A ROBUST WIRELESS MULTICAST PROTOCOL

A thesis presented to

the faculty of

the College of Engineering and Technology of Ohio University

In partial fulfillment

of the requirements for the degree

Master of Science

Vamshi Krishna Boinpalli

June 2005
This thesis entitled

A ROBUST WIRELESS MULTICAST PROTOCOL

by

VAMSHI KRISHNA BOINPALLI

has been approved for

the School of Electrical Engineering and Computer Science

and the Russ College of Engineering and Technology by

Carl T. Bruggeman

Assistant Professor of Computer Science

Dennis Irwin

Dean, Russ College of Engineering and Technology
The RoboCup Unicast Protocol and the RoboCup Multicast Protocol are two Real-time Application layer protocols used by Ohio University at the RoboCup International soccer competition played between autonomous mobile Robots. The RoboCup Unicast Protocol was used to transmit strategic information from the Base Station to each Robot individually over a designated wireless medium at regular intervals. It was replaced the following year by the RoboCup Multicast Protocol, which transmitted the same information to a single multicast group address, and thus reduced the total wireless network overhead. Both protocols worked well in the lab but failed in competition. This thesis delves into the reasons for the unexpected degradation in performance of both of the protocols in competition and describes a new Robust Wireless Multicast Protocol (RWMP) to overcome these problems. The RWMP protocol reduces the network overhead when compared to the RoboCup Unicast Protocol and also ensures reliability of transmission, unlike the RoboCup Multicast Protocol. The performance of the Robust Wireless Multicast Protocol, compared to the other two protocols, is validated by conducting a detailed experimental analysis of the three protocols under various wireless network loads and comparing the results in terms of total losses observed and time taken for transmission.
ACKNOWLEDGEMENTS

I am greatly indebted to Dr. Carl Bruggeman for his continuous support, advice, feedback and constructive criticism. This thesis work would not have been successfully completed without his ideas and thoughts.

I thank Dr. Shawn Ostermann for allowing me to use the resources at IRG lab. It's been a privilege to take his networking courses and he is an inspiration for my networking career. I would like to thank Dr. David Chelberg for allowing me to use the RoboCup lab for my experiments. I thank Dr. Hans Kruse, my external committee member, for his suggestions on implementing the queueing estimation. I would also like to thank Mr. Ankush Sawant for his feedback on my thesis writeup and Mr. Manikantan Ramadas, Mr. Joshua Blanton and Mr. Mark Goldman for their help and suggestions.

I would like to thank my parents, my sister Vani and my friends for all their love and affection, for being there with me whenever I was in need. Special thanks to my roommates at Ohio University for all the fun and enjoyment that I had with them for the past 2 years, and for cheering me up during tough times. I finally thank my parents and my dolls for being the inspiration and the driving force behind my success.

I dedicate this thesis and my Masters degree to my parents.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABSTRACT</td>
<td>3</td>
</tr>
<tr>
<td>LIST OF TABLES</td>
<td>8</td>
</tr>
<tr>
<td>LIST OF FIGURES</td>
<td>11</td>
</tr>
<tr>
<td>1 Introduction</td>
<td>14</td>
</tr>
<tr>
<td>1.1 Organization Of Thesis</td>
<td>17</td>
</tr>
<tr>
<td>2 Background</td>
<td>18</td>
</tr>
<tr>
<td>2.1 IEEE 802.11</td>
<td>18</td>
</tr>
<tr>
<td>2.2 IEEE 802.11 Architecture</td>
<td>18</td>
</tr>
<tr>
<td>2.2.1 802.11 Components</td>
<td>18</td>
</tr>
<tr>
<td>2.3 Physical Layer Specifications</td>
<td>21</td>
</tr>
<tr>
<td>2.4 MAC Protocols In 802.11</td>
<td>23</td>
</tr>
<tr>
<td>2.4.1 MAC Service Data Unit Frames</td>
<td>24</td>
</tr>
<tr>
<td>2.4.2 Difference Between Wired And Wireless LAN</td>
<td>25</td>
</tr>
<tr>
<td>2.4.3 Hidden Node Problem</td>
<td>26</td>
</tr>
<tr>
<td>2.4.4 Carrier Sense Multiple Access With Collision Avoidance</td>
<td>29</td>
</tr>
<tr>
<td>2.4.5 Distributed Coordination Function</td>
<td>30</td>
</tr>
<tr>
<td>2.4.6 Acknowledgement Procedure</td>
<td>32</td>
</tr>
<tr>
<td>3 Robust Wireless Multicast Protocol For RoboCup</td>
<td>33</td>
</tr>
<tr>
<td>3.1 RoboCup Introduction</td>
<td>33</td>
</tr>
<tr>
<td>3.2 OU RoboCup Application layer Protocol</td>
<td>34</td>
</tr>
<tr>
<td>3.2.1 The RoboCup Unicast Protocol</td>
<td>35</td>
</tr>
<tr>
<td>3.2.2 Problems With The RoboCup Unicast Protocol</td>
<td>37</td>
</tr>
</tbody>
</table>
3.2.3 The RoboCup Multicast Protocol .................................. 38
3.2.4 Problem With The RoboCup Multicast Protocol ................. 39
3.2.5 The Robust Wireless Multicast Protocol ........................... 41
3.2.6 Advantages Of The Robust Wireless Multicast Protocol ...... 43
3.2.7 Implementation Details Of The Robust Wireless Multicast Pro-
tocol ................................................................. 46

4 Experimental Results ....................................................... 49
   4.1 TestTCP Modification .................................................. 50
   4.2 Equipment Used For Experiments ..................................... 52
   4.3 Experiments Conducted .................................................. 54
       4.3.1 Experiment Setup For The RoboCup Unicast Protocol ....... 55
       4.3.2 Experiment Setup For The RoboCup Multicast Protocol .... 56
       4.3.3 Experiment Setup For The Robust Wireless Multicast Protocol 56
   4.4 Behavior Of Protocols In Absence Of ADDED_TRAFFIC ........ 57
   4.5 Performance Of The Protocols In Presence Of ADDED_TRAFFIC Generated By A Single Additional Wireless Network ......... 63
       4.5.1 Access Points A and B On The Same Channel ................. 64
       4.5.2 Access Points Placed On Adjacent Channels ................ 73
       4.5.3 Access Points A And B Placed One Channel Apart .......... 75
   4.6 Performance Of The Protocols In Presence Of ADDED_TRAFFIC Generated By Two Additional Wireless Networks ............... 79
       4.6.1 All Three APs On The Same Channel ......................... 81
       4.6.2 Access Points B And C Placed On Adjacent Channel ....... 82
       4.6.3 APs B And C Are Placed On Channels 1 And 3 Respectively 85
   4.7 Queue Model Simulation ............................................... 88
   4.8 Experiment Spanning Over Longer Time ............................ 94

5 Conclusion ................................................................. 98
   5.1 Summary ............................................................... 98
   5.2 Advantages And Disadvantages ...................................... 99
5.3 Future Work .................................................. 100
BIBLIOGRAPHY ............................................... 102

APPENDIX

A Results When Access Points A And B Are Placed 2 Channels Apart .... 105
TABLE OF CONTENTS

LIST OF TABLES

Table | Page
--- | ---
2.1 Comparing IEEE 802.11a, b And g Standards | 22
4.1 TOTAL LOSSES Observed, TTCP RATE=40, AP A On Channel 2 | 59
4.2 TOTAL LOSSES Observed, TTCP RATE=60, AP A On Channel 2 | 59
4.3 TOTAL LOSSES Observed, TTCP RATE=80, AP A On Channel 2 | 60
4.4 TOTAL LOSSES Observed, TTCP RATE=100, AP A On Channel 2 | 60
4.5 TOTAL LOSSES Observed, TTCP RATE=40, APs A, B On Channel 2 | 66
4.6 TOTAL LOSSES Observed, TTCP RATE=60, APs A, B On Channel 2 | 68
4.7 TOTAL LOSSES Observed, TTCP RATE=80, APs A, B On Channel 2 | 69
4.8 TOTAL LOSSES Observed, TTCP RATE=100, APs A, B On Channel 2 | 69
4.9 TOTAL LOSSES Observed, TTCP RATE=40, APs A, B On Channels 2 And 3 Respectively | 74
4.10 TOTAL LOSSES Observed, TTCP RATE=60, APs A, B On Channels 2 And 3 Respectively | 74
4.11 TOTAL LOSSES Observed, TTCP RATE=80, APs A, B On Channels 2 And 3 Respectively | 75
4.12 TOTAL LOSSES Observed, TTCP_RATE=100, APs A, B On Channels 2 And 3 Respectively

4.13 TOTAL LOSSES Observed, TTCP_RATE=40, APs A And B On Channels 2 And 4 Respectively

4.14 TOTAL LOSSES Observed, TTCP_RATE=60, APs A And B On Channels 2 And 4 Respectively

4.15 TOTAL LOSSES Observed, TTCP_RATE=80, APs A And B On Channels 2 And 4 Respectively

4.16 TOTAL LOSSES Observed, TTCP_RATE=100, APs A And B On Channels 2 And 4 Respectively

4.17 TOTAL LOSSES Observed, TTCP_RATE=40, APs A, B And C On Channel 2

4.18 TOTAL LOSSES Observed, TTCP_RATE=60, APs A, B And C On Channel 2

4.19 TOTAL LOSSES Observed, TTCP_RATE=80, APs A, B And C On Channel 2

4.20 TOTAL LOSSES Observed, TTCP_RATE=100, APs A, B And C On Channel 2

4.21 TOTAL LOSSES Observed, TTCP_RATE=40, AP A On Channel 2, B And C On Channel 3

4.22 TOTAL LOSSES Observed, TTCP_RATE=60, AP A On Channel 2, B And C On Channel 3

4.23 TOTAL LOSSES Observed, TTCP_RATE=40, APs A, B And C On Channels 2, 1 And 3 Respectively

4.24 TOTAL LOSSES Observed, TTCP_RATE=60, APs A, B And C On Channels 2, 1 And 3 Respectively
4.25 TOTAL LOSSES Observed On CAPTAIN, TTCP RATE=100, APs A, B On Channels 2,3 Respectively .................................................. 95

4.26 TOTAL LOSSES Observed On Non-CAPTAIN, TTCP RATE=100, APs A, B On Channels 2,3 Respectively ................................. 96

A.1 TOTAL LOSSES Observed, TTCP RATE=40, APs A And B On Channels 2 And 5 respectively .................................................. 105

A.2 TOTAL LOSSES Observed, TTCP RATE=60, APs A And B On Channels 2 And 5 Respectively .................................................. 106

A.3 TOTAL LOSSES Observed, TTCP RATE=80, APs A And B On Channels 2 And 5 Respectively .................................................. 106

