled as a wireless channel model. Other aspects of the physical layer and the channel such as modulation, channel coding, channel noise model, and fading were not considered in this model.

5.2 Implementation of the Simulated TDL System

The simulated TDL System was implemented in NS-2. NS-2 mainly contains two parts, i.e. the C++ modules and the OTcl modules as described in [9]. The C++ modules are implemented to provide network model functionalities; while the OTcl modules provide control to C++ modules such that a network model can be simulated and tested. In this section, the C++ modules that implemented the functionalities of the TDL System will be described, whereas the test programs based on the OTcl modules will be described in Chapter 6. Appendix B.1 lists NS-2’s C++ classes that were implemented and modified for the simulated TDL system.

5.2.1 Application Layer

The TDL application module was implemented to provide TDL messages to the TDL system. There are three types of TDL messages as specified in Section 3.2.2 that will be broadcasted among members in the network. In addition, there is one MAC control message, i.e. the TDL message updating message as specified in Section 4.5.4. This MAC control message will not be transmitted, but will be used to control the MAC sublayer within a node. The TDL application module also contains functions that process incoming TDL messages from other nodes. When a TDL message is received by the TDL application module at the receiver end, the arrival time is recorded in the trace file for being analyzed. In the real TDL system, the content of TDL messages will be further processed depended on the specific applications. However, those processes are ignored in this simulation.

The TDL application was implemented as a NS-2’s C++ class that generates TDL messages with the defined type and size. In a test program, the TDL Application object is created and attached
to each node in the simulation. When the TDL Application attached to a node is started by a test program, it will periodically create TDL messages with the defined size, and process TDL messages received from other nodes. The rate, in which messages are generated, is defined by the message type as specified in Section 3.2.2. The message type and size can be defined in a test program, and can be changed anytime during a simulation.

Each TDL message is formatted as illustrated by Figure 5-2. A TDL message contains a payload of defined size, a message type field, and a payload size field. A payload contains information of a TDL message, but the content of a payload was ignored in this simulation. A message type field is 4 bits that indicates the type of a generated message. Table 5-1 presents message types and their corresponding bits. In the real system, using only 2 bits will be enough to represent all four types of messages. However, the C++ programming language used in the simulation does not allow binary value assignment. The message types are defined by 4 bits such that their values can be assigned by hexadecimal values, in which are allowed in the C++ programming language. A payload size field is a 4-byte integer that indicates a payload’s size. Hence, the application header size is about 5 bytes.

<table>
<thead>
<tr>
<th>Message Types</th>
<th>Bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>RAP</td>
<td>0x1</td>
</tr>
<tr>
<td>Radar Track</td>
<td>0x2</td>
</tr>
<tr>
<td>Position Report</td>
<td>0x3</td>
</tr>
<tr>
<td>Message Updating</td>
<td>0x4</td>
</tr>
</tbody>
</table>

Table 5-1: Message types and the corresponding bits

A TDL message updating message is a MAC control message that interrupt the MAC sublayer, when an updating command is received from a test program. This MAC control message is formatted as illustrated in Figure 5-3. A TDL message updating message contains a message type field, a new TDL message type field, and a new payload size field. A message type field is 4 bits that indicates the type of a generated message. A new payload size field is a 4-byte integer indicating a size of each new TDL message’s payload.
5.2.2 Transport Layer

NS-2 already provides the C++ class that implements the UDP protocol. However, it cannot fit into the simulated TDL system in this thesis. The provided UDP protocol does not support porting to the implemented TDL application, and it does not have a segmenting function, which is useful for this simulated TDL system. Therefore, the existing UDP class was modified to provide the required functionalities.

In a test program, the modified UDP transport agent can be defined and attached to the TDL application of each node. After the application generates a message, it sends down a message to the modified UDP agent. The modified UDP agent segments a message into several datagrams, if the message size is larger than the maximum allowed data size of a datagram. If the message size is smaller than the maximum allowed data size of a datagram, a message can fit into one datagram. The modified UDP agent also assigns a sequence number to each TDL message. The datagrams of the same message are tagged with the same application’s ID, sequence number, and total message size.

Figure 5-4 illustrates a datagram structure. A datagram header contains a 1-byte integer of an application’s ID field, a 4-byte integer of a sequence number, a 4-byte integer of a data size field, and a 4-byte integer of a total message size field. Thus, the total UDP header length is 13 bytes. A data size field indicates amount of data from a message contained in a datagram, whereas a total message size field indicates total amount of data of a message. In this simulation, the maximum data size of a datagram is set to 216 bytes for the self-organized TDMA protocol and 258 bytes for the preprogramming TDMA protocol, such that a datagram can be processed by lower layer and fitted into one time slot. There is no need to specify port number because this modified UDP agent was implemented specifically for the TDL application.

When the modified UDP agent at the receiver end receives a datagram from the lower layer, it examines the application’s ID and sequence number contained in the header. If there are previous received datagrams with the same application’s ID and sequence number as the current received datagram, the data part of the current datagram is reassembled with the data parts of previous datagrams. Then, the modified UDP agent checks whether the total received data size is equal to the total message size. If so, reassembled data are sent to the application. Otherwise, it will keep reassemble latter datagrams from the same message until the total received data size is equal to the total message size.
5.2.3 Network Layer

In this simulated TDL system, the IP network protocol class provided by NS-2 was used without any modification. This IP class adds an IP header to each datagram to produce an IP packet, and forwards it to the lower layer. NS-2’s IP header is 20 bytes, and its structure is described in [9]. When the IP class receives an IP packet from the lower layer, it examines the destination IP address in the IP header whether the received IP packet should be sent to the upper layer or relayed to another node. In this system, all packets are one-hop broadcasted. Hence, no packets will be relayed. To achieve this, the No Ad-Hoc (NOAH) routing protocol as proposed in [10] was used. In this routing protocol, all packets will be broadcasted within one-hop neighbors once, and they will not be forwarded to other nodes. The IP and routing modules can be defined for each node in a test program.

5.2.4 Data Link Layer

The Data Link Layer contains two sublayers. Each sublayer is implemented separately as follows.

LLC Sublayer

The LL class provided by NS-2 was used without any modification to provide the LLC sublayer functionalities for this simulated TDL system. According to [9], this class provides the control of a packet queue to the MAC sublayer. It also detects errors in a received packet. When an error occurs in a packet, the LL class will drop a packet. NS-2 provides many packet queuing models as specified in [9], but the drop-tail queuing model was selected for this simulated TDL system. According to [9], the drop-tail queuing model drops an incoming packet from the upper layer, if the packet queue is already full. Otherwise, an incoming packet is queued according to the arrival order. The earliest packet in the queue will be read first. The MAC sublayer decides when to read the next packet from the queue. Normally it will read the next packet, when the MAC sublayer is in the idle state, where it is not transmitting any packet. The LL class also examines a received packet whether it contains error. If so, it will drop a packet. The LLC protocol and the queuing model for each node in a simulation can be defined in a test program.

MAC Sublayer

Two MAC protocols were implemented for this simulated TDL system. The first MAC protocol is the designed self-organized TDMA protocol, which is the main protocol for this thesis. The second protocol is the preprogramming TDMA protocol, which were implemented as a comparison to the main protocol. Both protocols were implemented based on the MAC/TDMA class provided by NS-2.

Both protocols are based on TDMA. It is very important that time slots of all nodes in a simulation are synchronized. In NS-2, a time slot cycle is started when a node is created in a test
program. In order to obtain synchronization, all nodes must be created at the same time. However, they can start operating in the network at different time. In the real system, each node can synchronize time slots to a master time source such as a GPS time source.

The MAC sublayer contains two main mechanisms. The first mechanism is a physical addressing, which addresses each physical node with a unique address. In this simulated TDL system, the physical addressing mechanism defined in [9] was used. In this mechanism, three parameters must be defined and included in a MAC frame header. These are a source’s physical address, a destination’s physical address, and the type of a physical address. According to [9], the physical address type is the Ethernet MAC address. NS-2 uses 4 bytes to specify each MAC address as mentioned in [9]. A source’s MAC address is assigned to each node, when a node is created. To broadcast a frame created by the MAC sublayer, a destination’s MAC address must be set to the MAC broadcast address. Hence, a frame will be received and processed by all neighbors within one hop.

The second mechanism is a channel access protocol. There were two channel access protocols implemented in this thesis, i.e. the self-organized TDMA protocol and the preprogramming TDMA protocol. Each packet from the upper layer is processed, and the MAC frame header is added to each packet to produce a frame. The MAC frame format for the self-organized TDMA protocol is illustrated in Figure 5-5. Each MAC frame contains a 1-byte value of a physical address type, a 4-byte value of a source’s physical address, a 4-byte value of a destination’s physical address, a 4-bit value of a MAC frame type according to Table 5-2, a 1-byte integer of a node’s ID field, a 4-bit value of a node’s message type, a 1-byte integer of a node’s seed, 15x1-byte integers of 15 neighbors’ IDs, 15x4-bit values of 15 neighbors’ message type, 15x1-byte integers of 15 neighbors’ seeds, a 1-byte integer of the number of included one hop neighbors, and a payload. Hence, the total MAC frame header length is about 51 bytes. According to the UDP header’s length and the IP header’s length described in previous sections, the total header length is 84 bytes. If each datagram contains 216 bytes of an application message, the total MAC frame size is 300 bytes which should exactly fit into one time slot according to the radio specification in Section 3.3. In this simulation, the application layer adds 5 bytes of the application header to each TDL message. Hence, the numbers of required time slots for each message type are as follows.

RAP requires \( \frac{5005}{216} \approx 24 \text{ slots in every 10 s.} \)

Track requires \( \frac{1005}{216} \approx 5 \text{ slots in every 2 s.} \)

Position Report requires \( \frac{55}{216} \approx 1 \text{ slot in every 2 s.} \)

The MAC frame header was designed to meet the specification of the study scenarios, which must be able to support 16 members. The neighbor information field in the header supports all 15
neighbors, but only one-hop neighbors will be included in the header. A payload of the MAC frame is an IP packet from the upper layer.

<table>
<thead>
<tr>
<th>Physical Address Type (1 byte)</th>
<th>Source’s physical address (4 bytes)</th>
<th>Destination’s physical address (4 bytes)</th>
<th>Frame Type (4 bits)</th>
<th>Node’s ID (1 byte)</th>
<th>Node’s Message Type (4 bits)</th>
<th>Node’s Seed (1 byte)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Neighbors’ IDs (15 x 1 bytes)</td>
<td>Neighbors’ Message Type (15 x 4 bits)</td>
<td>Neighbors’ Seed (15 x 1 bytes)</td>
<td>Number of one hop neighbors included (1 byte)</td>
<td>Payload</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 5-5: A Self-Organized TDMA’s MAC frame structure

<table>
<thead>
<tr>
<th>MAC Frame Types</th>
<th>Bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>TDL Message</td>
<td>0x1</td>
</tr>
<tr>
<td>Net Entry</td>
<td>0x2</td>
</tr>
<tr>
<td>Net Entry ACK</td>
<td>0x3</td>
</tr>
<tr>
<td>Transmission Update Request</td>
<td>0x4</td>
</tr>
<tr>
<td>Transmission Update ACK</td>
<td>0x5</td>
</tr>
</tbody>
</table>

Table 5-2: MAC frame types and the corresponding bits

The preprogramming TDMA protocol assigns time slots according to the configuration file, which specifies time slots reserved for each node. Figure 5-6 illustrates the time slots pre-assigned to each node according to the study scenarios. In the study scenarios, there can be only two ground stations within a two-hop distance. The third ground station that is beyond a two-hop distance can use the same time slots as the farthest ground station. There can be 14 aircrafts operated within a two-hop distance. It is very important that all nodes in the same network must use the same configuration file. The example of this configuration file is given in Appendix B.1.5. The preprogramming TDMA protocol does not need to include the neighbor information into the MAC frame header. Therefore, it only includes the physical addressing information in the MAC frame header. The preprogramming TDMA’s MAC frame format is illustrated in Figure 5-7. The total MAC frame header length is 9 bytes. Hence, the total header length is 42 bytes. If each datagram contains 258 bytes of an application message (253 bytes of payload data), the total frame size is exactly 300 bytes. For the preprogramming TDMA protocol, the numbers of required time slots for each message type are as follows.

\[ \text{RAP requires } \frac{5005}{258} \approx 20 \text{ slots in every 10 s.} \]

\[ \text{Track requires } \frac{1005}{258} \approx 4 \text{ slots in every 2 s.} \]

\[ \text{Position report requires } \frac{55}{258} \approx 1 \text{ slot in every 2 s.} \]
A ground station will only transmit the RAP messages, thus a ground station will require 20 slots in every 10 seconds (one time frame). The time slot allocation for each ground station is illustrated in Figure 5-6. The reserved time slots for a ground station are separately allocated such that four time slots in each block are assigned to each ground station. Each aircraft will transmit either track messages or position report messages, and hence it requires maximum of 4 slots in every 2 seconds. The time slots in the first, third, and fifth blocks of a time frame are reserved for Aircraft 1 to Aircraft 8. The time slots in the second and fourth blocks of a time frame are reserved for Aircraft 9 to Aircraft 14. Each aircraft will be assigned with four time slots in each block. According to this structure, the update rate for track and position report messages sent by each aircraft will not always be met with the required update rate. If a TDL message uses all reserved four slots in each block, Aircraft 1 to Aircraft 8 will have the update rate of either 2 seconds or 4 seconds and Aircraft 9 to Aircraft 14 will have the update rate of 4 seconds.

Both self-organized and preprogramming TDMA protocols were implemented based on the TDMA mechanism provided by the MAC/Tdma class. Hence, the time slot structures are the same for both protocols. The differences between two protocols are the time slot assignment algorithm and the MAC frame format. The time slot structure was implemented based on radio specification given in Section 3.2.3. According to the MAC/Tdma class, the built-in NS-2’s timer is used to control the duration of each time slot. In the self-organized TDMA protocol, each time slot is either tagged as a control slot or a data slot according to the TDMA frame structure in Section 4.2. Each node access time slots according to the slot assignment algorithm described in Chapter 4. In the case of the preprogramming TDMA protocol, all time slots have the same purpose, which are only used for data transmissions. Each node access time slots according the time slot configuration file.

![Figure 5-6: Preprogramming TDMA’s time slot allocation according to the scenarios](image)

![Figure 5-7: A preprogramming TDMA’s MAC Frame structure](image)
5.2.5 Physical Layer and Channel Model

There is no real physical interface and channel in the simulation. NS-2 provides virtual wireless physical interfaces and channel models that can be used in the simulated TDL system as described in [9]. The NS-2’s virtual wireless physical interface was configured to meet the specification in Section 3.2.3. These configuration parameters were a data rate, a channel frequency, and a transmitting power. The data rate was 50 Kbits/s according to the specification. The channel frequency is calculated according to the Net ID with the following equation.

\[ \text{Channel Frequency} = 300 \text{ MHz} + \text{Net ID} \times 25 \text{ kHz} \]  

(5.1)

Hence, all frequencies are in the UHF range and are spaced by 25 kHz. Section 3.3 indicates the required transmission range of a radio. However, the required transmission range cannot be explicitly set on a radio. The transmitting power of a radio can be set such that it give the required transmission range under the appropriate propagation model. In order to calculate required transmitting power, the propagation model must be defined. In this simulated system, the NS-2’s either two-ray ground propagation model or free space propagation model was used. According to [11], the two-ray ground model defines the received power by the following equation.

\[ P_r = \frac{P_t G_t G_r h_t^2 h_r^2}{d^4 L} \]  

(5.2)

\( P_r \) and \( P_t \) are a receiving power and a transmitting power in Watts respectively. \( G_t \) and \( G_r \) are antenna gains of a transmitter and a receiver respectively. \( h_t \) and \( h_r \) are the height of a transmitting antenna and a receiving antenna respectively. \( d \) is a distance between a transmitting node and a receiving node. \( L \) is a system loss. According to [9], this model is suitable for a long distance communication. This model is valid, if a distance between a transmitter and a receiver are over the cross over distance. According to [9], the cross over distance can be determined from the following equation.

\[ \text{Cross Over Distance} = \frac{4\pi h_t h_r}{\lambda} \]  

(5.3)

\( \lambda \) is a wavelength of an operating frequency in meters. In this simulation, all nodes are assumed to situate at the same height, and the antennas’ heights are the same for all nodes. \( h_t \) and \( h_r \) are assumed to be 1000 meters. Then, the cross over distance can be calculated for each operating frequency. If the required distance is below the cross over distance, the free space model should be applied instead of the two-ray ground model. According to [9], the free space model defines the received power by the following equation.

\[ P_r = \frac{P_t G_t G_r \lambda^2}{(4\pi)^2 d^2 L} \]  

(5.4)
Once the cross over distance for an operating frequency is known, it was compared with the required distance \((d)\) to determine the propagation equation that should be used. According to the radio specification, the required distance is 300 km. \(G_t, G_r,\) and \(L\) are normally set to 1 as recommended by [9]. \(P_r\) should be set to the received power threshold of the wireless physical interface. This value can be configured. In this simulation, the received power threshold was set to \(10^{-13}\) W. When the appropriated propagation equation and all parameters are known, the required transmitting power \((P_t)\) for the given transmission range can be calculated for each operating frequency.

The NS-2’s wireless physical interface class, wireless channel class and ground propagation class were used without modification to implement the physical interface and the channel of this simulated system. The configuration of these modules and the calculation of the transmitting power can be done in a test program. The NS-2’s wireless interface and channel classes only calculate whether transmitted packets would be detected by a receiver with the given transmitted power, propagation model, antennas, receiver sensitivity, and distance. It neglects all other aspects of the physical layer and the channel such as modulation schemes, channel coding methods, channel noises, and effects of fading.
Chapter 6

Testing and Results

6.1 Test programs

The test programs for the simulated TDL system were implemented in the TCL language. The main purpose of these test programs are to test the functionalities of the simulated TDL system implemented in C++ classes, in which are described in Chapter 5. The C++ classes and a test program are linked by the OTcl linkage class according to [9], but this linkage class will not be described in this report.

The main functions of a test program are to create mobile nodes with the simulated TDL system, movement profiles, and transmission profiles. When a node is created, the C++ objects containing the functionalities of each layer in the simulated TDL system are created and connected. A node is ready to transmit or receive data. A test program can control when a node should start or stop transmitting TDL messages. In addition, it controls when a node should change its message type.

Each node contains an X-Y coordinate specifying its position in the defined area. The moving speed can also be defined for each node. According to [9], the height of each node cannot be defined in NS-2, and all nodes are assumed to be at the same height. However, nodes can be defined with different antenna heights. In this simulation, all nodes were assumed to use the same antenna’s height. A test program can control when to move nodes to the specific position with the defined speed. Hence, the test scenarios can be setup such that it is similar to the study scenarios.

In addition, a test program is used to define the important parameters for the simulated TDL system. These parameters are protocols for each layer, message types, message sizes, physical interfaces, data rates, antennas, channels, transmitting powers, and receiver sensitivities.

After a test program is executed, the trace files will be generated. These trace files contain useful data that can be processed to obtain the output parameters as defined in Section 3.4 for the evaluation of the MAC protocols.

6.2 Performances of the Self-Organized TDMA protocol

The adaptability performance of the self-organized TDMA protocol can be measured in terms of the net entry time, the net leaving reallocation time, the transmission update time, and the time slots conflict resolve time as defined in Section 3.4. In addition, the network performance of the self-organized TDMA protocol can be measured and compared with the preprogramming TDMA protocol in terms of the maximum achievable throughput, the channel efficiency, and the
maximum channel utilization as defined in Section 3.4. These output parameters were measured and presented as follows.

6.2.1 Net Entry Time

The net entry time was measured from the time that the MAC sublayer received the first TDL packet to the time that the MAC sublayer was able to allocate time slots for the first time. The test scenario was set up such that the first node was started at the first second, and the rests were started in every next 3, 5, 10, 15, 20, 25, 30, 40, 50, and 60 seconds respectively. The total number of nodes was 16 nodes, and they were placed within one hop of each other. In addition, every node only broadcasted the position report messages. This set up was to investigate how the entry interval and the number of existing nodes in the network would affect the net entry time. The test program for measuring net entry time is included in Appendix B.2.1.

Figure 6-1a illustrates the average net entry time in the logarithm scale of all 16 nodes for different entry intervals. Figure 6-1a illustrate that the average net entry time was converged to some value as the entry interval increases. According to Figure 6-1b, the average net entry time during longer net entry intervals was approximately converged to 3.35 seconds. The very high average net entry time at the shortest entry interval was caused by several retransmissions of net entry requests.

Figure 6-1a: Average net entry time of all nodes in logarithm scale versus entry interval
Figure 6-1b: Average net entry time of all nodes in the linear scale between the net entry intervals of 10 seconds and 60 seconds

Figure 6-2: Average net entry time versus the number of existing nodes
The numbers of existing nodes, when a node was trying to enter the network, was also recorded. The average net entry times of all nodes entering the network with the same number of existing nodes were calculated. Note that the average net entry times were computed from the data sets with entry intervals of 20, 25, 40, 50, and 60 seconds. These data sets were chosen such that the net entry time was not affected by the entry interval, and the effect of the number of existing nodes on the net entry time can be clearly investigated. Figure 6-2 illustrates the average net entry time versus the number of existing nodes in the network. The result implies that the average net entry time was higher, when there were some existing nodes in the net. However, the average net entry time was converged as the number of existing nodes increases. According to Figure 6-2, the average net entry time was approximately converged to 2.75 seconds.

6.2.2 Net Leaving Reallocation Time

The net leaving reallocation time was measured from the time that a leaving node disappeared from the network to the time that a leaving node was deleted from the neighbor lists of existing nodes in the network. The test scenario was set up with 8 nodes. Initially, they were within one hop of each other. One node was assumed to be a ground station and did not move. Then, a mobile node moved to the distance beyond one hop of other mobile nodes. The leaving time when a mobile node moved outside coverage areas of other nodes and the deleted time when a leaving mobile node was deleted from the neighbor lists of other nodes in the network were recorded. The program for this test scenario is included in Appendix B.2.2. The net leaving reallocation times were presented in Table 6-1.

<table>
<thead>
<tr>
<th>Node Number</th>
<th>Leaving time (s)</th>
<th>Deleted time (s)</th>
<th>Net Leaving time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1323</td>
<td>1350</td>
<td>27</td>
</tr>
<tr>
<td>3</td>
<td>1144</td>
<td>1170</td>
<td>26</td>
</tr>
<tr>
<td>4</td>
<td>965</td>
<td>990</td>
<td>25</td>
</tr>
<tr>
<td>5</td>
<td>786</td>
<td>810</td>
<td>24</td>
</tr>
<tr>
<td>6</td>
<td>607</td>
<td>630</td>
<td>23</td>
</tr>
<tr>
<td>7</td>
<td>428</td>
<td>450</td>
<td>22</td>
</tr>
<tr>
<td>8</td>
<td>249</td>
<td>270</td>
<td>21</td>
</tr>
</tbody>
</table>

Table 6-1: Net leaving reallocation time

The result implies that the net leaving reallocation time can be calculated as

$$T_L = \Delta t + 2 * t_f$$  \hspace{1cm} (6.1)

$T_L$, $\Delta t$, $t_f$ are the net leaving allocation time, the time left in a frame when a node is leaving, and the time frame duration respectively. If a node reappears during this net leaving allocation time interval, it will not be removed from the network.
6.2.3 Transmission Update Time

The transmission update time was measured from the time that the MAC sublayer received a message update request from the upper layers to the time that a node was able to reallocate time slots according to a request. The test scenario was set up with 8 nodes, in which they were within one hop of each other. There was no movement. Initially, each node transmitted position report messages. Then, each node alternatively updated its message type and size with the intervals of 0, 3, 5, 10, 15, 20, 25, 30, and 60 seconds respectively. This set up was to investigate how the time interval between each update would affect the transmission update time. The test program for measuring the transmission update time is included in Appendix B.2.3.