A.4 TOTAL LOSSES Observed, TTCP RATE=100, APs A And B On Channels 2 And 5 Respectively .................................................. 107
### LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>IEEE 802 Standard Overview And Architecture</td>
<td>19</td>
</tr>
<tr>
<td>2.2</td>
<td>Typical IEEE 802.11 Network Infrastructure</td>
<td>20</td>
</tr>
<tr>
<td>2.3</td>
<td>MAC Architecture</td>
<td>24</td>
</tr>
<tr>
<td>2.4</td>
<td>Standard IEEE 802.11 Frame Format</td>
<td>25</td>
</tr>
<tr>
<td>2.5</td>
<td>Hidden Node Problem</td>
<td>27</td>
</tr>
<tr>
<td>3.1</td>
<td>The RoboCup Unicast Protocol</td>
<td>36</td>
</tr>
<tr>
<td>3.2</td>
<td>The RoboCup Multicast Protocol</td>
<td>40</td>
</tr>
<tr>
<td>3.3</td>
<td>The Robust Wireless Multicast Protocol</td>
<td>43</td>
</tr>
<tr>
<td>3.4</td>
<td>The Robust Wireless Multicast Protocol Flowchart</td>
<td>47</td>
</tr>
<tr>
<td>4.1</td>
<td>Method Of Packet Transmission</td>
<td>53</td>
</tr>
<tr>
<td>4.2</td>
<td>Experiment Setup With Single AP</td>
<td>58</td>
</tr>
<tr>
<td>4.3</td>
<td>Comparing ELAPSED.TIME, TTCP_RATE=100, AP A On Channel 2</td>
<td>61</td>
</tr>
<tr>
<td>4.4</td>
<td>Experiment Setup With Two APs</td>
<td>65</td>
</tr>
<tr>
<td>4.5</td>
<td>Percentage Of Losses Observed, TTCP_RATE=40; APs A, B On Channel 2</td>
<td>67</td>
</tr>
</tbody>
</table>
4.6 ELAPSED\_TIME Observed, TTCP\_RATE=40, APs A And B On Channel 2

4.7 Average ELAPSED\_TIME Observed, TTCP\_RATE=100, APs A And B On Channel 2

4.8 Percentage Of Losses Observed, TTCP\_RATE=100, APs A And B On Channel 2

4.9 Percentage Of Losses Observed For The RoboCup Unicast, TTCP\_RATE=100, APs A And B On Channel 2

4.10 Average ELAPSED\_TIME Observed, TTCP\_RATE=100, APs A And B On Channels 2 And 3 Respectively

4.11 Percentage Of Losses Observed, TTCP\_RATE=100, APs A And B on Channels 2 And 3 Respectively

4.12 Experiment Setup With Three APs

4.13 ELAPSED\_TIME Observed, TTCP\_RATE=100, APs A, B And C On Channel 2

4.14 Percentage Of Losses Observed, TTCP\_RATE=100, APs A, B And C on Channel 2

4.15 Average ELAPSED\_TIME Observed, TTCP\_RATE=40, AP A On Channel 2, B And C On Channel 3

4.16 Percentage of Losses Observed, TTCP\_RATE=40; AP A On Channel 2, B And C On Channel 3

4.17 Average ELAPSED\_TIME Observed, TTCP\_RATE=60, AP A On Channel 2, B And C On Channel 3

4.18 Percentage of Losses Observed, TTCP\_RATE=60, AP A On Channel 2, B And C On Channel 3
4.19 Average ELAPSED_TIME Observed, TTCP_RATE=40, APs A, B And C On Channels 2, 1 And 3 Respectively ........................................... 90

4.20 Percentage of Losses observed, TTCP_RATE=40, APs A, B And C on Channels 2, 1 And 3 Respectively ........................................... 91

4.21 Average ELAPSED.TIME Observed, TTCP_RATE=60, APs A, B And C On Channels 2, 1 And 3 Respectively ................................. 92

4.22 Percentage of Losses Observed, TTCP_RATE=60, APs A, B And C On Channels 2, 1 And 3 Respectively ................................. 93

4.23 Estimated ELAPSED_TIME, TTCP_RATE=100, Loss Probability = 30% 94

4.24 Average Percentage Of Losses Observed On CAPTAIN, TTCP_RATE=100, APs A On Channel 2 And B On Channel 3 ................. 96

4.25 Average Percentage Of Losses Observed On Non-CAPTAIN, TTCP_RATE=100, APs A on Channel 2 And B On Channel 3 ................. 97
1. Introduction

RoboCup is an international soccer competition played among teams consisting of autonomous mobile robots [6]. It is an international joint project to motivate research in inter-disciplinary fields including Artificial Intelligence, Robotics, Wireless technology and various Electrical Engineering and Mechanical Engineering fields [13], [11]. One of the goals of the project is to identify general problems that may be encountered when incorporating multiple standards and technologies into a single real time application. RoboCup technology is an attempt to develop innovations that can be used to solve real world problems, such as search and rescue operations in disasters like floods and earthquakes.

The RoboCup soccer competition is divided into five different league games. They are the simulation league, the small size league, the middle size league, the 4-legged league and the humanoid league. The Ohio University (OU) RoboCup team participates in the small size league competition that the international organization conducts [4]. The small size league game is played between two teams. Each team is comprised of five autonomous mobile robots. Apart from the robots, each team has an off field computer, referred to as Base Station (BS), used to compute and transmit strategic information to the robots. To facilitate the smooth running of the game, the rules of the competition require that the communication medium between the robots and the BS be wireless. Each team should be able to operate in at least two Wireless Local Area Networks (WLANs), or two channels on a single WLAN, so that any two teams can compete without using the same channel or frequency. A
WLAN, mentioned above, can be defined as a group of wireless devices which can communicate with each other using one of several standard procedures.

The small size league game is played in a small field. The size of the field is 3 meters wide and 6 meters long with a goal in the center on either ends. Each team must analyze the game situation using a camera placed on top of the field, and an optional camera on top of the goal for each team. The teams then use this information to develop game strategy and transmit data to the robots from the BS using their WLAN. There is no restriction whatsoever on the amount of data that the teams can transmit over the WLAN.

Different teams often use different WLAN standards to communicate with their respective robots. WLANs observed at the competition have operated at several frequency spectrums such as 400MHz, 900MHz, 2.4GHz, 5GHz and infrared. WLANs operating at the same frequency range affect the performance of the others. Most of the teams participating at the RoboCup were reported to be operating in the 2.4GHz frequency range. The OU RoboCup team uses the 802.11b standard proposed by the Institute of Electrical and Electronics Engineers (IEEE), which also operates in the 2.4GHz frequency range. This thesis is an analysis of the behavior of the 802.11b standard and the effects observed when operating with WLANs operating in the same 2.4GHz frequency spectrum.

The OU RoboCup team first implemented a simple protocol, referred to as the RoboCup Unicast Protocol, in which the BS transmits information to all the robots individually. This protocol worked well in the lab but yielded poor results in competition. It was believed that the degradation in performance of the protocol in competition was due to the over utilization of the WLAN, since much of the data being transmitted was redundant.

A second protocol, called the RoboCup Multicast Protocol, was developed to resolve the performance problem that the RoboCup Unicast Protocol faced in competition. In this protocol, the BS transmits all the information to a single multicast address
group [14]. Each robot listens to this multicast address. Thus, the RoboCup Multicast Protocol reduced the overhead on the WLAN by reducing the amount of data transmitted by almost five times. It was hoped that this new protocol would solve the problems that were experienced by the RoboCup Unicast Protocol in competition. Inspite of the reduced bandwidth utilization, the RoboCup Multicast Protocol did not perform up to the expectations.

This thesis proposes another solution for the above problems by carefully analyzing the 802.11b WLAN standard and the network environment prevailing at the competition. A new protocol called Robust Wireless Multicast Protocol is proposed and the performance of all the three protocols are compared under network conditions similar to that observed at the competition. In the new protocol, the BS transmits the information as a single unicast packet to a single Robot identified as the CAPTAIN. All the other robots capture the information over the WLAN indirectly using a sniffing technique explained in later chapters.

The advantages of the Robust Wireless Multicast Protocol over the previous two protocols is analyzed in terms of the 'percentage of losses observed' and 'time taken to transmit data'. The transmission rate of the BS and the interference generated by additional WLANs (similar to that observed at the competition) is varied for each experiment. The results are then presented and compared in terms of two factors. The first one is TOTAL LOSSES, which is the total losses observed. And the second factor is ELAPSED TIME, which is the total time taken for transmitting the data from the BS to the robots. Overall, under heavy network load, RWMP experienced negligible packet losses of approximately 0% to 2% when compared to 10% to 90% for the RoboCup Unicast Protocol and 15% to 25% for the RoboCup Multicast Protocol. The RoboCup Multicast Protocol experienced the lowest ELAPSED TIME when compared to the other two protocols. The RoboCup Unicast Protocol, on an average, observed an ELAPSED TIME of approximately 3 seconds more than the RoboCup Multicast Protocol. The RWMP observed an ELAPSED TIME of few milliseconds
more than the RoboCup Multicast Protocol which is negligible. Thus, the RWMP is better than both the RoboCup Unicast Protocol and the RoboCup Multicast Protocol given its low TOTAL LOSSES.

1.1 Organization Of Thesis

In chapter 2, a brief overview of the IEEE 802.11 architecture is given, followed by a detailed study of the underlying protocols used in the standard. Chapter 3 provides a detailed description of the RoboCup Unicast Protocol and the RoboCup Multicast Protocol, and an analysis of why they failed at the competition. It is then followed up by the proposal of the new protocol, the Robust Wireless Multicast Protocol, with a detailed description of its expected advantages over the other two protocols in competition. Chapter 4 then gives a detailed description of the various experiments conducted and the results observed. Chapter 5 then gives a brief summary of the new protocol and its advantages, followed up with some obvious directions for future work.
2. Background

In this chapter the Wireless Local Area Networks (WLANs) and the Institute of Electrical and Electronics Engineers (IEEE) standard for WLANs are discussed. Section 2.1 outlines the WLAN protocol. Section 2.2 describes the WLAN Architecture. Section 2.3 and Section 2.4 specify the various Physical Layer Specifications and the Medium Access Control (MAC) Layer Specifications of the proposed IEEE standard for WLANs respectively.

2.1 IEEE 802.11

IEEE 802.11 is the WLAN standard proposed by the IEEE organization. It specifies the “Physical layer and Medium Access Control (MAC) sublayer of the Open Systems Interconnection (OSI) seven-layer reference model” [15]. As shown in Figure 2.1, IEEE 802.11 protocol resides at the MAC sublayer of the Data Link layer and also on the Physical layer, similar to all the IEEE 802 LAN standards like Ethernet, Token Ring, FDDI and ATM.

2.2 IEEE 802.11 Architecture

Figure 2.2 depicts a typical network infrastructure which contains wireless networks along with traditional wired networks [10].

2.2.1 802.11 Components

Following are some of the commonly identified components in a WLAN [2]:

- **WLAN Station (STA)** is the basic component of 802.11 wireless network. A STA can be any 802.11 device which implements the 802.11 protocol and has some medium of connection to the wireless network. A STA could be either a
### IEEE 802 Standard Overview And Architecture

IEEE 802 standards define the MAC layer technologies and underlying Physical medium access for seven different LAN and MAN environments [16].