Figure 6-3 illustrates average transmission update time versus update interval. Figure 6-3 implies that the average transmission update time is higher when all nodes updated transmissions at the same time. The average transmission update time was fluctuated around 4 seconds, when nodes did not update transmissions at the same time.

![Figure 6-3: Average transmission update time versus update interval](image-url)
Figure 6-4: Average transmission update time versus number of existing nodes

Other test scenarios were set up with different numbers of nodes to investigate the effect of the number of existing nodes on the transmission update time. This test scenario used the fixed update interval of 30 seconds. Figure 6-4 illustrates average transmission update time versus number of existing nodes. Figure 6-4 implies that the average transmission update time was higher, when there was less number of existing nodes.

6.2.4 Time Slot Conflict Resolve Time

The time slot conflict resolve time is the time that the MAC protocol must spend to resolve time slot conflicts once they occur. The time slot conflict resolve time was measured from the running number of time frames between a conflicted frame and a conflict-free frame. There were two scenarios for measuring the time slot conflict resolve time. The first scenario was set up such that conflicts occurred when a node moved into a hidden network area between two other nodes. This test scenario contained three nodes. Two nodes were set as ground stations transmitting the RAP messages. They were placed beyond one hop of each other, but there was the hidden network area between these two ground stations. The third node was set as a mobile node transmitting the position report messages. Initially, the mobile node was outside the hidden network area. Then, the mobile moved into the hidden network area and the time slot conflicts would occur. The number of conflicted time slots in the frame was counted and the time slot conflict resolve time was measured. Then, the size of RAP messages of both ground stations
were changed to adjust the number of conflicted time slots in a frame, and the time slot conflict resolve time was measured as the number of conflicted time slots in a frame changed. The test program for this scenario is included in Appendix B.2.4.

Table 6-2 presents time slot conflict resolve time versus number of conflicted time slots in a frame for the first time slot conflict scenario. The result implies that time slot conflict resolve time was always one time frame.

<table>
<thead>
<tr>
<th>Number of conflicted time slots in a frame</th>
<th>Resolve time (# of time frame)</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
</tr>
<tr>
<td>12</td>
<td>1</td>
</tr>
<tr>
<td>13</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 6-2: Time slot conflict resolve time versus number of conflicted time slots in a frame for the first time slot conflict scenario

The second test scenario simulated the situation where a node left the network and returned to the network after a time slot reallocation had occurred. This test scenario was set up with four nodes. One node was set as a ground station and was transmitting the RAP messages. One mobile node was set to transmit the track messages, and other mobile nodes were set to transmit the position report messages. A mobile node with the track messages then left the network and returned after a time slot reallocation had occurred. The number of conflicted time slots in a time frame was counted and the time slot conflict resolve time was measured. Then, size of the RAP messages of the ground stations were changed to adjust the number of conflicted time slots in a frame, and the time slot conflict resolve times were measured as the number of conflicted time slots in a frame changed. The test program for this testing scenario is included in Appendix B.2.4.

Table 6-3 presents time slots conflict resolve time versus number of conflicted time slots in a frame for the second time slot conflict scenario. The result from this scenario also implies that the time slot conflict resolve time was always one time frame. The second test scenario simulates the different trait that will cause time slot conflicts. The results from both scenarios show that the designed self-organized TDMA protocol was able to resolve time slot conflicts as quickly as one time frame.
<table>
<thead>
<tr>
<th>Number of conflicted time slots in a frame</th>
<th>Resolve time (# of time frame)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 6-3: Time slots conflict resolve time versus number of conflicted time slots in a frame for the second test scenario

### 6.2.5 Maximum Achievable System Throughput, Channel Efficiency, and Maximum Channel Utilization

The maximum achievable system throughput of the simulated TDL systems is defined in term of the maximum possible number of transmitted data bits per second from all nodes within two hops. For the self-organized TDMA protocol, all data slots can be distributed and utilized. In contrast, the preprogramming TDMA protocol only allows the data slots reserved for the existing nodes in the network to be utilized.

The maximum achievable system throughput for the self-organized TDMA protocol can be calculated as follows.

\[
TP_{max} = \frac{Ns(Nb-Nh)}{t_f} \tag{6.2}
\]

\(TP_{max}\) is the maximum achievable system throughput in Kbits/s.

\(Ns\) is the number of useable data slots in a frame.

\(Nb\) is the number of data bits that can be fitted in one time slot.

\(Nh\) is the number of bits for the MAC frame header.

\(t_f\) is the time frame duration in seconds

The maximum achievable system throughput for the preprogramming TDMA protocol can be calculated as follows.

\[
TP_{max} = \frac{Nr(Nb-Nh)}{t_f} \tag{6.3}
\]

\(TP_{max}\) is the maximum achievable system throughput in Kbits/s.

\(Nr\) is the number of data slots reserved for the existing nodes in the network.

\(Nb\) is the number of data bits that can be fitted in one time slot.

\(Nh\) is the number of bits for the MAC frame header.
\( t_f \) is the time frame duration in seconds

Figure 6-5 illustrated calculated maximum achievable system throughputs versus number of existing nodes in the network for both self-organized TDMA and preprogramming TDMA protocols according to the simulation parameters. According to Figure 6-5, the first and second nodes were assumed to be ground stations and all other nodes were assumed to be aircrafts. In the case of the preprogramming TDMA protocol, time slots were pre-allocated to each node according to the time slot schedule illustrated in Figure 5-6. The first and second nodes were in the first group of the time slot schedule, and each node in this group reserved 20 time slots in each time frame. The third through tenth nodes were in the second group of the time slot schedule, and each node in this group reserved 12 time slots in each time frame. The eleventh through sixteenth nodes were in the third group of the time slot schedule, and each node in this group reserved 8 time slots in each time frame. Hence, Figure 6-5 shows the different slopes for each group in the preprogramming TDMA protocol when the nodes from different groups were added in the network. Note that the different orders of added nodes will give different slopes for the maximum achievable system throughput.

The channel efficiency of the simulated TDL system is defined as the percentage of the maximum achievable system throughput to the channel capacity. Figure 6-6 illustrates calculated channel efficiency for both MAC protocols versus number of nodes in the net. The maximum Channel utilization of the simulated TDL system is defined as the percentage of the maximum possible transmitted data bits including MAC headers per second to the channel capacity. Figure 6-7 illustrates calculated maximum channel utilization for both MAC protocols versus number of nodes in the net. In this simulation, the channel is assumed to be a noise-free channel. Hence, the channel capacity can be assumed to be equal to the channel bandwidth (50 Kbits/s).

According to Figure 6-5, neither of the two protocols can reach the channel capacity of 50 Kbits/s. In addition, the channel efficiency and the maximum channel utilization of both protocols cannot reach 100%. This is because there are some overheads in each time frame, in which cannot be used to transmit data.

Figure 6-5 shows that the calculated maximum achievable system throughput for the self-organized protocol is constant at 37.848 Kbits/s, and the calculated maximum achievable system throughput for the preprogramming protocol is increasing as the number of nodes increased. When the number of nodes in the network is below 14 nodes, the self-organized TDMA protocol gives higher maximum achievable system throughput than that of the preprogramming TDMA protocol. When there are 16 nodes in the net, the preprogramming TDMA protocol can give the maximum achievable throughput as high as 42.835 Kbits/s.
Figure 6-5: Calculated maximum achievable throughput versus number of nodes

Figure 6-6: Calculated channel efficiency versus the number of nodes
Two scenarios were simulated to measure the performance of the self-organized TDMA protocol and the preprogramming TDMA protocol in terms of the maximum achievable system throughput for the given numbers of nodes. The maximum achievable system throughput in the given scenario can be determined from the instantaneous system throughput in steady state, where reserved data slots were overloaded with large messages and there is no more throughput gain.

The instantaneous system throughput for each time frame was measured as follows.

\[
TP_{ins} = \frac{Nf - No}{t_f}
\]

(6.4)

\(TP_{ins}\) is the instantaneous system throughput for a time frame in Kbits/s

\(Nf\) is the total number of bits transmitted in a time frame.

\(No\) is the total number of transmitted MAC frame header bits in a time frame.

\(t_f\) is the time frame duration in seconds.

The first scenario was simulated with eight aircraft nodes (Node 3 to Node 10) within one hop of each other. Each node increasingly changed its transmitted message size until its reserved data...
slots were overloaded. Table 6-4 describes the nodes’ activities of this scenario. Figure 6-8 illustrates instantaneous system throughput for each time frame versus simulation time. This figure shows that the instantaneous system throughput for the self-organized TDMA protocol was converged to 37.85 Kbits/s as large messages were injected into the system. In contrast, the instantaneous system throughput for the preprogramming TDMA protocol converged to 22.35 Kbits/s as messages with large size were injected into the system. This scenario gave higher maximum achievable system throughput for the self-organized TDMA protocol than that of the preprogramming TDMA protocol, since the self-organized TDMA protocol had more usable data slots in a time frame.

The second scenario was simulated with 2 ground stations (Node 1 and Node 2) and 14 aircrafts (Node 3 to Node 16) within one hop of each other. Each node increasingly changed its transmitted message size until its reserved data slots were overloaded. Table 6-5 describes the nodes’ activities of this scenario. Figure 6-9 illustrates instantaneous system throughput for each time frame versus simulation time. This figure shows that the instantaneous system throughput for the self-organized TDMA protocol converged to 37.85 Kbits/s as large messages were injected into the system. In contrast, the instantaneous system throughput for the preprogramming TDMA protocol converged to 42.84 Kbits/s as large messages were injected into the system. This scenario gave higher maximum achievable system throughput for the preprogramming TDMA protocol than that of the self-organized TDMA protocol. When both protocols uses most of data slots in each time frame, the preprogramming TDMA protocol is expected to give higher throughput because it has smaller overhead.

<table>
<thead>
<tr>
<th>Time (second)</th>
<th>Node Number</th>
<th>Activities</th>
</tr>
</thead>
<tbody>
<tr>
<td>1st – 50th</td>
<td>3-10</td>
<td>Start TDL system and send position report message</td>
</tr>
<tr>
<td>60th</td>
<td>3-10</td>
<td>Move to specific point</td>
</tr>
<tr>
<td>150th – 300th</td>
<td>3-10</td>
<td>Change to track message with message size of 200 bytes</td>
</tr>
<tr>
<td>500th – 700th</td>
<td>3-10</td>
<td>Change message size to 800 bytes</td>
</tr>
<tr>
<td>900th – 1100th</td>
<td>3-10</td>
<td>Change message size to 1000 bytes</td>
</tr>
<tr>
<td>1400th – 1600th</td>
<td>3-10</td>
<td>Change message size to 1200 bytes (overload reserved data slots)</td>
</tr>
<tr>
<td>1800th – 2000th</td>
<td>3-10</td>
<td>Change message size to 1500 bytes (overload reserved data slots)</td>
</tr>
</tbody>
</table>

Table 6-4: Nodes’ activities in the first scenario of measuring the maximum achievable system throughput
Table 6-5: Nodes’ activities in the second scenario of measuring maximum achievable system throughput

<table>
<thead>
<tr>
<th>Time (second)</th>
<th>Node Number</th>
<th>Activities</th>
</tr>
</thead>
<tbody>
<tr>
<td>1st – 10th</td>
<td>1-2</td>
<td>Start TDL system and send RAP message with message size of 5200 bytes (overload reserved data slots for ground stations)</td>
</tr>
<tr>
<td>10th – 50th</td>
<td>3-9</td>
<td>Start TDL system and send position report message</td>
</tr>
<tr>
<td>50th – 100th</td>
<td>10-16</td>
<td>Start TDL system and send track message with message size of 1200 bytes (overload reserved data slots)</td>
</tr>
<tr>
<td>120th</td>
<td>3-16</td>
<td>Move inside one hop distance of two ground stations</td>
</tr>
<tr>
<td>150th – 300th</td>
<td>3-9</td>
<td>Change message size to 200 bytes</td>
</tr>
<tr>
<td>1600th – 1800th</td>
<td>3-9</td>
<td>Change message size to 1000 bytes</td>
</tr>
<tr>
<td>2300th – 2500th</td>
<td>3-9</td>
<td>Change message size to 1200 bytes (overload reserved data slots)</td>
</tr>
<tr>
<td>3000th – 3200th</td>
<td>3-9</td>
<td>Change message size to 1500 bytes (overload reserved data slots)</td>
</tr>
</tbody>
</table>

Figure 6-8: Instantaneous system throughput versus simulation time in the first scenario of measuring maximum achievable system throughput
Figure 6-9: Instantaneous system throughput versus simulation time in the second scenario of measuring maximum achievable system throughput

6.3 Performance of the Self-Organized TDMA protocol in the Study Scenarios

The performances of both self-organized TDMA and preprogramming TDMA protocols in the study scenarios presented in Section 3.1 were measured. In the defensive scenario, 7 nodes were set up. Three nodes were set up as ground stations, i.e. Node 0, Node 1 and Node 2 respectively. Initially, Node 3 and Node 4 represented two aircrafts located inside the coverage area of Node 0, while Node 5 and Node 6 represented two aircrafts located inside the coverage area of Node 2. Node 1 represented a ground station situated between Node 0 and Node 2 as illustrated in Section 3.1.1. The movement profiles and the transmission profiles of each node were setup according to the descriptions in Section 3.1.1. In addition, a node with the track messages gradually increased the message size until its reserved data slots were overloaded with large messages. Note that reserved data slots will be overloaded when the track messages’ size is increased over 1000 bytes. This setting was to investigate the behaviors of the MAC protocols in the burdened situation. Table 6-6 shows the nodes’ activities in the simulated defensive scenario and the corresponded functions for the self-organized TDMA protocol. The test programs for the defensive scenario configured with the self-organized TDMA protocol and the preprogramming TDMA protocol are included in Appendix B.2.5.
<table>
<thead>
<tr>
<th>Time (second)</th>
<th>Node Number</th>
<th>Activities</th>
<th>Self-Organized TDMA’s functions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1\textsuperscript{st}</td>
<td>0,1,2</td>
<td>Start TDL system and send RAP message</td>
<td>Perform Net Entry</td>
</tr>
<tr>
<td>16\textsuperscript{th} – 46\textsuperscript{th}</td>
<td>3,4,5,6</td>
<td>Start TDL system and send position report</td>
<td>Perform Net Entry</td>
</tr>
<tr>
<td>60\textsuperscript{th} – 62\textsuperscript{nd}</td>
<td>5,6</td>
<td>Start moving from Node 2’s coverage area to Node 1’s coverage area</td>
<td>Resolve time slot conflict while passing hidden network area, Perform Net leaving when they are beyond Node 2’s coverage area</td>
</tr>
<tr>
<td>65\textsuperscript{th} – 67\textsuperscript{nd}</td>
<td>3,4</td>
<td>Start moving from node 0’s coverage area to node 1’s coverage area</td>
<td>Resolve time slot conflict while passing hidden network area, Perform Net leaving when they are beyond node 2’s coverage area</td>
</tr>
<tr>
<td>1500\textsuperscript{th}</td>
<td>3</td>
<td>Moving back to node 0</td>
<td>Resolve time slot conflict while passing hidden network area, Perform Net leaving when they are beyond node 1’s coverage area</td>
</tr>
<tr>
<td>1500\textsuperscript{th}</td>
<td>4,5,6</td>
<td>Move to the battle zone outside coverage area of any ground station</td>
<td>Perform Net leaving when they are beyond node 1’s coverage area</td>
</tr>
<tr>
<td>1640\textsuperscript{th} – 1650\textsuperscript{th}</td>
<td>5,6</td>
<td>Change message type from position report message to track message (400 bytes)</td>
<td>Perform transmission update</td>
</tr>
<tr>
<td>1840\textsuperscript{th} – 1860\textsuperscript{th}</td>
<td>5,6</td>
<td>Change track message size to 1000 bytes</td>
<td>Perform transmission update</td>
</tr>
<tr>
<td>2220\textsuperscript{th} – 2240\textsuperscript{th}</td>
<td>5,6</td>
<td>Change track message size to 1200 bytes</td>
<td>Perform transmission update</td>
</tr>
<tr>
<td>2640\textsuperscript{th} – 2660\textsuperscript{th}</td>
<td>5,6</td>
<td>Change track message size to 1500 bytes</td>
<td>Perform transmission update</td>
</tr>
<tr>
<td>3030\textsuperscript{th} – 3060\textsuperscript{th}</td>
<td>5,6</td>
<td>Change track message size to 1800 bytes</td>
<td>Perform transmission update</td>
</tr>
</tbody>
</table>

Table 6-6: Nodes’ activities in the simulation according to the defensive scenario

The offensive scenario was simulated with 12 aircraft nodes, i.e. Node 3 to Node 14. Aircrafts were divided into four groups, and each group started from different bases as described in Section 3.1.2. The first group contained Node 3 to Node 5. The second group contained Node 6 to Node 8. The third group contained Node 9 to Node 11. The fourth group contained Node 12 to Node 14. The movement profiles and the transmission profiles of each node were setup according to the description in Section 3.1.2. To investigate the behaviors of the MAC protocols in the burdened situation, a node with track messages gradually increased its message size until its reserved data slots were overloaded. In the offensive scenario, more nodes were involved in the transmission update and simulated a more congested network than the defensive scenario. Table 6-7 shows the nodes’ activities in the simulated offensive scenario and the corresponded functions for the self-organized TDMA protocol. The test programs for the offensive scenario configured with the self-organized TDMA protocol and the preprogramming TDMA protocol are included in Appendix B.2.5.
<table>
<thead>
<tr>
<th>Time (second)</th>
<th>Node Number</th>
<th>Activities</th>
<th>Self-Organized TDMA’s functions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1(^{st}) – 15(^{th})</td>
<td>3-14</td>
<td>Start TDL system and send position report message</td>
<td>Perform Net Entry</td>
</tr>
<tr>
<td>60(^{th})</td>
<td>3-14</td>
<td>Move to entry point</td>
<td>-</td>
</tr>
<tr>
<td>1050(^{th})</td>
<td>3-14</td>
<td>Move to enemy zone</td>
<td>-</td>
</tr>
<tr>
<td>1150(^{th}) – 1260(^{th})</td>
<td>3-10</td>
<td>Change message type from position report message to track message (400 bytes)</td>
<td>Perform transmission update</td>
</tr>
<tr>
<td>1550(^{th}) – 1630(^{th})</td>
<td>3-7</td>
<td>Change track message size to 800 bytes</td>
<td>Perform transmission update</td>
</tr>
<tr>
<td>2080(^{th}) – 2110(^{th})</td>
<td>8-10</td>
<td>Change track message size to 800 bytes</td>
<td>Perform transmission update</td>
</tr>
<tr>
<td>2450(^{th}) – 2580(^{th})</td>
<td>3-7</td>
<td>Change track message size to 1000 bytes</td>
<td>Perform transmission update</td>
</tr>
<tr>
<td>3010(^{th}) – 3090(^{th})</td>
<td>3-6</td>
<td>Change track message size to 1200 bytes</td>
<td>Perform transmission update</td>
</tr>
<tr>
<td>3410(^{th}) – 3490(^{th})</td>
<td>3-6</td>
<td>Change track message size to 1500 bytes</td>
<td>Perform transmission update</td>
</tr>
</tbody>
</table>

Table 6-7: Nodes’ activities in the simulation according to the offensive scenario

The situation where reserved data slots are overloaded, are the limitation of the MAC protocol in handling high demands for a bandwidth. This limitation can be compensated with an improved radio, which can provide a high data rate with a reduced coverage area. Both study scenarios were simulated again with the self-organized TDMA protocol and the improved radio specification. This improved radio specification provided the data rate of 100 Kbits/s, and the coverage area was reduced to 150 km. The maximum size of a RAP message and a track message were increased to 10000 bytes and 2000 bytes respectively. Hence, a track message with any size between 1000 bytes to 2000 bytes will not overload reserved data slots in the self-organized TDMA protocol with a 100 Kbits/s data rate. The test programs for both study scenarios configured with the self-organized TDMA protocol and a 100 Kbits/s data rate are included in Appendix B.2.5.

The performances of the self-organized TDMA protocol with a 50 Kbits/s data rate, the preprogramming TDMA protocol with a 50 Kbits/s data rate, and the self-organized TDMA protocol with a 100 Kbits/s data rate in the study scenarios can be measured and compared from the packet delay at the MAC sublayer, the instantaneous system throughput, the message update rate, and the time slot utilization.

### 6.3.1 Packet Delay

The packet delay is defined as the time from packet arriving at the MAC sublayer to the time it being sent. Figure 6-10a and Figure 6-10b illustrates the packet delays of the transmitted packets for all three configurations in the defensive scenario. Figure 6-11a and Figure 6-11b illustrates the packet delays of the transmitted packets for all three configurations in the offensive scenario.
It can be seen from Figure 6-10a and Figure 6-11a that there were sharp peaks of the packet delays for both setups of the self-organized TDMA protocol at the beginning of both scenarios, which occurred when the self-organized protocols performed the net entry process.

According to the figures packet delays were fluctuated. These fluctuations can be explained by the position of slots reserved in a frame. When there are many consecutive slots reserved, the delay will be lower because it can send many packets consecutively. In contrast, if reserved slots are separated, the delay will be higher because a packet has to wait for the next available slot to be transmitted. Figure 6-11b illustrates that packet delays of the self-organized TDMA protocol with a 50 Kbits/s data rate and a 100 Kbits/s data rate are constant at 2 seconds during 500th packet and 600th packet, when only position report messages were transmitted. This is because the self-organized TDMA protocol reserve data slots based on the required update rate. Hence, it will reserve data slots such that each data slot is separated by 2 seconds and packets will always be sent in the interval of 2 seconds.

![Figure 6-10a: Packet delays of the first 1000 packets in the defensive scenario](image-url)
Figure 6-10b: Packet delays of the 500th packet to the 600th packet in the defensive scenario

Figure 6-11a: Packet delays of the first 1000 packets in the offensive scenario
Figure 6-11b: Packet delays of the 500\textsuperscript{th} packet to the 600\textsuperscript{th} packet in the offensive scenario

<table>
<thead>
<tr>
<th>MAC protocol</th>
<th>Average Packet Delay (seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Defensive Scenario</td>
</tr>
<tr>
<td>Self-Organized TDMA</td>
<td>0.556</td>
</tr>
<tr>
<td>Preprograming TDMA</td>
<td>0.595</td>
</tr>
<tr>
<td>Self-Organized TDMA with double data rate</td>
<td>0.704</td>
</tr>
</tbody>
</table>

Table 6-8: Average packet delay in both study scenarios

The average packet delays for all three configurations in the study scenarios are presented in Table 6-8. According to Table 6-8, the self-organized TDMA protocol had the smallest average packet delays than the other configurations in both study scenarios. However, the average packet delays in the same study scenario did not show much difference between each configuration.

6.3.2 Instantaneous System Throughput

The instantaneous system throughput in each time frame is defined as Equation (6.4). Figure 6-12a to Figure 6-12c illustrate instantaneous system throughputs for each time frame versus simulation time for all three configurations in the defensive scenario. Figure 6-13a to Figure 6-13c illustrate instantaneous system throughputs for each time frame versus simulation time for all three configurations in the offensive scenario.
According to Figure 6-12b and Figure 6-13b, the instantaneous system throughputs for the preprogramming TDMA protocol with a 50 Kbits/s data rate were better than the instantaneous system throughputs for the self-organized TDMA protocol with a 50 Kbits/s data rate in the first 10 seconds and the first 30 seconds of the defensive scenario and the offensive scenario respectively. This result occurred when nodes in the self-organized TDMA system had to perform the net entry function. However, the instantaneous system throughput of the self-organized TDMA protocol with a 50 Kbits/s data rate was equal to or better than the instantaneous system throughput of the preprogramming TDMA protocol with a 50 Kbits/s data rate after the net entry process and later on as illustrated in Figure 6-12b, Figure 6-12c, Figure 6-13b, and Figure 6-13c.