#### Basic Service Set (BSS)

A Basic Service Set (BSS) is another fundamental building block of the IEEE 802.11 architecture. A BSS is a group of STAs that is controlled by a Coordination Function (CF). Basic Service Area (BSA) is the area within which the STAs of a BSS communicate. Section 2.4.5 describes the basic Coordination Function which is implemented by all 802.11 devices. In Figure 2.2, a single “dashed circle” denotes a BSS. The area encircled by these circles denote the BSA of the corresponding BSS.

#### Extended Service Set (ESS)

Extended Service Set (ESS) is one or more BSSs interconnected. The ESS appears as a single BSS to the Logical Link Control layer of any STA within that ESS. Extended Service Area (ESA) is the area of communication associated

---

<table>
<thead>
<tr>
<th>Layer</th>
<th>802.2 LOGICAL LINK CONTROL</th>
<th>802.1 BRIDGING</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>802.3 MEDIUM ACCESS</td>
<td>802.12 MEDIUM ACCESS</td>
</tr>
<tr>
<td></td>
<td>802.4 MEDIUM ACCESS</td>
<td>802.11 MEDIUM ACCESS</td>
</tr>
<tr>
<td></td>
<td>802.5 MEDIUM ACCESS</td>
<td>802.9 MEDIUM ACCESS</td>
</tr>
<tr>
<td></td>
<td>802.6 MEDIUM ACCESS</td>
<td>802.11 MEDIUM ACCESS</td>
</tr>
<tr>
<td></td>
<td>802.7 MEDIUM ACCESS</td>
<td>802.12 MEDIUM ACCESS</td>
</tr>
<tr>
<td></td>
<td>802.8 MEDIUM ACCESS</td>
<td>802.12 MEDIUM ACCESS</td>
</tr>
<tr>
<td></td>
<td>802.9 MEDIUM ACCESS</td>
<td>802.12 MEDIUM ACCESS</td>
</tr>
<tr>
<td></td>
<td>802.10 MEDIUM ACCESS</td>
<td>802.12 MEDIUM ACCESS</td>
</tr>
<tr>
<td></td>
<td>802.11 MEDIUM ACCESS</td>
<td>802.12 MEDIUM ACCESS</td>
</tr>
<tr>
<td></td>
<td>802.12 MEDIUM ACCESS</td>
<td>802.12 MEDIUM ACCESS</td>
</tr>
</tbody>
</table>

---

Laptop, a PC with a PCMCIA wireless card [5], or a hand held wireless device like a PDA. In Figure 2.2, $STA_1$, $STA_2$, … $STA_7$ are all STAs.
Figure 2.2. Typical IEEE 802.11 Network Infrastructure: In the above figure, the Access Points AP1 and AP2 also act as Portals. STA denote wireless stations with ESS. The solid circle in Figure 2.2 denotes an ESS. The area encircled by this ESS denotes the corresponding ESA.

- **Access Point (AP)**, a component within BSS, acts as a communication medium between STAs within a particular BSS. It also provides connectivity to the Distribution System, which is another major component of a WLAN. AP₁ and AP₂ in Figure 2.2 denote two APs corresponding to two different BSSs.

- **Distribution System (DS)**, described by the IEEE 802.11, is a service which provides connectivity between APs of different BSSs and also helps in mobility of STAs between different BSSs. It can be a wired or wireless network. IEEE 802.11 does not specify any standards on how the DS is to be implemented.
• **PORTAL** is the “logical point” where data frames from a non IEEE 802.11 LAN enter the DS of an ESS. An AP can also work as a PORTAL. In Figure 2.2, both the APs also work as PORTALs.

As depicted in Figure 2.2, BSSs are classified into two types as follows:

1. **Independent Basic Service Set** is the fundamental building block of IEEE 802.11 architecture. An Independent BSS, also called an *ad hoc network*, consists of STAs which can communicate directly with each other.

2. **Infrastructure Basic Service Set** is a BSS where STAs communicate with each other via an AP. The network setup for RoboCup consists of a single Infrastructure BSS corresponding to every single team.

This thesis concentrates on the basic properties and functions of the Physical layer and MAC sublayer of the 802.11 network. Also, the thesis mainly concentrates on those aspects of 802.11 which affect the working of the proposed Application layer protocol and its predecessors. This thesis does not explain the frame format of 802.11 in detail, nor does it give any details about the various types of frame types that are provided by 802.11. Section 2.3 explains the physical layer specifications of 802.11 and Section 2.4 elaborates on the MAC layer protocols defined by the 802.11 standard.

### 2.3 Physical Layer Specifications

The IEEE has proposed three different specifications for the 802.11 standard. These standards mainly differ in terms of their physical layer specifications which are given in Table 2.1. The IEEE 802.11b standard is used by the OU RoboCup team and also most of the other playing teams at the competition. Therefore, this thesis concentrates on the study of the 802.11b standard and its effects at the competition. Following is a brief explanation of the 802.11b standard.

The 802.11b operates in the 2.4 to 2.4835 GHz frequency range. This range is divided into 11 different channels numbered from channel 1 to channel 11. Each
Table 2.1 Comparing IEEE 802.11a, b And g Standards

<table>
<thead>
<tr>
<th></th>
<th>802.11a</th>
<th>802.11b</th>
<th>802.11g</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frequency Range</td>
<td>5.15 - 5.35 GHz and 5.75 - 5.825 GHz</td>
<td>2.4 - 2.4835 GHz</td>
<td>2.4 - 2.4835 GHz</td>
</tr>
<tr>
<td>Modulation Technique</td>
<td>OFDM (Orthogonal Frequency Division Multiplexing)</td>
<td>DSSS (Direct Sequence Spread Spectrum)</td>
<td>FHSS (Frequency Hopping Spread Spectrum)</td>
</tr>
<tr>
<td>Number of Channels</td>
<td>12</td>
<td>11</td>
<td>11</td>
</tr>
<tr>
<td>Data Rates Available</td>
<td>6, 9, 12, 18, 24, 36, 48, 54 Mbps</td>
<td>1, 2, 5.5, 11 Mbps</td>
<td>6, 9, 12, 18, 24, 36, 48, 54 Mbps</td>
</tr>
<tr>
<td>Maximum Channels Allowed</td>
<td>12</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>Channel Bandwidth</td>
<td>20 MHz</td>
<td>5 MHz</td>
<td>5 MHz</td>
</tr>
</tbody>
</table>

channel has a bandwidth of 5 MHz. The center frequencies of these 11 channels are 2.412, 2.417, 2.422, 2.427, 2.432, 2.437, 2.442, 2.447, 2.452, 2.457 and 2.462 GHz. Countries in the UK allow 2 more channels with center frequencies 2.467 and 2.472 GHz. Adjacent channels operate at overlapping frequencies and hence using them would result in degradation of performance. The IEEE 802.11 requires a minimum 25 MHz spacing between active channels. This leaves only 3 channels, Channels 1, 6 and 11, to be used at a single instance of time. Using more than 3 channels in the 802.11b wireless network would result in performance degradation. Also, any metal devices or thick walls absorb 802.11b signals and deteriorate the signal strength drastically [1]. Using cordless phones and wireless devices operating in the 2.4GHz frequency range like bluetooth also deteriorate the performance of 802.11b devices [17]. It should be
noted that these were some of the major reasons for the degradation in performance of the RoboCup Unicast and the RoboCup Multicast protocols.

In the later parts of this thesis, any reference to WLAN implicitly means that it is an 802.11b wireless network.

2.4 MAC Protocols In 802.11

*Carrier Sense Multiple Access with Collision Avoidance* (CSMA/CA), is used as the basic transmission protocol in every 802.11 wireless device. In any 802.11 capable wireless device, *Frame Formatting, Error Checking, Fragmentation and Reassembly* and *Channel Allocation Procedures* are all done at the MAC sub layer [12]. Figure 2.3 shows two basic access methods provided by the MAC sublayer in an 802.11 device. Each of the methods are used during two different modes of transmission in a wireless medium which are explained below:

1. *Contention Mode*: In this mode, every single device in the network competes with all the other devices within its BSS to gain access to the wireless medium. This contention for network is performed for every packet transmitted. This time period, where all the devices contend with each other for network access, is called *Contention Period* (CP). Section 2.4.5 describes the *Distributed Coordination Function* (DCF) protocol implemented by the wireless devices used during the CP.

2. *Intermittent Contention Mode*: In this mode the network time is divided into intermittent periods of CP and *Contention Free Period* (CFP). During CFP, the access to the medium is controlled by the AP within its BSS. Again during the CP, the devices implement the conventional DCF protocol. The protocol implemented during the CFP is called *Point Coordination Function* (PCF). Most of the existing wireless APs do not implement PCF. PCF is not considered for this thesis, as it does not fall within the scope of this thesis.
2.4.1 MAC Service Data Unit Frames

MAC Service Data Units (MSDU) is defined as the basic unit of data exchange between two 802.11 peer MAC entities. IEEE 802.11 standard divides the MSDU frames transmitted by the MAC sublayer into three different types:

- Management Frames: These are frames used by the wireless STAs for association and disassociation with the AP, timing and synchronization, and for authentication and deauthentication.

- Control Frames: These are frames used for handshaking and positive acknowledgements during the CP and to end CFP.
• **Data Frames**: These are used for transmitting data during the CP and CFP. The data frames can also be combined along with polling and acknowledgements during the CFP.

It should be noted that this thesis does not delve into the details of these frame types and their use. This section is used to explain the notation of MSDU frames and the IEEE 802.11 frame format. Figure 2.4 gives the standard IEEE 802.11 frame format. The *Duration connection ID* field is used later on in Section 2.4.5.

![Figure 2.4. Standard IEEE 802.11 Frame Format](image)

**2.4.2 Difference Between Wired And Wireless LAN**

This section describes the fundamental difference between the wired and wireless LAN at the physical layer. An 802.11 wireless device cannot detect collisions as a conventional wired 802 compatible device does. Following are some of the reasons for the inability of a wireless device to detect collisions:

- The signals emitted by the wireless STAs are weak and deteriorate as the distance increases.

- Hidden Node problem, described in Section 2.4.3, makes the detection of collisions at the receiver’s end impossible.
Most of the present wireless cards available can only operate in Half Duplex\(^1\) mode. In order to detect collisions, the wireless STAs should be able to operate in Full Duplex\(^2\) mode like a normal LAN device does.

Due to the above reasons, wireless STAs in a WLAN will not be able to detect collisions. Thus the access method, CSMA with Collision Detection, used in a normal wired LAN cannot be used in a WLAN. Instead of trying to detect collisions, the wireless STAs try to avoid collisions using the CSMA/CA protocol.

### 2.4.3 Hidden Node Problem

In Figure 2.5, A, B and C represent three different STAs of a wireless network. Each STA has its own visibility range which is represented by a circle surrounding each STA. STAs A and C fall outside of the visibility range of each other. Hence, at any instance of time, there is no way for either A or C to detect whether the wireless network at the other end is idle or not. Consider the case where both A and C need to transmit data to STA B. When A (or C) needs to transmit any data, it waits for the network to be idle (within its area of vicinity) using the carrier sense mechanism explained in a later section. When it finds the network to be idle, it starts transmitting. It is entirely possible that both A and C start transmitting data to B, after sensing the medium to be idle at their respective ends. Such situations would result in a collision of packets at the receiver’s end. The sender cannot detect this collision in advance by using the conventional Collision Detection technique employed in wired LANs. This is defined as the *Hidden Node Problem*. Such scenarios are very common in wireless networks due to various reasons. The smaller area of vicinity of wireless devices and the frequent mobility of STAs in and out of the range of the AP are some of the reasons for the *Hidden Node Problem*.

\(^1\)A mode where the device can either send or listen for data, but cannot perform both the operations simultaneously.

\(^2\)Device operating in this mode can both receive and transmit packets simultaneously.
Figure 2.5. Hidden Node Problem: Station A and Station C are hidden from each other and cannot detect whether the medium is free at the other end.

2.4.3.1 Interframe Space

*Interframe Space (IFS)* is defined as the time interval between two MSDU frames. Every STA in the network first determines whether the medium is idle, according to the specifications of the carrier sense mechanism mentioned in Section 2.4.5, for the corresponding IFS. The IEEE 802.11 standard defines four different IFSs. These four IFSs differ in terms of the length of the time interval. The type of frame being transmitted and the present situation of the WLAN determines the type of IFS selected. The MSDU transmission following a short time interval denotes a higher priority transmission, and one with a higher time interval denotes a lower priority transmission. Following are the four IFSs and the different situations where they are used:

1. *Short Interframe Space (SIFS)*: SIFS is defined as the idle time allowed by any 802.11 STA before transmitting an Acknowledgement (ACK) frame, a Clear To Send (CTS) frame, second and subsequent packets of a fragment burst, or an ACK frame responding to polling by a PCF. SIFS is the shortest among all the IFSs. It is also used between two frame fragments belonging to the same
network connection, where two STAs have setup a network connection and need to maintain access of the medium for transmitting subsequent fragments of the current connection. Other STAs requiring the transmission of packets other than those mentioned above use a longer IFS, and thus such STAs wait for a longer time. Therefore, STAs transmitting any of the above mentioned packets get higher priority. SIFS is measured in microseconds ($\mu s$).

2. **PCF Interframe Space (PIFS):** PIFS is defined as the idle time of the STA operating under PCF trying to gain control over the access medium at the start of the CFP. Any STA using PCF is allowed to transmit CF traffic after it senses the network to be idle for PIFS amount of time. PIFS is larger than SIFS, thus STAs waiting for SIFS amount of time before transmitting have a higher priority to access the network when compared to those waiting for PIFS amount of time. The PIFS value is given as the sum of SIFS and slotTime (defined in Section 2.4.4).

3. **DCF Interframe Space (DIFS):** DIFS is defined as the amount of time waited by a STA after it senses the channel as idle for the basic access of a WLAN (for the first packet in any normal transmission). Again, DIFS is larger than PIFS. Thus, STAs waiting for PIFS amount of time before transmitting have a higher priority for network access when compared to those waiting for DIFS amount of time. The DIFS value is given as the sum of PIFS and slotTime.

4. **Extended Interframe Space (EIFS):** EIFS is used by the DCF in case of errors in transmission. EIFS is larger than DIFS.

Of the above four Inter Frame Spaces, the effect of SIFS, DIFS and EIFS is taken into consideration. PIFS is applicable only when polling is done by the AP. In a normal contention mode environment, SIFS is used by a STA when sending MSDU frames over the same connection to a single destination or when the receiver sends an
Acknowledgement back to its sender. All of the three protocols studied in this thesis are based on a connectionless protocol. Hence, SIFS is used by only the receiving STAs when they send an acknowledgement back to the sender after receiving an error-free MSDU frame.

2.4.4 Carrier Sense Multiple Access With Collision Avoidance

Carrier Sense Multiple Access with Collision Avoidance is the fundamental access method of any 802.11 wireless device. In a Carrier Sense Multiple Access (CSMA) environment, any station which desires to transmit data first senses if the medium of transmission is idle. If the medium is busy, i.e another station is transmitting at the same time, the station defers its transmission for a random period of time. It then repeats the same process again until the medium is idle. If it finds the medium idle, it starts transmitting. The CSMA technique is implemented by all wireless devices for every single packet transmitted over the WLAN.

A conventional wired LAN uses the Collision Detection technique to detect any collisions caused during transmission. Collisions normally occur when more than a single station sense the channel to be idle and start transmitting at the same time. Stations operating in a wired LAN can detect such collisions at the MAC layer and retransmit the packet. Collision Detection cannot be used by stations working in a wireless LAN due to the reasons described in Section 2.4.2. In order to overcome the shortcomings of a wireless LAN, 802.11 uses CSMA along with Collision Avoidance (CA) with a random backoff time.

In CSMA/CA, before transmitting data, every station waits for a DIFS or EIFS (depending on the scenarios explained in Section 2.4.3.1) amount of time and senses if the medium is idle. If the medium is busy, the station defers the transmission until the end of the current transmission. After sensing the medium to be idle for the required period of time (DIFS or EIFS), every station employs an additional random Backoff-Interval to further avoid collisions between stations sensing for idle traffic at the same time. The backoff time interval is generated using the following formula:
Backoff-Interval = n*slotTime; where n is a random value between 0 and CongestionWindow (CW). slotTime is a value defined such that a STA will always be capable of determining whether the medium was busy or idle during the previous slotTime.

This Backoff-Interval timer is decremented by slotTime for each idle backoff slot. If the medium is busy during any backoff slot, then the Backoff-Interval is not decremented for that slot. Once the timer expires (reaches zero value), the station starts transmitting. Choosing the random value allows multiple STAs contending for access of the medium to avoid collisions.

Apart from the normal CSMA/CA technique, 802.11 protocol defines two protocols called Distributed Coordination Function (DCF) and Point Coordination Function (PCF). This thesis concentrates on the basic principles of DCF and does not deal with PCF.

2.4.5 Distributed Coordination Function

Distributed Coordination Function (DCF) is the “basic medium access protocol” that allows multiple 802.11 capable devices to share a common wireless medium using the CSMA/CA technique. DCF also employs a Positive Acknowledgment technique along with the CSMA/CA. Positive Acknowledgment is a technique in which every end station responds to the sender with an acknowledgment called an ACK frame if the Frame Check Sequence (FCS) of the received frame is correct. This ACK frame ensures that the receiver did receive the frame transmitted by the sender. In case of not receiving an ACK frame from the receiver, the sender tries to retransmit the frame for a fixed number of times before it drops the frame. The maximum number of retransmissions that can be performed on a given frame depends on the length of the frame relative to a set threshold which is explained later on this section.

DCF employs the carrier sensing mechanism in two ways. Physical carrier sensing is performed using the CSMA/CA technique explained in Section 2.4.4. Apart from this, DCF also uses a virtual carrier sensing mechanism provided by the MAC layer.
DCF uses *RequestToSend* (RTS) and *ClearToSend* (CTS) frames prior to actual data transfer. These frames allow the sender and receiver to inform other end devices in the network that the medium is being used by the former to transfer data. RTS and CTS frames contain a *Duration* field (as shown in Figure 2.4) that gives an estimate of the time that the medium will be busy. All other devices use the value in the *Duration* field to set a timer called *Network Allocation Vector* (NAV). All stations would wait for the NAV counter to be zero before sensing the network for traffic. The RTS frame enables the sender to inform all the stations within its vicinity about the impending traffic. The CTS frame allows the receiver to Acknowledge the sender of its RTS frame and also to inform other STAs, which might be hidden from the sender, about the impending traffic. This helps in solving the *Hidden Node Problem* explained in Section 2.4.3.

It should be noted that the positive Acknowledgement and the use of RTS/CTS techniques is possible only in case of unicast frames, i.e MSDU frames sent with a single unicast MAC address. IEEE 802.11 standard does not allow the use of the above techniques in case of Multicast and Broadcast frames since there is no single destination which would respond with an ACK frame.

DCF uses the RTS/CTS frames only when the length of the MSDU to be transmitted is greater than a threshold value of *RTS Threshold*. If the length of the MSDU is less than *RTS Threshold*, the sender does not use RTS/CTS frames to avoid unnecessary overhead. The updates transmitted by the Base Station are approximately 250 bytes, which is far less than the *RTS Threshold*. Hence, this thesis does not delve into further details of RTS/CTS frame sequences.

The MAC layer of 802.11 device also deals with fragmentation and de-fragmentation but is only done when the MSDU frame being sent is greater than a threshold value of *Fragmentation Threshold*. In the context of this thesis, all MSDU frames in question will be far less than the *Fragmentation Threshold*. This thesis does not elaborate on the fragmentation mechanism of the 802.11 standard.
2.4.6 Acknowledgement Procedure

As explained in Section 2.4.4, every frame sent with a unicast receiver address is ACKed back by the receiver. When the receiver receives an error-free frame, it waits for SIFS and then sends an ACK back to the sender. If the receiver receives a frame with errors, it discards the frame. The sender waits for the ACK until a timeout called the ACKTimeout interval. If the sender fails to receive an error-free ACK frame for an already sent frame before the timeout, it will consider this as an error. Note that there may be situations where the receiver sent an ACK but the sender did not receive it. The sender does not distinguish between an error in the ACK and an error in the actual frame transmitted. If the sender does not receive an ACK before the ACKTimeout expires, it invokes its backoff procedure and re-contains for access to the medium. It then tries to retransmit the lost frame. The number of times the sender attempts to retransmit a frame before dropping it is determined by two values called ShortRetryLimit and LongRetryLimit. If the frame being sent is not greater than the RTS_Threshold, the sender will try ShortRetryLimit times before dropping the frame. Otherwise, it will retry for LongRetryLimit times before dropping the frame. The procedure followed during the retransmissions is not different from the normal transmission.
3. Robust Wireless Multicast Protocol For RoboCup

In this chapter the need for a Robust Wireless Multicast Protocol, in the context of the International RoboCup competition, is presented. Firstly, a brief overview of the RoboCup competition and the two protocols used by the Ohio University (OU) RoboCup team in the competition is given. A detailed theoretical analysis of both these protocols is then performed and the reasons for their failure in the competition amidst poor network conditions are presented. Finally, a third protocol, called Robust Wireless Multicast Protocol, is proposed and its advantages over the existing protocols are presented.