According to Figure 6-12a, Figure 6-12b, Figure 6-12c, Figure 6-13a, Figure 6-13b, and Figure 6-13c, the instantaneous system throughputs in each time frame of the self-organized TDMA protocol with a 100 Kbits/s data rate in both study scenarios were lower than that of the other two configurations, when small RAP messages (lower than or equal to 5000 bytes) or small track messages (lower than or equal to 1000 bytes) were transmitted. This is because the self-organized TDMA protocol with a 100 Kbits/s data rate transmits messages at higher rate. As a result, some parts of messages had already been transmitted in previous time frames and the instantaneous system throughputs in the current frame will be lower than that of the other two protocols. Figure 6-12b illustrates that the instantaneous system throughputs in each time frame of the self-organized TDMA protocol with a 100 Kbits/s data rate was higher at the beginning, and then it started to drop to lower value.

![Figure 6-12a: Instantaneous system throughputs in the defensive scenario](image-url)
Figure 6-12b: Instantaneous system throughputs during 0 to 100 seconds in the defensive scenario

Figure 6-12c: Instantaneous system throughputs during 1500 to 3500 seconds in the defensive scenario
Figure 6-13a: Instantaneous system throughputs in the offensive scenario

Figure 6-13b: Instantaneous system throughputs during 0 to 100 seconds in the offensive scenario
Figure 6-13c: Instantaneous system throughputs during 1000 to 3500 seconds in the offensive scenario

The instantaneous system throughputs for every configuration increased significantly, when nodes with track messages increased the track messages’ size as illustrated in Figure 6-12c and Figure 6-13c. However, there was no more gain in the throughput when the track message’s size was increased more than the size of reserved data slots for a track message. This was illustrated for the self-organized TDMA protocol with a 50 Kbits/s data rate in Figure 6-12c at the 2640th second and the 3030th second when the track messages’ size was increased to 1500 bytes and 1800 bytes respectively, and Figure 6-13c at the 3400th second when the track messages’ size was increased to 1500 bytes. However, the self-organized TDMA protocol with a 100 Kbits/s data rate can overcome this limitation. As illustrated in Figure 6-12c and Figure 6-13c, the self-organized TDMA protocol with a 100 Kbits/s data rate gained higher throughput when there were demands on transmitting messages with a very large size.

The instantaneous system throughput of the self-organized TDMA protocols was even growing faster than that of the preprogramming TDMA protocol when the message’s size was increased to a very large size. This is illustrated in Figure 6-12c at the 1840th second and Figure 6-13c at the 1500th second. When the message size is small, there is no obvious difference in the throughput. The message transmission rate in the preprogramming TDMA protocol will fluctuate according to the time slot schedule. Small messages can be sent faster when time slots are reserved consecutively, whereas messages will be sent slower when reserved time slots are separated. In contrast, messages in the self-organized TDMA protocol will always be sent at the
required update rate. By average, the message transmission rate of both protocols will become closer to each other when the message size is small. When the message size is large, the message transmission rate of the preprogramming TDMA protocol will be slower because it cannot send large messages in consecutive slots. In contrast, the message transmission rate of the self-organized TDMA protocol will not be affected because it always sends messages at the required update rate. Hence, the instantaneous system throughput for the self-organized TDMA protocol will be much larger than the instantaneous system throughput for the preprogramming TDMA protocol when the message size is large.

6.3.3 Message Update Rate

The message update rate is the received message rate of the same message type from the same transmitter. There are three types of messages in the study scenarios, i.e. RAP, track, and position report. The simulation in the defensive scenario implies that the self-organized TDMA protocol with a 50 Kbits/s data rate, the preprogramming TDMA protocol with a 50 Kbits/s data rate, and the self-organized TDMA protocol with a 100 Kbits/s data rate were able to transmit RAP messages with the required update rate of one message in every 10 seconds throughout the simulation as long as the RAP messages’ size is below the maximum allowed size (5000 bytes with a 50 Kbits/s data rate and 10000 bytes with a 100 Kbits/s data rate).

The behaviors of the track messages’ update rate are similar for all aircraft nodes. Therefore, only the result from one aircraft node is illustrated. The track messages’ update rates for Node 6 in the defensive scenario versus the simulation time are illustrated in Figure 6-15. Figure 6-16 illustrates the track messages’ update rates of Node 6 in the offensive scenario. These figures imply that the self-organized TDMA protocol with a 50 Kbits/s data rate was able to transmit track messages under the required update rate of one message in every 2 seconds as long as the track messages’ size did not overloaded reserved data slots. When the track messages’ size overloads the reserved data slots, the track messages’ update rate in the self-organized TDMA protocol will be over 2 seconds. This can be seen from Figure 6-15 for the defensive scenario. It is also illustrated in Figure 6-16 for the offensive scenario. The overloaded problem in the self-organized TDMA protocol with a 50 Kbits/s data rate was solved by the self-organized TDMA protocol with a 100 Kbits/s data rate. Figure 6-15 and Figure 6-16 illustrate that the self-organized TDMA protocol with a 100 Kbits/s data rate was able to transmit track messages with the required update rate, when the track messages’ size was increased to a very large size (between 1000 bytes to 2000 bytes).

Figure 6-15 and Figure 6-16 illustrate that the track messages’ update rates in the system with the preprogramming TDMA protocol fluctuated between low and high values. This can be explained by the structure of the time slot schedule as described in Figure 5-6. When the message size is small, many messages can be transmitted in the consecutive time slots. However, messages will be transmitted with longer delay if reserved time slots are separated. According to the time slot schedule in Figure 5-6, time slots reserved by each aircraft can be separated by either 2 seconds
or 4 seconds. Hence, the update rate will be very low when small messages are transmitted in the consecutive time slots and the update rate will be either 2 seconds or 4 seconds when messages are transmitted in separated time slots. When the message size is larger than the size of consecutive time slots, messages will be transmitted with the delay of either 2 seconds or 4 seconds. Hence, not all track messages will be transmitted with the required update rate in the preprogramming TDMA protocol.

The behaviors of the position report messages’ update rate are similar for all aircraft nodes. Therefore, only the result from one aircraft node is illustrated. The position report messages’ update rates for Node 3 in the defensive scenario as a simulation time are illustrated in Figure 6-17. Figure 6-18 illustrates the position report messages’ update rates of Node 3 in the offensive scenario. These figures imply that both self-organized TDMA protocol with a 50 Kbits/s data rate and self-organized TDMA protocol with a 100 Kbits/s data rate were able to transmit position report messages with the required update rate of one message in every 2 seconds. The preprogramming TDMA protocol transmitted position reports with fluctuated rates due to the structure of the time slot schedule explained earlier. Hence, the position reports’ update rates of the preprogramming TDMA protocol can be very low when position reports were transmitted in the consecutive time slots, and the position reports’ update rates can be either 2 seconds or 4 seconds when position report messages were transmitted in the separated time slots.

Figure 6-15: Track’s update rate of Node 6 in the defensive scenario during 2150th second and 2300th second
Figure 6-16: Track’s update rate of Node 6 in the offensive scenario during 2450\textsuperscript{th} second and 2600\textsuperscript{th} second

Figure 6-17: Position report’s update rate of Node 3 in the defensive scenario during 100\textsuperscript{th} second and 150\textsuperscript{th} second
6.3.4 Time Slot Utilization

The time slot utilization is the percentage of time slots used for transmitting data to time slots reserved in the system. Figure 6-19a, Figure 6-19b, and Figure 6-19c illustrate the time slot utilizations for the self-organized TDMA protocol with a 50 Kbits/s data rate, the preprogramming TDMA protocol with a 50 Kbits/s data rate, and the self-organized TDMA protocol with a 100 Kbits/s data rate in the defensive scenario respectively. Figure 6-20a, Figure 6-20b, and Figure 6-20c illustrate time slot utilizations for the self-organized TDMA protocol with a 50 Kbits/s data rate, the preprogramming TDMA protocol with a 50 Kbits/s data rate, and the self-organized TDMA protocol with a 100 Kbits/s data rate in the offensive scenario respectively.

As illustrated in Figure 6-19b, RAP messages and position report messages were transmitted during 0 to 500 seconds in the defensive scenario. The sizes of RAP messages and position report messages were 5000 bytes and 50 bytes respectively. The self-organized TDMA protocol with a 50 Kbits/s data rate required all reserved data slots in a time frame to be used in order to transmit these messages. Hence, the time slot utilization for the self-organized TDMA protocol with a 50 Kbits/s data rate was always 100% during this period. In contrast, a time slot in the self-organized TDMA protocol with a 100 Kbits/s data rate can hold more bytes than a time slot in the self-organized TDMA protocol with a 50 Kbits/s data rate. The self-organized TDMA
protocol with a 100 Kbits/s data rate did not required all reserved data slots in a time frame to be used to transmit a 5000-byte RAP message. Hence, the time slot utilization for the self-organized TDMA protocol with a 100 Kbits/s data rate was less than 100 % during this period.

Figure 6-20b illustrates the time slot utilizations during 0 to 500 seconds in the offensive scenario, in which only position report messages were sent. The size of position report message is always 50 bytes, in which both self-organized TDMA protocol with a 50 Kbits/s data rate and self-organized TDMA protocol with a 100 Kbits/s data rate reserve data slots base on this size. Hence, the time slot utilization for both self-organized TDMA protocol with a 50 Kbits/s data rate and self-organized TDMA protocol with a 100 Kbits/s data rate were always 100 % during this period.

Figure 6-19c and Figure 6-20c illustrate the time slot utilizations during track messages had been transmitted in the defensive scenario and the offensive scenario respectively. Both self-organized TDMA protocol with a 50 Kbits/s data rate and self-organized TDMA protocol with a 100 Kbits/s data rate reserve data slots for track messages base on the maximum track message size of 1000 bytes and 2000 bytes respectively. The time slot utilization for both self-organized TDMA protocol with a 50 Kbits/s data rate and self-organized TDMA protocol with a 100 Kbits/s data rate were dropped, when the nodes changed from transmitting position report to transmitting small track messages. When larger track messages had been transmitted, more reserved data slots were used and the time slot utilizations for both self-organized TDMA protocol with a 50 Kbits/s data rate and self-organized TDMA protocol with a 100 Kbits/s data rate were increasing.

The number of reserve time slots in the preprogramming TDMA protocol is constant and independence of transmitted message type. Hence, the time slot utilization of the preprogramming TDMA protocol increased, when the size of messages was increased as illustrated in Figure 6-19a, Figure 6-19b, Figure 6-19c, Figure 6-20a, Figure 6-20b, and Figure 6-20c.
Figure 6-19a: Time slot utilizations in the defensive scenario

Figure 6-19b: Time slot utilizations during 0 to 500 seconds in the defensive scenario
Figure 6-19c: Time slot utilizations during 1500 to 3000 seconds in the defensive scenario

Figure 6-20a: Time slot utilizations in the offensive scenario
Figure 6-20b: Time slot utilizations during 0 to 500 seconds in the offensive scenario

Figure 6-20c: Time slot utilizations during 1000 to 2500 seconds in the offensive scenario
Chapter 7

Discussions

7.1 Performances of the Self-Organized TDMA protocol

The adaptability performance of the self-organized TDMA protocol can be investigated in terms of the net entry time, the net leaving reallocation time, the transmission update time, and the time slots conflict resolve time. These parameters are only valid for the self-organized TDMA protocol, and cannot be measured from the preprogramming TDMA protocol. The network performance of the self-organized TDMA protocol can be investigated and compared with that of the preprogramming TDMA protocol in terms of the maximum achievable system throughput, the channel efficiency, and the maximum channel utilization. In addition, the performance of the self-organized TDMA protocol in the defined study scenarios can be investigated and compared with that of the preprogramming TDMA protocol in terms of the average packet delay, the instantaneous system throughput, the message update rate, and the time slot utilization.