Section 3.1 gives a brief description of the RoboCup international competition and the challenge it poses for wireless communication during the game. Section 3.2 gives the application level specifications of the RoboCup team at Ohio University and the different protocols used to implement these specifications. Section 3.2.1 and Section 3.2.2 detail the RoboCup Unicast protocol and its limitations respectively. Section 3.2.3 and Section 3.2.4 describe the RoboCup Multicast protocol and its limitations respectively. Finally, Section 3.2.5 and Section 3.2.6 delve into the proposed Robust Wireless Multicast Protocol and its advantages over the other two protocols. Section 3.2.7 then illustrates the implementation details of the Robust Wireless Multicast Protocol.

3.1 RoboCup Introduction

The RoboCup is a soccer competition between teams consisting of autonomous mobile robots. Each team consists of five robots and a Base Station (BS). The game is played on a pool table size field. The BS is used to send strategic information
to the robots. The network medium between the BS and the robots is wireless to facilitate the easy functioning of the game. A camera placed over the field monitors the position of all the participating ten robots. The BS collects this information from the camera and uses it to develop its strategy. The BS can transmit any kind of information to its robots using the wireless channel provided at the competition.

The competition is played in a closed stadium and comprises of multiple teams playing against each other. Typically, the number of games being played simultaneously range from around 10 to 20. Each team playing the game has its own Access Point (AP) and operates in one of the 11 channels explained in Section 2.3. It should be noted that for a conventional 802.11b wireless network, no more than 3 channels can be used simultaneously. But typically, there will be more Access Points active at the same time during the competition and hence more than 3 channels will be in use. This results in multiple teams sharing the same channel and also adjacent overlapping channels during the competition. Wireless devices operating in the same channel typically share the bandwidth. But those operating in adjacent and overlapping channels will observe high error rates due to collisions. This phenomenon is the major reason for the extreme network conditions observed during the competition.

Prior to the proposed solution, the OU RoboCup team used two different approaches to transmit data packets from the BS to the robots. A brief overview of these methodologies and their drawbacks are explained in the following sections.

### 3.2 OU RoboCup Application layer Protocol

The RoboCup competition is played in a real time environment. Accordingly, the application developed should support the features of a real time application. The application layer protocol differentiates the data transmitted over the wireless network into two different types of packets depending upon their characteristics as follows:

- **Vision** updates are data packets which describe the position of robots (opposition team’s robots also). Vision updates are transmitted to the robots at two
different rates, 30 Hz and 60 Hz, depending on the situation of the game. Vision update is same for all the robots, and is transmitted to all the robots of the team to keep them updated of the game situation. The method used to transmit these updates to the robots is protocol dependant.

- **Command** updates are data packets which are used to send commands to the robots from the BS. Command updates are transmitted to the robots at a rate similar to that of the Vision updates. The Command updates provide information to robots about the action to be taken at that particular instance. Similar to the Vision update, the method used to transmit the Command update is protocol dependant.

The BS sends the Vision and Command updates periodically to all the robots.

User Datagram Protocol (UDP) [18] is preferred over Transmission Control Protocol (TCP) [9] for sending packet updates for two reasons. The first reason is to avoid the retransmissions made by TCP for lost packets at the transport layer. The second reason is to ensure that new data is given higher preference than old data. This is necessary in a real-time application where current information always outdates the previous data and gains higher precedence. By using UDP instead of TCP, the extra overhead generated by TCP at the transport layer (acknowledgements and retransmissions) is also avoided.

Section 3.2.1 and Section 3.2.3 give a brief overview of the previous two approaches used by the OU RoboCup team to transmit updates over the WLAN.

### 3.2.1 The RoboCup Unicast Protocol

Figure 3.1 depicts the working of the RoboCup Unicast protocol. In this method a single Vision update is transmitted as a Unicast packet to each of the robots’ unicast address. Thus every Vision update results in 5 identical packets, each of which is transmitted to one of the robots. This ensures that all the robots receive the Vision updates. On the other hand, Command updates are different for each Robot, and
informs every robot about the next action to be performed by it. Thus, on average a minimum of 5 Vision updates are transmitted at a rate of 30 or 60 Hz, and 5 Command updates are transmitted at a similar rate. The combined size of the single Vision update and all the five Command updates is less than 250 bytes.

Figure 3.1. The RoboCup Unicast Protocol: Updates are transmitted to each of the robots individually. Note that the packet transmission is done at different instances of time though the figure indicates as if they are transmitted simultaneously. The next packet in queue is transmitted only after the previous one has been transmitted successfully or dropped after the number of failed retransmissions reaches ShortRetryLimit}
3.2.2 Problems With The RoboCup Unicast Protocol

A satisfactory performance was observed when Vision and Command updates were transmitted as Unicast packets to all the robots in the lab. However, problems such as high packet errors and huge delay were observed during the competition due to the network environment at the competition. As mentioned in Section 3.1, there could be multiple teams in the competition using the same wireless network channel and also teams operating in adjacent channels also. Typically the competition is conducted in a closed stadium. At any point of time, more than one game could be played in the same closed building. In one of the competitions held at Japan in the year 2002, approximately 100 wireless devices with a few dozen Access Points were reportedly seen active. Moreover, the presence of various mobile devices such as “cell phones”, and “laptops”, that the spectators and participants bring along to the competition, should also be taken into account. The signals transmitted by these devices also caused interferer bursts. This scenario lead to a large number of collisions and resulted in huge packet losses.

The RoboCup Unicast protocol, by transmitting 5 packets for each Vision update, further increased the probability of packet loss rate. As described in Section 2.4.5 a lost or corrupted packet is not acknowledged by the receivers’ MAC layer. The updates transmitted by the BS are far less than the RTS_Threshold. In an 802.11b WLAN, every unacknowledged unicast packet is retransmitted ShortRetryLimit number of times before the MAC layer drops the packet. The large packet losses thus result in large number of retransmissions. Moreover, each retransmission adds up to the network overhead, which in turn results in more packet loss. Ultimately it results in increase of the probability of losing a packet. This increase is even higher in case of the RoboCup Unicast protocol due to the 5 packets it generated for each packet update. Chapter 4 explains in detail, the percentage of losses observed for the RoboCup Unicast Protocol and how it increases exponentially with respect to time.
Another major problem that this protocol suffered was the huge amount of delay that it introduces. It should be noted that more retransmissions at the MAC layer result in giving higher precedence to outdated packets by retransmitting old data instead of the current data at hand. Thus, in an attempt to retransmit lost packets $ShortRetryLimit$ number of times, the protocol introduced delay in the transmission of current packet. This lead to the same potential problem which was being avoided by selecting UDP instead of TCP. Any application which introduces huge packet delay will not yield good results in a real time environment, which was the case with the RoboCup Unicast protocol.

Moreover, the RoboCup Unicast Protocol did not satisfy the basic requirement of ensuring that data was transmitted to all the robots. The RoboCup Unicast Protocol created 5 duplicate packets of the same Vision update and transmitted each of them individually to one of the robots. Thus even when the Vision update reached a robot successfully, it did not ensure that the update was received by all the other robots also. It was entirely possible that the BS failed in transmitting the same packet to the other robots. And this caused serious problems in building strategy for the actual game.

Based on the above facts, it can be concluded that due to high probability of packet loss, large number of retransmissions and unnecessary packet delay, the RoboCup Unicast protocol did not succeed in the competition. In order to overcome the above drawbacks, the OU RoboCup team implemented a second protocol which transmitted minimal number of packets and at the same time avoided unnecessary retransmissions.

### 3.2.3 The RoboCup Multicast Protocol

An effort was made to reduce the number of transmissions, so as to reduce the packet losses and to avoid packet delay. As mentioned before, the Vision updates transmitted to all the robots were identical. Instead of transmitting Vision and Command updates separately, it was found beneficial to transmit a single packet containing all the information. This increased the packet size by a few bytes. It was
noted before that the combined size of the Vision update and the five Command updates was around 250 bytes. That was smaller than both the RTS Threshold and Fragmentation Threshold values mentioned in Section 2.4.5.

Instead of transmitting the Vision and Command updates separately, the RoboCup Multicast protocol transmitted Vision and Command updates encapsulated into a single update packet. This reduced the network overhead by 5 times. Every update contained the Vision update and Command update for all five robots. Previously, all the robots only knew the position of all the other robots. In addition to that, the RoboCup Multicast protocol also ensured that each Robot knows the action being performed by the other four. The new approach helped the team to build a strategy that leads to better performance in the game.

Figure 3.2 depicts the packet transmission made by the RoboCup Multicast Protocol. The BS transmits all the packets destined to a designated multicast address and port. The wireless interface of each robot joins that multicast group and listens for packets destined to that multicast address. The robots capture only those packets that are destined to the designated destination port with source address as that of the BS. Multicast address 224.1.1.1 and port 5001 was used in all the experiments conducted later on.

This approach solved the need of transmitting minimal number of packets and avoiding unnecessary delay at the MAC layer as pointed out in Section 3.2.2. The performance was exemplary in terms of bandwidth used, amount of network overhead reduced and reduced packet delay. Although it solved the problems faced by the RoboCup Unicast protocol, the RoboCup Multicast protocol did not fare well at the competition due to another set of problems

3.2.4 Problem With The RoboCup Multicast Protocol

The RoboCup Multicast protocol, similar to the RoboCup Unicast protocol, did not succeed at the competition. Though the reasons for the failure of the RoboCup Multicast were different to that of the RoboCup Unicast protocol. The RoboCup
Multicast performed well in reducing network overhead but failed in situations where the packet losses were high due to the overloaded channels.

In the RoboCup Unicast protocol, reliability in packet transmission was achieved at the MAC layer using the MAC layer acknowledgements from the robots. However, the RoboCup Multicast protocol transmitted packets to a single multicast address. This resulted in elimination of the MAC layer acknowledgements. Section 2.4.5 explained the reason for absence of acknowledgements in transmissions involving multicast packets. This resulted in the protocol being highly unreliable in comparison...
with the RoboCup Unicast protocol, which was highly reliable due to the MAC layer acknowledgements and retransmissions. A wireless network is highly unreliable and have higher loss rates when compared to a conventional wired network. This difference would be even higher under increased network overhead. Moreover, high contention for network bandwidth was observed at the competition between wireless devices that operated in same and adjacent channels. This heavy utilization of available channels further increased the probability of packet losses. Such situations create the need for a robust protocol. The RoboCup Multicast protocol failed to provide such robustness and reliability. This resulted in huge packet losses and resulted in failure of the protocol at the competition.

This thesis is to provide a solution to the problems faced by the above mentioned protocols in the competition. The ideal solution for the problem could be characterized by the following features:

- Reduce Network Overhead
- Reduce redundant re-transmissions of old data
- Reduce packet delay
- Reduce total packets transmitted and still transmit data to all
- Current packet should be given precedence over obsolete packet
- Ensure reliable transmission

3.2.5 The Robust Wireless Multicast Protocol

A new protocol is designed by considering the above discussed features. In the new approach, called the Robust Wireless Multicast Protocol (RWMP), a single packet is transmitted by the BS as in the RoboCup Multicast Protocol. However, the packet is transmitted to one of the five robots instead of transmitting it to a multicast address.
Thus, the packet transmitted is no longer a multicast packet, but a unicast packet transmitted to one of the five robots called the CAPTAIN.