7.1.1 Net Entry

According to the protocol’s description in Section 4.5.2, the net entry time may be depended on the chance of a virtual slot to win a control slot, the chance of an entry node to successfully transmit a net entry request, and the chance of an existing node to send a net entry acknowledgement.

The chance of a virtual slot to win a control slot is influenced by the number of existing node in the network and the hash values of neighbors within two hops, in which a hash value is calculated from a random seed assigned to each node and a slot position in a time frame. When there is no existing node in the network, an entry node does not have to send a net entry request and wait for a net entry ACK. Hence, the net entry time should be very short. In contrast, the net entry time should be longer when there are any existing nodes in the net. In addition, the chance of finding a net entry slot is reduced as the number of existing node is increasing. This is because a virtual node has lower probability to win a control slot. However, using multiple virtual nodes as described in Section 4.5.2 can compensate for the lowered probability of a virtual node winning a control slot. An entry node can use a control slot as a net entry slot, when any virtual slot wins a control slot. As a result, the chance that an entry node would find a net entry slot will not be reduced significantly. Thus, increasing the number of existing nodes in the net will not strongly influence the net entry time. The result from Section 6.2.1 shows that the average net entry time tends to converge as the number of existing nodes is increasing.

The result from Section 6.2.1 implies that the net entry time is strongly influenced by the shorter interval between each entry. Assuming that there is no error in the channel, a net entry request can be lost due to two conditions. The first condition is that there is a collision in a net entry slot.
The second condition is that a net entry request is dropped by a node that is processing another net entry request. When the interval between each entry is short, there is higher probability that these conditions would occur. Hence, the net entry time would be much longer because an entry node will repeatedly retransmit the requests on the next available net entry slots until it successfully transmits the request.

The net entry process is considered success, if the net entry acknowledgement is received. A net entry ACK will be transmitted on control slot by an existing node who receives a net entry request. Hence, the chance that a net entry ACK will be sent is depended on the chance that a control slot is won by the node with a net entry ACK. When the number of existing nodes is less than the number of virtual nodes, there is lower probability for an existing node with a net entry ACK to win a control slot and send a net entry ACK. The result from Section 6.2.1 demonstrates that the net entry times at lower numbers of existing nodes are higher. When the number of existing nodes is more than the number of virtual nodes, the probability that one of the existing nodes will win a control slot is increased. However, the existing nodes must compete with each other to win a control slot. As the number of existing nodes increases, the winning opportunity of each node becomes lower. Hence, the probability for an existing node sending a net entry ACK should become lower. However, the NAPA protocol compensates for this reduced probability. A node may not win a control slot, but it may receive a polling message indicating that it can transmit after a back off period. As a result, the lower number of existing nodes will result in higher net entry time due to higher winning probability of virtual nodes, while the higher number of existing nodes will not greatly affect the net entry time due to the compensation made by the NAPA mechanism.

### 7.1.2 Net Leaving

The result from Section 6.2.2 implies that the net leaving reallocation time is depended on the time instance in a time frame when a node is leaving the network. The net leaving reallocation time can be calculated as given in Equation (6.1). The remaining time in a time frame, when a node disappears, including two time frames are used by the existing nodes to determine that a node is actually leaving the network. Hence, an leaving node can be deleted from the neighbor list. The reason is explained as Figure 7-1. Before a node disappears, it had transmitted some messages on the first time frame. Hence, a leaving node is still considered as an existing node in the first time frame that it just disappeared. At the end of each time frame, a node looks for a neighbor that had neither transmitted nor been included in any MAC headers of received packets in the time frame, and deletes that neighbor from the neighbor list. Hence, a node that just disappears in the first time frame will be considered as a one-hop neighbor of other nodes by the end of the first time frame, and will not be deleted from the neighbor list. A leaving node will also be included in the MAC header of each packet transmitted in the second time frame, even though it has already disappeared from the net. At the end of the second time frame, a leaving node is considered as a two-hop neighbor of other nodes because its transmission had not been
detected by any existing nodes but it had been included in the MAC header of each packet transmitted in the second time frame. In the third time frame, a leaving node will not be included in the MAC header. Thus, the existing nodes will be able to delete a leaving node from the neighbor list at the end of the third time frame.

![Diagram showing 3 time frames: 1st time frame: Leaving Node transmits some data before disappearing. 2nd time frame: Leaving Node is not transmitting but it is included in MAC header. 3rd time frame: Leaving Node is neither transmitting nor included in MAC Header.]

Figure 7-1: Explanation of the net leaving process

### 7.1.3 Transmission Update

According to the protocol’s description in Section 4.2.4., the transmission update time may be depended on the chance of a node to send a transmission update message or a transmission update acknowledgement.

The chance of a node to send a transmission update message or a transmission update acknowledgement is related to the probability that a node would win a control slot. This probability is influenced by the number of existing nodes in the network and the hash values of each node. The lower number of existing nodes will result in higher transmission update time due to higher winning probability of virtual nodes. The higher number of existing nodes will result in the lower winning probability for a node, but the chance of transmitting control information will not be greatly affected due to the compensation made by the NAPA mechanism. This effect can be explained with the same reason as the effect of the number of existing node on sending a net entry ACK. The result from Section 6.2.3 implies that the transmission update times are higher when there is lower number of existing nodes, while the transmission update times are fluctuated around lower values when there is higher number of existing nodes.

When the update interval is short, there is a higher probability that there are multiple nodes trying to send control information on the same control slot. An updating node with a stronger hash value will be able to send control information on a control slot, and other updating nodes with weaker hash values have to wait longer. The result from Section 6.2.3 implies that the transmission update time is higher, when multiple nodes are trying to update the transmission characteristic at the same time. However, the chance that nodes will access control slot to send control information at the same time is quite low because the update control messages arrive at the MAC sublayer at different time for each node. The simulated system was implemented such that a packet can be read from a queue only after previous packet is sent. Because of the TDMA structure, each node send packet on different time slots. Therefore, an update control message
will be read from a queue at different time for each node and nodes will hardly access a control slot to send control information at the same time. The result from Section 6.2.3 only shows the high update time when nodes update at the same time.

7.1.4 Time Slot Conflict Resolution

The result from Section 6.2.4 implies that time slot conflicts can be resolved as quickly as one time frame. A conflict occurs because nodes within two hops are not updated with the recent neighbor information during a time frame. This occurs when some nodes move into a hidden network area, or there are some leaving nodes re-enter the network after leaving nodes have been deleted from the net. A node can collect new neighbor information from any transmitted packet containing the information about the conflicted node in any conflict-free data slot or conflict-free control slot. At the end of each time frame, each node will reallocate time slots with the updated neighbor information. Hence, a conflict will be resolved and there will be no collision in the next time frame after receiving information about the conflicted node.

7.1.5 Maximum Achievable System Throughput, Channel Efficiency, and Maximum Channel Utilization

Maximum achievable system throughput, channel efficiency, and maximum channel utilization are parameters that can be used to evaluate the network performance of the self-organized TDMA protocol. The calculation and simulation results from Section 6.2.5 implies that the maximum achievable system throughput, the channel efficiency, and the maximum channel utilization versus the number of existing nodes for the self-organized TDMA protocol are constant. The reason is that the self-organized protocol can allocate and use all data slots regardless of the number of nodes in the network. In contrast, the preprogramming TDMA protocol can only use the data slots reserved to the existing nodes in the net. Hence, the maximum achievable throughput, the channel efficiency, and the maximum channel utilization for the preprogramming protocol are increasing as the number of nodes in the net is increased. According to the result in Section 6.2.5, when the number of nodes in the net is below 14 nodes, the self-organized TDMA protocol performs much better than the preprogramming TDMA protocol in terms of the maximum achievable system throughput and the channel efficiency. In practical airborne scenarios, the number of node in operation is usually low. Mostly there are only four to six aircrafts and one ground station operating within two hops of each other. The scenarios, in which the preprogramming TDMA protocol with large number of nodes gives higher network performance, are usually rare in practical missions. In addition, the time slot schedule for the preprogramming TDMA protocol must be redesigned to handle more number of nodes.

The maximum channel utilization of the self-organized TDMA protocol is always higher than the maximum channel utilization of the preprogramming protocol. The reason is that there are 184 usable data slots in each time frame for the preprogramming TDMA protocol according to
the time slot schedule illustrated in Figure 5-6, whereas there are 190 usable data slots in each time frame for the self-organized TDMA protocol. In other word, the self-organized TDMA protocol can access the channel more often than the preprogramming TDMA protocol. However, the data slots in the self-organized TDMA protocol contain less data bits than the data slots in the preprogramming TDMA protocol because of the larger MAC header of the self-organized TDMA protocol. Hence, the maximum achievable system throughput and channel efficiency of the preprogramming protocol can be higher when the number of existing nodes in the net is higher and more data slots are used.

7.1.6 Performance of the self-organized TDMA protocol in the Study Scenarios

The performance of the self-organized TDMA protocol in the study scenarios can be evaluated and compared to the performance of the preprogramming TDMA protocol in terms of the packet delay at the MAC sublayer, the instantaneous system throughput in each time frame, the message update rate, and the time slot utilization. Note that the self-organized TDMA protocol may perform differently in different scenarios. The study scenarios defined in this thesis are based on the practical airborne missions; however the real missions may have different scenarios other than these study scenarios. The performance of the self-organized TDMA protocol is also limited by the radio specifications. In order to demonstrate the possible improvement of the self-organized TDMA protocol with the better radio specification, the performance of the self-organized TDMA protocol with a 100 Kbit/s data rate is investigated and compared with the performance of the self-organized TDMA protocol with a 50 Kbits/s data rate.

The results from Section 6.3.1 imply that the average packet delay for the self-organized TDMA protocol with a 50 Kbits/s data rate, the preprogramming TDMA protocol with a 50 Kbits/s data rate, and the self-organized TDMA protocol with a 100 Kbits/s are not significantly different. The delay of the first packet in the self-organized TDMA protocol is very high because a node has to wait for the net entry process to be completed before transmitting the first data packet. After the net entry process, packet delays in the self-organized TDMA protocol are about the same as packet delays in the preprogramming TDMA protocol.

The instantaneous system throughput measures the rate of transmitted information in the system during each time frame. The result from Section 6.3.2 shows that the instantaneous system throughput for the self-organized TDMA protocol is low during the net entry process. This is because there is no transmitted data packet during the net entry process. After a net entry process is completed, a node in the self-organized TDMA system can transmit message with the required update rate. The instantaneous system throughput of the self-organized TDMA protocol will be equal to the instantaneous system throughput of the preprogramming TDMA protocol, when the transmitted messages’ size is very small. In contrast, the instantaneous system throughput of the self-organized TDMA protocol will be much better than the instantaneous system throughput of
the preprogramming TDMA protocol, when the transmitted messages’ size is very large. However, there is the limit on the instantaneous system throughput of the self-organized TDMA protocol, in which it cannot be improved anymore as the messages’ size is increased. This problem can be solved by a radio with a higher data rate. According to the result from Section 6.3.2, the instantaneous system throughput of the self-organized TDMA protocol with a 100 Kbits/s can be further improved when the messages’ size is increased to a very large value. Hence, the limitation of the instantaneous system throughput for the self-organized TDMA protocol can be extended to the higher value with a radio that can provide a higher data rate. Most modern radios can switch among different data rates; hence they can be used to overcome the limitation of the self-organized TDMA protocol in term of the throughput. By increasing the data rate, the transmission range will be reduced and the TDL system will operate in a smaller network.