Within a WLAN, every end station can capture the packets destined to any of the stations if they are in the same vicinity range. This is similar to an Ethernet LAN where every machine can listen to packets transmitted or received by any other machine on the same Ethernet segment. The Robust Wireless Multicast Protocol applies this concept to ensure that all the robots receive the single packet transmitted to the CAPTAIN. When the packet is transmitted to the CAPTAIN, the CAPTAIN and the AP ensure the reliable transmission of the packet at the MAC layer with the help of the acknowledgement transmitted by the CAPTAIN to the AP.

Since the area of the field is limited, and the range of the robots and the AP is larger than the area of the field, this protocol assumes that the packets received by the CAPTAIN can also be received by all the other robots on the field. Also the wireless interface of all the robots are configured to be in the same network. With the above assumption, each of the other four robots is set to listen in Promiscuous mode\(^1\) for packets destined to their network (but not to itself).

As shown in Figure 3.3, the BS transmits only a single Unicast packet destined to the CAPTAIN but the other four robots also receive the data by listening in promiscuous mode over the WLAN. The robots listening in promiscuous mode do not have to worry about acknowledging the AP, since the acknowledgement is done by the MAC layer of the CAPTAIN. The other robots also do not have to keep track of who the CAPTAIN is, at any instance of time. They only have to identify whether the destination address belongs to their network or not.

\(^1\)In a normal mode the Network Interface Card captures packets destined only to its hardware address and filters all the other packets from reaching the data-link layer. Promiscuous mode is a special mode where the Network Interface Card is told to listen for all the packets over the network and not for packets destined only for that particular hardware address.
Figure 3.3. The Robust Wireless Multicast Protocol: Transmits single unicast packet to the Captain (Robot1). The other four robots listen for the packet in promiscuous mode indicated by the dashed lines.

3.2.6 Advantages Of The Robust Wireless Multicast Protocol

The Robust Wireless Multicast Protocol is a combination of both the RoboCup Unicast and the RoboCup Multicast Protocols. It has the characteristics of the RoboCup Multicast Protocol, as it only transmits one single packet for each update transmitted by the Base Station (BS). It also has the characteristics of the RoboCup Unicast Protocol, as the MAC layer acknowledgements and retransmissions between the BS and the CAPTAIN make it robust and reliable similar to the former. Due to these characteristics, it has the advantage that both the RoboCup Unicast and the
RoboCup Multicast protocols had. Following is an explanation of how the RWMP overcomes all the limitations that the earlier two protocols faced.

As explained in Section 3.2.2, the RoboCup Unicast protocol suffered huge packet losses due to the large network overhead that it contributed to, by unnecessary retransmissions and packet transmissions. The Robust Wireless Multicast Protocol overcomes this limitation by transmitting fewer packets as compared to the former. It also serves the purpose of updating all the robots with the game situation, and does that without introducing any time delay unlike the former. The RoboCup Unicast Protocol generates five acknowledgements for each unique data packet transmitted. On the other hand, the Robust Wireless Multicast Protocol makes only the CAPTAIN to generate the acknowledgements and thus reduces the total acknowledgements generated by five times.

Due to the huge packet losses observed at the competition, the RoboCup Unicast Protocol generated more retransmissions than the Robust Wireless Multicast Protocol as it has to ensure the reliable transmission of more number of packets than the later. This resulted in huge packet delay as explained in Section 3.2.2. Even while experiencing huge packet losses, the Robust Wireless Multicast Protocol makes less number of attempts for retransmissions and backoffs\(^2\) (for a single unique packet transmitted by the BS), before it finally drops the packet. As discussed in Section 2.4.3.1, a packet loss results in the AP deferring its transmission for Extended Inter Frame Space (EIFS) followed by a backoff procedure. During high error rates, the RoboCup Unicast protocol would have to follow the above procedure \(5 \times ShortRetryLimit\) times for each unique packet update, before it can transmit the next update in queue. The amount of delay observed in such scenarios would increase tremendously and might exceed the update timer of 30 (or 60 Hz). However, the Robust Wireless Multicast Protocol reduces this delay by five times. Thus the probability that, the RoboCup Unicast Protocol still tries to retransmit the lost packets even after the expiration of

---

\(^2\)In case of packet loss 802.11 wireless protocol backoff a random time as explained in Chapter 2
the update timer, is far more when compared to that of the Robust Wireless Multicast Protocol. This results in the conclusion that the later also overcomes the limitation of high packet delays mentioned in Section 3.2.2 that the RoboCup Unicast Protocol faced in competition.

The RoboCup Unicast Protocol experienced huge network overhead due to unnecessary duplicate packets (as every packet was transmitted five times). And the resulting acknowledgements and retransmissions generated from the transmission of these duplicate packets further increased this network overhead. The Robust Wireless Multicast Protocol, similar to the RoboCup Multicast Protocol, overcomes this problem by reducing the total number of packets transmitted by more than 5 times in normal network conditions. The difference is more in situations when the network experiences heavy packet losses. In fact, the RoboCup Multicast Protocol does even better than the Robust Wireless Multicast Protocol in terms of reducing network overhead by eliminating the acknowledgements and retransmissions as explained before.

Increasing network overhead and huge packet delay were major reasons to replace the RoboCup Unicast protocol with the RoboCup Multicast protocol. The RoboCup Multicast protocol solved the problem of packet delay by totally eliminating acknowledgements and retransmissions. However, this resulted in loss of reliability at the MAC layer as explained in Section 3.2.4. The Robust Wireless Multicast Protocol overcomes this limitation of the RoboCup Multicast protocol by transmitting unicast packets instead of multicast. Thus, when the network is experiencing intermittent packet losses, it prompts the AP to retransmit the packets that were not acknowledged by the CAPTAIN. In case of extreme conditions where the probability of packet loss is very high, the Robust Wireless Multicast Protocol attempts to retransmit each lost packet ShortRetryLimit number of times as described in Section 2.4.5 before it drops the packet.
The performance of the Robust Wireless Multicast Protocol could be further increased in situations of frequent packet losses. The BS can switch between the robots to select the CAPTAIN in situations when the percentage of losses exceeds a Threshold value\(^3\). Its always possible that the CAPTAIN of the team goes down due to some internal problem\(^4\). Even in such situations, the BS can switch the CAPTAIN and use another robot as its target. This would increase the robustness of the protocol. The selection criteria for chosing the CAPTAIN can be varied depending on the application requirements and the situation of the game. Selecting the CAPTAIN using a Round Robin technique is one such option.

The only limitation of the Robust Wireless Multicast Protocol is that it requires all the robots and the AP to be within each other’s vicinity range. Thus, if the CAPTAIN acknowledges a packet, then it ensures that the packet is received by all the other robots. However, the above statement may not be true in conditions when the network environment, surrounding the robots, is not identical. A few such situations are explained in Chapter 4.

3.2.7 Implementation Details Of The Robust Wireless Multicast Protocol

The BS transmits all the packet updates to the CAPTAIN to destination port 5001\(^5\) (RELAY-PORT). The robots capture the updates at the final destination port 2001\(^6\) (DESTINATION-PORT). An application runs on all the robots which forwards the packet updates destined to the CAPTAIN’s RELAY-PORT to its corresponding DESTINATION-PORT.

The application is divided into two major components called PACKET-SNIFFER and PORT-FORWARD. Both the applications run on all the robots at any given instance of time during the game. The first application, PORT-FORWARD, captures

---

\(^3\)No optimum value has been proposed by the author yet
\(^4\)Internal problem could be classified as battery problem or some mechanical problem in the robot or for that matter any malfunction which will make that robot a liability for the team
\(^5\)You can opt for any unassigned port number.
\(^6\)You can opt for any unassigned port number.
Figure 3.4. The Robust Wireless Multicast Protocol Flowchart: This flow chart describes how the packet update transmitted by the BS reaches the final destination port (2001) of all the robots. If the robot is the CAPTAIN then the UDP Relay application captures the packet and transmits it to the final destination port. Else the Packet Sniffer application captures it and transmits it to the Destination port.
the packet transmitted by the BS if the robot on which it is running is the current CAPTAIN of the team. If the robot on which the application is running is not the CAPTAIN, the packets are captured by the other application called PACKET-SNIFFER. PORT-FORWARD listens for packets at the port RELAY-PORT of the HOME-MACHINE\(^7\). PACKET-SNIFFER listens for packets destined to the CAPTAIN, if the HOME-MACHINE is not the CAPTAIN, in Promiscuous mode. In either case the packets captured are forwarded to the DESTINATION-PORT over the loopback interface of the HOME-MACHINE. Figure 3.4 gives the flow chart of the application.

These applications ensure that any packet transmitted by the BS is captured and transmitted to the DESTINATION-PORT of all the robots. The functionality of the protocol is preserved irrespective of which robot acts as the CAPTAIN of the team. This allows the BS to change the CAPTAIN depending upon the current situation of the game.

The next chapter contains the results of various experiments conducted on the RoboCup Unicast, the RoboCup Multicast and the Robust Wireless Multicast protocols. A detailed analysis of all the results observed is then presented.

\(^7\)It is the robot on which the application is running on.
4. Experimental Results

This chapter illustrates the experiments conducted to test the performance of all the three protocols explained in chapter 3. The experiments characterize the behavior of the protocols over several parameters such as the rate at which the packets are transmitted and the external interference generated on the same channel or adjacent channels relative to that of the robots. This external interference is referred to as \textit{ADDED\_TRAFFIC}. It should be noted that the ADDED\_TRAFFIC is not any unknown traffic outside the experiment setup. It is generated during the experiment to emulate the high utilization of bandwidth observed during the competition on the same channel and adjacent overlapping channels relative to the channel used by the OU RoboCup team in the 802.11b spectrum.

Every experiment is repeated on all the three protocols, the RoboCup Unicast Protocol, the RoboCup Multicast Protocol and the Robust Wireless Multicast Protocol (RWMP), in approximately\(^1\) the same network environment. The results obtained, are used to derive the following values corresponding to each protocol:

- Number of packets lost during the experiment, denoted by \textit{TOTAL\_LOSSES}
- Time taken to transmit the packets relative to the first packet received, denoted by \textit{ELAPSED\_TIME}

The above two factors are used to compare the three protocols and to analyze the performance of the protocols.

\(^1\)We cannot ensure perfectly identical scenario for any two experiments as there is no control over the external wireless networks running in the lab. However every effort was made to avoid any unwanted traffic that could effect the network environment of the experiments
A network diagnostic application called TestTCP (TTCP) [8] is used by the Base Station (BS) to generate UDP packets of the same size as that of the updates transmitted at the competition. Section 4.1 gives a detailed explanation of the TTCP application used and its functions. Section 4.2 gives an outline of the equipment used for the experiments. Section 4.3 gives a brief overview of the experiments conducted and then delves into the experiment setup used for each protocol. Sections 4.4, 4.5 and 4.6 give a detailed description of the experiments conducted in varying network conditions and the results observed in each case.

4.1 TestTCP Modification

TestTCP (TTCP) is a standard network diagnostic tool used to generate TCP or UDP packets based on the input parameters passed on to the application. TTCP is an application used to measure network bandwidth and is widely used to test network performance. Thus, being a standard network diagnostic tool, TTCP is preferred to the existing application that the RoboCup team has, for transmitting updates to the robots. This thesis mainly concentrates on identifying the loopholes of the applications and protocols used by OU RoboCup team, and hence the use of any existing software belonging to OU RoboCup team is avoided.

For all our experiments, TTCP is used to generate UDP packets of length 250 bytes, which is approximately the size of the updates transmitted by the BS. TTCP transmits the UDP packets to port 5001 by default. The rate at which the packets are transmitted by the BS, and the amount of ADDED\_TRAFFIC applied, are varied to test the performance of the protocols in different situations.

TTCP\_RATE is defined as the number of ‘unique’ packets transmitted by the BS to the robots every second.

The word ‘unique’ is used for identifying packets with the same payload. Remember that for the RoboCup Unicast protocol, every update is transmitted to all the robots. While calculating TTCP\_RATE, packets with the same payload are con-
sidered as a single unique packet. Thus, though the BS transmits more number of packets in case of the RoboCup Unicast protocol when compared to the other two protocols, the effective TTCP RATE is still going to be same as in the other two protocols.

Though TTCP can be used as a packet generator for the experiments, the packets generated and the method of packet transmission used by TTCP do not allow it to be used directly by the BS. Three main shortcomings of TTCP are identified. The modifications made to overcome these shortcomings are explained below.

Firstly, TTCP does not have an option to transmit packets at a specific rate. TTCP just tries to transmit UDP packets as fast as it can, and does not give an option to transmit the packets intermittently at a fixed rate. It does not allow the BS to introduce any time gap\(^2\) between the packets transmitted. The Application layer protocols specified in chapter 3 require the BS to transmit updates at a fixed rate. In order to satisfy this constraint, the original application is modified to introduce a proper time delay between any two successive packets transmitted by the BS. The time delay introduced between packets can be varied to change the rate at which the packets are transmitted by the BS. This packet rate is denoted as the TTCP RATE for that particular experiment.

Secondly, TTCP uses the same payload for all the packets generated. But the experiment needs the packets to be numbered so as to determine the number of packet losses. Differentiating original packet from its retransmissions will also be impossible without any number indicating the Sequence Number of the packet. Hence, the original TTCP is modified to include the Sequence Number within the payload. By identifying the Sequence Number in the payload transmitted by the modified TTCP, it is possible to calculate not only the total number of packets lost but also the

\(^2\)In the previous competition, approximately 30-60 packet updates were transmitted every second. The packets transmitted should be spread evenly over the one second time period.
exact packets that were lost. In addition to this, the packets that were retransmitted during the experiment can also be identified.

Finally, TTCP is also used for the experiments involving the testing of the RoboCup Unicast protocol. The original TTCP does not provide an option to transmit packets to multiple destinations. In the RoboCup Unicast protocol, each packet update is to be transmitted to all the robots. In order to facilitate this, an option to transmit packets to multiple destinations is introduced in the modified TTCP application.

In summary, three changes were made to TTCP as follows:

1. An option to introduce delay between packets was introduced. The amount of delay introduced is same for any two consecutive packets and is determined from the TTCP\_RATE value set.

2. Each packet is numbered before it is transmitted. This Sequence Number is introduced at the beginning of the payload of the UDP datagram generated.

3. An option to transmit packets to multiple destinations was facilitated. This option is used in the experiments involving the testing of the RoboCup Unicast protocol.

Figure 4.1 depicts the method of packet transmission used for each of the three protocols. Any reference to TTCP in the rest of this thesis indicates to the modified TTCP application including the above features.

4.2 Equipment Used For Experiments

Three robots are used for the experiments conducted, and are identified as Gumby5, Gumby6 and Gumby8. Though five robots are required to simulate the exact conditions of the game, the results obtained can be projected to get the results for five robots. A total of three Access Points (APs) are used during the experiments. Access Point (AP) A is used to connect the Base Station (BS) and the robots. The BS runs Linux Operating System, and is connected to AP A over a 100 Mbps wired
link. In all the experiments conducted, AP A always runs on channel 2 (see section 2.3). The robots also run Linux and have an Orinoco pcmcia wireless card, a wireless product of Lucent Technologies, installed on them. The pcmcia card is the only external interface on any Robot, and is used to connect the Robot to the AP A over the WLAN.

Two other APs, B and C, are used only when ADDED_TRAFFIC is applied during the experiments. Two laptops are used to generate the ADDED_TRAFFIC. Laptop X_PC is a Gateway machine running Linux. Laptop Y_PC is a Dell Inspiron 8000 series machine, also running Linux. The Laptops connect to either AP B or AP C, but are never connected to AP A. An experiment involving AP B or AP C implicitly means that the experiment is conducted to test the performance of the protocols in presence of ADDED_TRAFFIC, and vice versa.
AP A is a product of *Apple inc.* and is 802.11b capable. APs B and C are wireless routers (APs) from *Linksys inc.* and are also 802.11b capable. APs A and B are capable of both 802.11b and 802.11g technologies and AP C is capable of only 802.11b. All the wireless stations (robots and the two laptops) used in the experiments only have 802.11b capable wireless equipment attached to them. Hence, AP A and AP B would automatically turn onto 802.11b protocol and switch off 802.11g.

### 4.3 Experiments Conducted

In all the experiments conducted, the size of each packet transmitted by the BS is fixed to 250 bytes. This is approximately the size of the packet updates transmitted by the BS in the competition. Each experiment comprises of the BS sending $5000^3$ UDP packets to the robots. It should be noted that, for the RoboCup Unicast protocol, each UDP packet is transmitted to all the three robots. This results in the BS sending a total of 15000 UDP packets. The experiments done are mainly divided into three categories as follows:

- The first set of experiments check the functionality of all the three protocols, the RoboCup Unicast protocol, the RoboCup Multicast protocol and the Robust Wireless Multicast protocol (RWMP), in absence of any *ADDED_TRAFFIC*. These experiments are used to verify the authenticity of all the three protocols specified in chapter 3. They are also used to validate the testing methodology and establish the background for proper analysis of the three protocols. The results observed are given in section 4.4.

- The second set of experiments are used to check the performance of the protocols in presence of *ADDED_TRAFFIC*. AP B and laptops X_PC and Y_PC are used to generate the *ADDED_TRAFFIC*. The details of the *ADDED_TRAFFIC* would give us enough number of packets for a proper detailed analysis of the protocols. Also the experiment would last from 50-125 seconds depending upon the value selected for TTCP_RATE.
generated and the results observed for each protocol are explained in section 4.5.

- The third set of experiments, similar to the previous case, test the performance of all the three protocols in presence of ADDED_TRAFFIC. However, in this case two additional APs, B and C, along with the two laptops X_PC and Y_PC, are used to generate the ADDED_TRAFFIC. The details of the experiments conducted and the results observed are presented in section 4.6.

It should be noted that if an experiment involves more than one AP, it implies that the AP other than AP A, creates a WLAN in which the ADDED_TRAFFIC is generated. Sections 4.3.1, 4.3.2 and 4.3.3 explain the experiment setup for the RoboCup Unicast, the RoboCup Multicast and the RWMP protocols respectively.

### 4.3.1 Experiment Setup For The RoboCup Unicast Protocol

This section describes the experiment setup for the RoboCup Unicast protocol explained in section 3.2.1. This experiment setup is same for all the experiments involving the RoboCup Unicast protocol. The experiments use the three robots and the Base Station (BS) mentioned in section 4.2, which also explains the network connection between them.

All three robots run the \textit{PORT_FORWARD} application defined in section 3.2.7. The BS transmits UDP packets to each of the robots individually using TTCP explained in section 4.1. As mentioned before, TTCP transmits the UDP packets to port 5001 by default. The \textit{RELAY_PORT} and the \textit{DESTINATION_PORT}, defined in section 3.2.7, are set to ports 5001 and 2001 respectively. \texttt{Tcpdump} [7] is used to capture the packets forwarded by the \textit{PORT_FORWARD} application at the \textit{DESTINATION_PORT}.

The order in which a single packet is transmitted to the three robots is noted for each experiment. It should be noted that, for the RoboCup Unicast protocol, this order is important in certain scenarios which are explained later in this chapter. This
order is pointed out explicitly in the tables and helps in analyzing the results. If the packets are transmitted to Gumby5, Gumby6 and Gumby8 in that order, then it is indicated in the corresponding table as \textit{RoboCup Unicast(5,6,8)}.

### 4.3.2 Experiment Setup For The RoboCup Multicast Protocol

This section describes the experiment setup for the RoboCup Multicast Protocol explained in section 3.2.3. As in the previous section, the same BS, three robots and the AP A are used for the experiments. TTCP is used by the BS to generate UDP packets of the desired number, size and TTCP\_RATE. However, the packets generated are multicast packets transmitted to a specified multicast address. The experiment setup mentioned below, is same for all the experiments testing the RoboCup Multicast Protocol.

All three robots run an application called \textit{FORWARD\_MULTICAST}. The BS transmits UDP packets to a multicast address 224.1.1.1\textsuperscript{4}, instead of sending it to the robots directly. The \textit{FORWARD\_MULTICAST} application, running on the robots, listens for packets destined to this multicast address. It then forwards all those packets from the \textit{RELAY\_PORT} (5001) to the \textit{DESTINATION\_PORT} (2001), similar to the \textit{PORT\_FORWARD} application. Tcpdump is used on all the robots to capture the packets at the final \textit{DESTINATION\_PORT}.

### 4.3.3 Experiment Setup For The Robust Wireless Multicast Protocol

This section describes the experiment setup for the Robust Wireless Multicast Protocol (RWMP) explained in section 3.2.5. The setup is same for all the experiments conducted on the RWMP. The same equipment is used in all the experiments. TTCP is again used by the BS to generate UDP packets of the desired number, size and TTCP\_RATE. However, all the UDP packets generated are unicast packets transmitted to one of the robots (CAPTAIN).

\textsuperscript{4}Can use any valid multicast address for implementation.
All the robots run the applications \texttt{PACKET\_SNIFFER} and \texttt{PORT\_FORWARD} defined in section 3.2.7. TTCP transmits the UDP unicast packets to port 5001 of the CAPTAIN. The \texttt{RELAY\_PORT} and the \texttt{DESTINATION\_PORT}, for both the applications, are set to ports 5001 and 2001 respectively. Tcpdump is used to capture the packets at the final \texttt{DESTINATION\_PORT}.

The Robot used as the CAPTAIN is noted for each experiment conducted. This is pointed out in the corresponding table. Thus, if the \texttt{CAPTAIN} is Gumby5, then it is indicated in the corresponding results table as \textit{RWMP(5)}. This information is used later on to analyze the results.