The message update rate measures the quality of services of the TDL system. In the military mission, it is very important to meet the required message update rate. In this TDL system, three types of the TDL messages are simulated, i.e. RAP, track, and position report. Track and position report require update in every 2 seconds, while RAP requires update in every 10 seconds. The result from Section 6.3.3 shows that the message update rates for all TDL messages in the self-organized TDMA protocol meet the required update rate, as long as the messages’ size does not exceed the maximum size of each message type. The self-organized TDMA protocol can transmit messages with the required update rate because the data slot assignment algorithm is designed based on the maximum size and the update rate of each message type. Data slots will be reserved in the position that will meet the require update rate. Unlike the preprogramming TDMA protocol, data slots are only reserved based on the maximum size of message and the maximum number of nodes in the net. The update rate cannot be met, because the time slot schedule of the preprogramming TDMA protocol must be designed in order to allocation data slots to all nodes. Hence, the update rate requirement cannot be met for the preprogramming TDMA protocol.

The time slot utilization measures the percentage of the time slot usage for data transmissions. The result from Section 6.3.4 implies that the self-organized TDMA protocol always use 100% of the reserved time slot given that messages are transmitted with the maximum message size for each message type. The time slot utilization of the self-organized TDMA protocol will be lower than 100%, if messages are transmitted with the message size that is lower than the maximum message size for each message type. In contrast, the preprogramming TDMA protocol cannot use data slots reserved by nodes that are not presented in the network. Hence, there are reserved data slots that are left unused in the preprogramming TDMA protocol, which is considered to be waste of bandwidth.
7.2 Strengths and Weaknesses of the Self-Organized TDMA Protocol

The main strength of the self-organized TDMA protocol is the ability to adapt the time slot usage to any changes in the network environment. Comparing to the preprogramming TDMA protocol, the time slot usage in the self-organized TDMA protocol is more effective. Throughput can be maximized regardless of the number of nodes. The maximum achievable system throughput of the self-organized TDMA protocol is better than that of the preprogramming TDMA protocol in the scenario with the fair amount number of nodes. For the given input parameters defined in this thesis, the maximum achievable system throughput of the self-organized TDMA protocol is better than that of the preprogramming TDMA protocol when number of nodes in the network is less than 14 nodes. Throughput gain of the self-organized TDMA protocol over the preprogramming TDMA protocol is depended on the scenario. Throughput gain can be very high, when large messages are transmitted in the network with small number of nodes. In contrast, there will be only minor throughput gain or no throughput gain when small messages are transmitted. However, the self-organized TDMA protocol gives some throughput gain over the preprogramming TDMA protocol in most practical scenarios.

The quality of service considered in this thesis can be measured from the message update rate. The self-organized TDMA protocol can adjust a time slot allocation to fulfill the requirement of the message update rate because its data slot assignment algorithm is based on the required message update rate. In contrast, the preprogramming TDMA protocol cannot adjust a time slot allocation to fulfill the required message update rate due to the restriction in the time slot schedule. Therefore, the self-organized TDMA protocol is better than the preprogramming TDMA protocol in term of the quality of service.

Even though the self-organized TDMA protocol has many advantages, it also has some drawbacks. One of the drawbacks of this self-organized TDMA protocol is the large overhead. The self-organized TDMA protocol must include the neighbor information into the MAC header of every transmitted packet. According to the simulations, the MAC header of the self-organized protocol is 84 bytes. This is approximately 26.88 % of a 300-byte time slot, when a 50 Kbits/s data rate is used. In contrast, the MAC header of the preprogramming protocol is 9 bytes which is only 3 % of a 300-byte time slot, when a 50 Kbits/s data rate is used. In the channel with a 50 Kbits/s data rate, the self-organized protocol will use about 13.44 milliseconds to transmit a MAC header, while the preprogramming protocol will only use about 1.44 milliseconds to transmit a MAC header. However, the MAC header designed in the simulation was not optimized due to restrictions in the simulation tool that cannot allow binary value assignment to the header and the header must be fixed size. In the real system, the MAC header could be designed such that it is more optimized than the design in the simulation.
The self-organized TDMA protocol must dedicate some time slots to be used as control slots. Hence, the number of time slots available for data transmissions is reduced. The time frame structure for the self-organized TDMA protocol presented in this thesis contains 10 control slots in each time frame. This is 5% of a 200-slot time frame. When the MAC headers are also considered, the total overhead in each time frame is about 31.6% of a time frame. Because of large MAC headers and control slots, the self-organized protocol cannot utilize all time slots in a time frame. The effect of the overheads in the self-organized TDMA protocol can be reduced by a radio with a higher data rate. The transmission of overhead bits will take shorter time with a higher data rate. If a radio with a 100 Kbits/s data rate is used, the total overhead in each time frame is reduced to 18.3% of a time frame.

The adaptation of time slot schedule according to the network environment is a complex process and takes some time to exchange information. However, the result shows that the average time required for the adaptation processes, i.e. net entry, net leaving, transmission update, and conflict resolve are within acceptable range.

The data slot assignment mechanism in the self-organized TDMA protocol is solely based on the maximum size and the required update rate of the transmitted TDL messages. If the transmitted message size is less than the maximum size, some reserved data slots are unused. Hence, there are some losses in the time slot utilization. The solution is to allocate data lots based on the size of transmitted message. However, message size information must be included in the MAC frame header which makes the overhead larger.

### 7.3 Stability of the Self-Organized TDMA Protocol

The stability of the self-organized TDMA protocol can be considered in terms of the ability to handle a large amount of messages injected into the system, the ability to handle a large amount of packet losses during transmissions, and the ability to resolve a large amount of time slot conflicts.

This self-organized TDMA protocol is designed to work with the TDL application that generates periodic messages at the specified rate. The self-organized TDMA protocol was simulated with the drop-tail queue of 50 packets. This queue will drop new incoming packet from the upper layer, if the queue is full. A packet will be read from the queue according to the order of arrivals by the MAC sublayer, when it has completed transmitting of a previous packet. If the self-organized TDMA protocol cannot find a free data slot to transmit a packet before the queue is full, a newly arrive packet will be dropped at the queue. However, the self-organized TDMA protocol presented in this thesis uses the data slot assignment algorithm such that a node is guaranteed to reserve at least one data slot in every 2 seconds. Therefore, the chance that a packet will be dropped at the queue will be rare given that the queue size has appropriate size (about 50 to 100 packets) and the message injection rate is not too fast. Even though a node may not be able to reserve all data slots required for transmission of a very large message with the
required update rate, it is guaranteed to have some data slots for transmissions. Hence, a node will be able to transmit a very large message with some degradation in the required update rate.

In a real system, there may be a lot of packet losses during transmissions. Because the messages transmitted in the TDL system presented in this thesis are periodic, the packet losses in short period have little effect on the system. Even though some packets may be lost and a message cannot be reassembled at the receiver, the same information can be obtained from other successful reassembled messages that are periodically transmitted. However, the system may not function properly when a node encounters the long period of packet losses such that many packets have been successively lost. A node may be considered as a leaving node, in which it cannot transmit any packet to any neighbor for the long period. However, its information can be transmitted again when it recover from the long period of packet losses.

A packet loss also has effect on the control information. If a control packet is lost during a transmission, the control information required for the self-organized TDMA protocol to work properly will also be lost. The self-organized TDMA protocol presented in this thesis implemented a handshake mechanism when control information is transmitted. The protocol makes use of a request message and an acknowledgement message to implement a handshake mechanism. When a node transmits a request, it will wait for an acknowledgement message for some time period. If a node has not received an acknowledgement message after the waiting period is ended, it will retransmit the new control information.

Time slot conflict can be occurred in the system as mention earlier. When time slot conflict occurs in a time frame, it can be resolved by the end of a time frame and the system can operate without conflict in the next time frame. However, there is the case where time slot conflict cannot be resolved. This happens when all data slots and control slots are conflicted. In this case, a node cannot obtain updated neighbor information from any conflict-free slots; hence time slot conflicts cannot be resolved. However, there is a very low probability that control slots will be in conflicts successively. This is because the winner of a control slot is selected based on the hash values of random seeds that are changed in every new frame. Hence, control slots are prevented from indefinite conflicts. A node can update its neighbor information from a control packet transmitted in any conflict-free control slot, and it can recover from time slot conflicts. As a result, there a very low chance that time slot conflicts will occur indefinitely.

### 7.4 Security of the Self-Organized TDMA Protocol

Security is an important issue in designing a TDL system. A good TDL system should be resistance to any jamming or eavesdropping. Jamming and Eavesdropping can be prevented by using a secure channel. If a secure channel is not available, information may be destroyed or stolen. Jamming may be used to destroy the functionalities of the TDL system by interrupting the control messages transmitted in some control slots. The self-organized TDMA protocol presented in this thesis has the mechanism that prevents the TDL system from being destroyed
by jamming on control slots. A winner of a control slot is selected randomly based on random seeds. It is hard to determine which control slots are transmitting the control information. An enemy may choose to jam on several control slots, and the control information will be destroyed if it is transmitted on any of these jammed control slots. However, the self-organized TDMA protocol presented in this thesis has the mechanism that can retransmit lost control information and it will hardly retransmit on the same control slot. Hence, the control information can be prevented from being destroyed by jamming on specific time slots.

7.5 Weaknesses of the simulation

The simulation was implemented based on many assumptions. Therefore, there are some deviations from a real situation. The obvious weakness of this simulation is that channel noise and error models are neglected. The channel is assumed to be noise-free, error-free, and no fading. Hence, dropped packets due to noise and fading in the channel were ignored. Dropped packets may result in losses of neighbor information and control information, and the performance of the self-organized protocol should be degraded.

The propagation model is determined from the cross over distance between a transmitter and a receiver. If the require transmission range is more than the cross over distance, the two-ray ground model will be applied. Otherwise, the free space model will be applied. The required transmission range is entered into the simulation to calculate the required transmission power according to the appropriate propagation model. Both two-ray ground propagation model and free space propagation model are not suitable for mobile communication. The appropriate model should take fading due to the movements of nodes into account. Fading may result in losses of transmitted packets, and the performance of the self-organized protocol should also be degraded.

The neighbor information part in the MAC header has fixed size regardless of the number of one hop neighbors. When the number of one hop neighbors is less than the maximum number of one hop neighbors that the MAC header can hold, there will be an unnecessary overhead. If the MAC header has variable sizes based on the number of one hop neighbors, the performance of the self-organized protocol should be improved. In addition, the simulation tool does not allow a binary value assignment. The smallest header fields must be assigned with a hexadecimal value. Hence, unnecessary bits are added to the header.

7.4 Alternative Solutions

In addition to the self-organized TDMA protocol proposed in this thesis, there are the alternative solutions that may be used to implement the dynamic MAC protocol. One of the alternative solutions is called the unifying dynamic distributed multichannel TDMA slot assignment (USAP) protocol, which is proposed in [11]. The USAP mechanism is similar to the NAMA and NAPA mechanisms, which only select one node to transmit in a given time slot based on the neighbor information within two hops. The way USAP select transmitting node of a given time
slot is different from NAMA and NAPA. Instead of using hash values to determine a winner, USAP uses the time slot reservation information alternatively broadcasted by each neighbor within two hops to select a free slot. USAP does not need to include seed value in every transmitted packet, but it will need dedicated slots to broadcast time slot reservations. Each node will alternatively broadcast its time slot reservation in these dedicated slots. Hence, the adaptation process will be slower but the overhead is shorter than those of the NAMA and NAPA algorithms. USAP can be implemented as the control slot assignment protocol as an alternative to the NAPA algorithm proposed in this thesis.

Another solution is to use the carrier sense multiple access with collision avoidance (CSMA/CA) mechanism as described in [12]. The CSMA/CA technique can implement a control slot assignment protocol to handle contentions on control slots. A node that has control information to send will sense a channel before transmitting its control information in a given time slot. If some transmission is detected, a node should be backed off for some random period before transmitting. CSMA/CA encounters collisions due to a hidden network area problem. However, it can be solved with a mechanism similar to the IEEE 802.11 RTS/CTS mechanisms [13]. CSMA/CA does not require seed values to be included in every transmitted packet; hence the overhead is smaller. However, CSMA/CA requires longer time slot or faster data rate to sense other transmissions before a time slot is ended. Thus, there is a need to upgrade the radio specification to enable CSMA/CA technique.

### 7.5 Future Works

There are still plenty of rooms to improve the self-organized TDMA protocol proposed in this thesis. There should be a solution to reduce the overhead caused by the MAC frame header and control slots, and the performance of the self-organized TDMA protocol would be improved. One simple solution is to use a radio with a higher data rate. In addition, the alternative solutions presented in previous section may be tried out and compared with the solution proposed in this thesis. Even though the results from the simulation were very satisfied, it had only been tested with the specific scenarios that are commonly operated in the military missions. Various scenarios should be set up to test the performance of the protocol under other circumstances.

The simulation tool used in this thesis does not provide decent physical layers and channel models. The simulation results may be quite different from a real system where a physical interface and a channel model have great effect on a system. In the future, there may be an improved version of NS-2 or other simulation tools that would give more realistic physical interfaces and channel models. Hence, the simulation results should be more realistic.
Chapter 8

Conclusion and Recommendations

This thesis presents the self-organized TDMA protocol that is designed based on the specific military scenarios and the existing self-organized slot assignment algorithms. The purposes of designing and simulating the self-organized TDMA protocol are to assess the advantages of the self-organized TDMA protocol over the preprogramming TDMA protocol, and to encourage a deployment of the self-organized TDMA protocol in a TDL system.

The self-organized TDMA protocol presented in this thesis divides time slots into data slots and control slots. Data slots are used to transmit the TDL messages, while control slots are used to convey the control information required for the adaptation processes. The self-organized TDMA protocol allocates data slots using the data slot assignment algorithm that is based on the message type and the required update rate. The allocation of data slots is dynamically performed, when there are any changes in the network environment; i.e. net entry, net leaving, and transmission update. In order for every node to allocate time slots accordingly, the control information is required to be broadcasted in a control slot. The algorithm that selects a node to transmit in the selected control slot is based on the NAPA algorithm [5]. In addition, the VSLOT algorithm [6] is working together with the NAPA algorithm to perform the selection of a control slot for the net entry process.

The main simulation tool used to test functionalities of the designed self-organized TDMA protocol is NS-2. NS-2 provides good simulation architecture for testing protocol mechanism. However, a real system is quite different from this simulated system. In this simulated system, many assumptions have been made. One major assumption that affects the simulated system differently from a real system is a noise-free and non-fading channel model. The performance of the system should be degraded when noise and fading are introduced into the system. The result of this thesis is the best case result where errors in the channel are ignored.

According to the simulation results, the self-organized TDMA protocol presented in this thesis offers the efficient performance and the flexibility to practical military missions in which mostly operate in the dynamic environment and requires high quality of services. Comparing to the preprogramming TDMA protocol with the same data rate, the self-organized TDMA protocol provides higher performance in terms of throughput, time slot utilization, and message update rate in practical military scenarios. The maximum achievable system throughput of the self-organized TDMA protocol is independent of the number of nodes in the net, because of its ability to reserve any free data slots. In contrast, the maximum achievable system throughput of the preprogramming TDMA protocol is directly proportional to the number of nodes in the net. The preprogramming TDMA protocol can perform better than the self-organized TDMA protocol in term of the maximum achievable system throughput, when the number of nodes in the network is approaching the maximum number of nodes for which the pre programming
TDMA time slot schedule is designed. However, the scenarios, in which the preprogramming TDMA protocol is better than the self-organized TDMA protocol in term of the maximum achievable system throughput, are rare in practical airborne missions.

The adaptability of the self-organized TDMA protocol is measured from its processing time when performing the net entry, net leaving, transmission update, and time slot conflict resolve. The result from the simulation indicates that this self-organized TDMA protocol offers satisfied adaptability. The average processing times of each function are quite low and modestly influenced by the network environment, i.e. the number of existing nodes in the net.

A TDL system requires a stable MAC protocol such that it can operate in most practical scenarios or in the extreme scenario where a TDL system is still able to operate under acceptable degradations. The results and discussions from this thesis illustrates that the TDL system with the presented self-organized TDMA protocol is able to operate in the practical scenarios and in the extreme environment with some degradations. In addition to the performance and stability, security is an important issue in designing the TDL system. The discussion from this thesis shows that the presented self-organized TDMA protocol offers some level of security, which prevents the system from being crashed by intentionally jamming.

Even though the self-organized TDMA protocol offers higher performance and flexibility, it has some drawbacks. The main drawback is that the self-organized TDMA protocol contains a large overhead from the MAC header and control slots. Another drawback is that the adapting processes requires some processing time.

The results from this thesis present many advantages of the self-organized TDMA protocol over the preprogramming TDMA protocol in the practical military scenarios. It is also worth to note that the performance of the self-organized TDMA protocol may be different in different scenarios based on the size of transmitted messages and the number of nodes in the network. Therefore, scenarios that a TDL system is operating must be taken into account when designing a self-organized TDMA protocol. In addition, the performance of the self-organized TDMA protocol is limited by the radio specification. Hence, the available hardware specification must also be taken into account when designing a protocol. According to the results, the presented self-organized TDMA protocol may be deployed into a real TDL system to provide better performance and flexibility. However, this protocol still needs some improvements and more realistic simulations before implementing into a real working system. In addition, the alternative self-organized TDMA protocols should be tried out to find the best solution that suite the specific needs in a TDL system.
References


Appendix A

Basic Algorithms

A.1 Background of NAMA Algorithm

Node Activation Multiple Access (NAMA) algorithm is an algorithm that selects only one member within two-hop distance to transmit in given time slot. The NAMA algorithm calculates hash values of each node within two-hop distance. The NAMA algorithm’s eliminates a hidden terminal problem and ensures that all nodes within one-hop distance of a transmitter will receive data without any collision. The NAMA algorithm makes use of random seeds which are exchanged within network’s participating nodes and a time slot ID to compute a hash value. A node with the highest hash value is the winner of a time slot and has the first opportunity to transmit information in that time slot as illustrates in Figure A-1.

![Figure A-1: NAMA scheduling algorithm](image)

In the first time slot, Node B has the highest hash value, thus it can transmit information in this slot. This is the same mechanism for the second, third, and fourth slots.
A.2 Background of NAPA Algorithm

Node Activation Polling Access (NAPA) algorithm is an algorithm which determines a time slot’s winner similar to the NAMA algorithm, but the NAPA algorithm also allows the other node to transmit data if a winning node has nothing to transmit. When a winning node has no packet to transmit, it will poll one or multiple one-hop neighbors if they are willing to transmit. A neighbor who receives a short polling packet, will determines if it can transmit based on its entity in the poll list.

The NAPA operation is illustrates as Figure A-2. At the beginning of each time slot, each node runs the node activation algorithm to select a winner of a time slot. If the winner has packet to transmit, it will transmit data. Otherwise, it will select one or more one-hop neighbor to poll by transmitting a short polling packet which contains the list of polled nodes. A node which is not a time slot’s winner will listen for polling packet. When a polling packet is received correctly, it will determine if it can transmit a data packet in response to the poll. A polled node which has no data packet to send will remain quiet for the remaining of a time slot. A polled node, in which has data packet to transmit and is in the poll list, will listen to channel for some back-off time period that proportional to its position in the poll list. If the polled node detects carrier before the back-off time period is expired, it will remain quiet. Otherwise, it will transmit a packet.

A.3 Background of VSLOT Algorithm

Virtual Slot (VSLOT) Scheduling algorithm is an algorithm for minimizing a contention, which allows only nodes that do not share a consistent topology view to contend a time slot. VSLOT uses a notion of virtual nodes which are a set of random seeds that are pre-allocated and known to every node in a network for its scheduling approach. Each virtual node will be assigned with random seed and each node will treat as if there is another node in a network. In each time slot, a winner of time slot is determined similar to the NAMA and NAPA algorithms. When a virtual slot wins a time slot, a time slot simply goes unused due to no real node will transmit during the slot. Example of the VSLOT scheduling is illustrated in Figure A-3. As an example, Virtual Node V is known to every node in the same network. A hash value of node V is computed by using a seed known to every node in the network. In the fourth time slot, Virtual Node V has the highest hash value. Therefore, all nodes in the network will be in the receiving mode. Thus, the fourth time slot can be referred as a virtual slot. Any nodes which do not share the same topology as the existing nodes in the network can transmit their control information in this time slot. The winning chance of virtual nodes can be controlled by varying the number of virtual nodes.
Figure A-2: Operation of NAPA
Figure A-3: NAMA with VSLOT scheduling
Appendix B

Program Listings

B.1 C++ classes

B.1.1 TDL message generator class

Full class descriptions of TDL message generator class are provided in the following URLs.
http://code.google.com/p/ns2-self-organized-tdma/source/browse/tdl/tdl_data_msg.h
http://code.google.com/p/ns2-self-organized-tdma/source/browse/tdl/tdl_data_msg.cc

B.1.2 Modified UDP transport class

Full class descriptions of modified UDP class are provided in the following URLs.
http://code.google.com/p/ns2-self-organized-tdma/source/browse/tdl/tdl_data_udp.h
http://code.google.com/p/ns2-self-organized-tdma/source/browse/tdl/tdl_data_udp.cc

B.1.3 Self-Organized TDMA MAC class

Full class descriptions of self-organized TDMA class are provided in the following URLs.
http://code.google.com/p/ns2-self-organized-tdma/source/browse/tdl/tdl_dynamic_tdma_2.h
http://code.google.com/p/ns2-self-organized-tdma/source/browse/tdl/tdl_dynamic_tdma_2.cc

B.1.4 Preprogramming TDMA MAC class

Full class descriptions of preprogramming TDMA class are provided in the following URLs.
http://code.google.com/p/ns2-self-organized-tdma/source/browse/tdl/tdl_fixed_tdma.h
http://code.google.com/p/ns2-self-organized-tdma/source/browse/tdl/tdl_fixed_tdma.cc

B.1.5 Preprogramming TDMA’s time slots configuration file

A sample of time slots configuration file for the preprogramming TDMA protocol is provided in the following URL.
http://code.google.com/p/ns2-self-organized-tdma/source/browse/sims/tdma_table

In this configuration file, the first element in each row indicates a node’s ID and other elements in the same row indicate slot number reserved by a node.
B.2 Tcl Test Programs

B.2.1 Net Entry Test Programs

The source codes for testing net leaving are provided in the following URLs.

http://code.google.com/p/ns2-self-organized-tdma/source/browse/sims/sim_netEntry.tcl
http://code.google.com/p/ns2-self-organized-tdma/source/browse/sims/sim_netEntry_sc2.tcl

B.2.2 Net Leaving Test Program

The source code for testing net leaving is provided in the following URLs.

http://code.google.com/p/ns2-self-organized-tdma/source/browse/sims/sim_netleaving.tcl

B.2.3 Transmission Update Test Programs

The source codes for testing transmission update function are provided in the following URLs.

http://code.google.com/p/ns2-self-organized-tdma/source/browse/sims/sim_msgupdate.tcl
http://code.google.com/p/ns2-self-organized-tdma/source/browse/sims/sim_msgupdate_sc2.tcl

B.2.4 Conflict Resolve Test Programs

The source codes for testing time slot conflict resolve function are provided in the following URLs.

http://code.google.com/p/ns2-self-organized-tdma/source/browse/sims/sim_conflict_sc2.tcl

B.2.5 Tests Programs for the study scenarios

The source codes for testing two study scenarios with the self-organized TDMA protocol with a 50 Kbits/s, the preprogramming TDMA protocol with a 50 Kbits/s data rate, and the self-organized TDMA protocol with a 100 Kbits/s are provided in the following URLs.

http://code.google.com/p/ns2-self-organized-tdma/source/browse/sims/sim_studyscene_dyn.tcl
http://code.google.com/p/ns2-self-organized-tdma/source/browse/sims/sim_studyscene_alt.tcl
http://code.google.com/p/ns2-self-organized-tdma/source/browse/sims/sim_studyscene2_dyn.tcl
http://code.google.com/p/ns2-self-organized-tdma/source/browse/sims/sim_studyscene2_alt.tcl