4.4 Behavior Of Protocols In Absence Of ADDED\_TRAFFIC

Four experiments were conducted in this category. Each experiment was repeated on all the three protocols, the RoboCup Unicast protocol, the RoboCup Multicast protocol and the RWMP protocols. The physical experiment setup looks similar to the one shown in figure 4.2. It should be noted though, that the position of the machines, shown in the figure, is not relevant in anyway to the results presented. The four experiments vary in terms of the TTCP\_RATE used at the BS in transmitting packets.

It should be noted though, that the results observed in each of the four cases presented in this section are largely identical. However, the results are still presented so as to maintain parallel structure throughout the experimental analysis made in this chapter. Also in later sections, some of the experiments have results from three trials. It should be noted though, that every experiment was repeated several times. However, for a few experiments conducted, there was significant difference in the results observed between different trials of the same experiment. In such cases, the results from three trials are presented.
Following are the results observed in each of the four cases for the three protocols:

1. In the first case the TTCP_RATE value is set to 40. The TTCP application mentioned in section 4.1 is used directly for all the experiments conducted. The modified TTCP application ensures that the packets are transmitted at the set TTCP_RATE for all the three protocols. The TOTAL_LOSSES observed for each of the three protocols is given in Table 4.1. Observe that the TOTAL_LOSSES is, on an average, zero for all the three protocols on all the robots.
Table 4.1 TOTAL LOSSES Observed, TTCP RATE=40, AP A On Channel 2

<table>
<thead>
<tr>
<th>Robot</th>
<th>RoboCup Unicast(5,6,8)</th>
<th>RoboCup Multicast</th>
<th>RWMP(5)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gumby5</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Gumby6</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Gumby8</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 4.2 TOTAL LOSSES Observed, TTCP RATE=60, AP A On Channel 2

<table>
<thead>
<tr>
<th>Machine Name</th>
<th>RoboCup Unicast(6,8,5)</th>
<th>RoboCup Multicast</th>
<th>RWMP(6)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gumby5</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Gumby6</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Gumby8</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

2. In the second case the TTCP RATE value is set to 60. The TOTAL LOSSES observed for each of the three protocols is given in Table 4.2. The result is similar to that seen in the previous case when TTCP RATE was set to 40.

3. In the third case the TTCP RATE value is set to 80. The TOTAL LOSSES observed for each of the three protocols is given in Table 4.3. The results are approximately same in this case also. A few discrepancies can be seen for the RoboCup Multicast and the RWMP protocols. These are explained later.

4. In the fourth case the TTCP RATE value is set to 100. The TOTAL LOSSES observed for each of the three protocols is given in Table 4.4. The results in this case are similar to the previous case when TTCP RATE was set to 80.
Table 4.3 TOTAL LOSSES Observed, TTCP RATE=80, AP A On Channel 2

<table>
<thead>
<tr>
<th>Machine Name</th>
<th>RoboCup Unicast(8,5,6)</th>
<th>RoboCup Multicast</th>
<th>RWMP(8)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gumby5</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Gumby6</td>
<td>0</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>Gumby8</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 4.4 TOTAL LOSSES Observed, TTCP RATE=100, AP A On Channel 2

<table>
<thead>
<tr>
<th>Machine Name</th>
<th>RoboCup Unicast(5,6,8)</th>
<th>RoboCup Multicast</th>
<th>RWMP(5)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gumby5</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>Gumby6</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Gumby8</td>
<td>0</td>
<td>1</td>
<td>3</td>
</tr>
</tbody>
</table>

Tables 4.1, 4.2, 4.3 and 4.4 indicate that all the three protocols gave approximately 100 percent results in all the four cases. Even when the TTCP RATE was increased to 100, there were hardly any packet losses. The ELAPSED TIME observed on Gumby5 for all the protocols, when TTCP RATE is 100, is given in figure 4.3. Observe that the time taken by Gumby5 to receive the packets transmitted was also same for all the three protocols.

The data files used to draw the graph are generated using the packets captured by Tcpdump at the DESTINATION PORT of the robots. As mentioned before, a few experiments later on have results from three trials. In such cases, the amount of variation in the ELAPSED TIME observed is negligible. Hence the average of the three trials is considered for comparing the ELAPSED TIME observed in the three protocols. However, the percentage of losses observed, varied significantly from one
trial to another in some cases. Hence while comparing the percentage of losses for the three protocols, the results from all the three trials are considered in generating the graphs. Hence the graphs comparing the percentage of losses gives a range of values for the percentage of losses observed for each of the three protocols.

For all the experiments, only the results observed at Gumby5 are analyzed. However, this does not compromise the authenticity of our final analysis though. For the RoboCup Unicast Protocol the order in which the update is transmitted to the robots is varied. Also for the RWMP, the CAPTAIN of the team is varied from one experiment to another. Thus even by analyzing the results only from Gumby5, the overall analysis still covers all the necessary points. Additionally, for the RoboCup Unicast protocol, a single graph indicating the Percentage of Packet Losses on all the three robots is presented. This is to validate our point that the order of the robots, in
the list of the destinations used by the BS, can effect the performance of the robots. These points should be noted while the results are analyzed throughout this chapter.

As noted in case 3, when TTCP_RATE was set to 80, there were a few discrepancies seen in the RoboCup Multicast protocol and the RWMP protocol. Similar behavior was also found in case 4. In case of the RoboCup Multicast Protocol, a few packet losses were observed as shown in Tables 4.3 and 4.4. This behavior could be attributed to the fact that the RoboCup Multicast protocol is unreliable. However these packet losses observed are negligible when compared to the number of packets transmitted. Also observe that the number of packets received by the three robots varied in a few cases. In Table 4.4, Gumby5 lost 2 packets but Gumby6 and Gumby8 lost just 1 packet. Similar results were observed for RWMP. At times the robots listening in promiscuous mode lost a few packets. This is attributed to the varying noise levels on the robots. There will be situations when one Robot gets an error free packet, but another Robot might get a corrupted packet due to noise at its end. Since the RoboCup Multicast Protocol does not have any retransmissions, the number of packets captured varies from one Robot to the other in such scenarios. However in the RWMP, the CAPTAIN has the privilege of seeking a retransmission in such situations. Thus it receives all the packets at least from retransmitted packets if not from the original. But those listening in promiscuous mode do not get retransmissions if the CAPTAIN receives an error free packet. Hence in the RWMP, the CAPTAIN always receives at least as many packets as the other robots. And in situations where there is external interference, the CAPTAIN receives more packets than the other robots.

After analyzing the results shown above and the results given later on in this chapter, it is concluded that robots listening for multicast packets, or those capturing packets in promiscuous mode, might experience varying results due to the above explained reasons.
In all the four cases discussed above, there is hardly any difference in the performance of the three protocols. Results similar to these made the OU RoboCup team assume that the RoboCup Unicast protocol, and its successor, the RoboCup Multicast protocol would yield good results at the competition. They failed to consider the ADDED_TRAFFIC that was experienced during the competition and its effect on these protocols. The following sections give a detailed description of similar ADDED_TRAFFIC generated using the equipment mentioned in section 4.2. The results obtained for the three protocols in ADDED_TRAFFIC are then compared against each other and a detailed analysis is performed.

In the later phases of analysis, the results observed on Gumby5 are considered for detailed analysis. The results observed on the other two robots are provided in a Table format and are explained in places where a few discrepancies were seen. For Gumby5 though, a detailed analysis of the TOTAL_LOSSES and ELAPSED_TIME observed for each of the three protocols is made by comparing the results in all the cases. The percentage of losses observed and the ELAPSED_TIME experienced are presented in graphical format.

4.5 Performance Of The Protocols In Presence Of ADDED_TRAFFIC Generated By A Single Additional Wireless Network

The experiments described in this section are used to test the performance of all the three protocols in presence of ADDED_TRAFFIC. The previous set of experiments explained in section 4.4, were conducted in ISOLATED conditions. By saying that the network is “ISOLATED”, it is meant that there was neither additional internal traffic generated in the local WLAN (the WLAN to which the BS and the robots were connected to), nor any ADDED_TRAFFIC from external WLANs. The observations made in section 4.4 showed that the number of packet losses were minimal. Thus it is concluded that there was no external traffic which had any substantial effect on our experiments.
A second WLAN is used to generate the ADDED_TRAFFIC mentioned above. Laptops X_PC, Y_PC and AP B are used for this. The BS and robots are still connected to AP A. APs A and B are configured to be on different IP subnetworks. TTCP is used to generate the ADDED_TRAFFIC. The application is run on X_PC to transmit UDP packets of size 1KB at a TTCP_RATE of 1000. Laptop Y_PC is used as the target of the udp packets, and the packets are destined to port 6001 (instead of the default 5001) of the target Linux machine. In all the experiments illustrated in the following subsections, the above external traffic is applied on the network continuously during the experiments by running TTCP.

The experiments are mainly divided into three cases. These cases differ in terms of the Channel on which the AP B operates. The four experiments described in section 4.4 are repeated in each of these cases, but in presence of ADDED_TRAFFIC described above. AP B is run on three different Channels and the results are analyzed in each of the following subsections. Figure 4.4 depicts the experimental setup for the experiments discussed in this section. It should be noted though, that the position of the machines in the figure is not relevant in anyway to the results shown.

4.5.1 Access Points A and B On The Same Channel

In the first set of experiments, both the APs, A and B, are configured to run on the same channel (Channel 2) of the 802.11b spectrum. For each experiment, the results from three trials are presented and the results are analyzed. The results observed in each of the four experiments are as follows:

1. In the first case the TTCP_RATE value is set to 40. The experiments are conducted in the presence of ADDED_TRAFFIC as explained above. The same ADDED_TRAFFIC is applied during the testing of all the three protocols. The TOTAL_LOSSES observed for each of the three protocols is given in Table 4.5. Observe that in all the three trials for the RoboCup Multicast protocol, Gumby5
did better than the other two robots. This phenomenon can be observed consistently in all the experiments. Also observe that the RoboCup Multicast protocol experienced high error rates (approximately 10% to 15%), when compared to the other two protocols. Figures 4.5 and 4.6 depict the ELAPSED\_TIME and Percentage of losses observed for each of the three protocols. Observe that the RoboCup Unicast protocol, similar to the RWMP, has 0% loss rate. But it generates higher ELAPSED\_TIME than the other two. The ELAPSED\_TIME observed would be greater for the robots Gumby6 and Gumby8 as the packets are transmitted in that order. Though, the ELAPSED\_TIME observed in this case is just a few milliseconds and is negligible, the difference would increase as the percentage of losses increases for higher TTCP\_RATEs. The percentage of
Table 4.5 TOTAL LOSSES Observed, TTCP_RATE=40, APs A, B On Channel 2

<table>
<thead>
<tr>
<th>Robot</th>
<th>RoboCup Unicast(5,6,8)</th>
<th>RoboCup Multicast</th>
<th>RWMP(5)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Trial 1</td>
<td>Trial 2</td>
<td>Trial 3</td>
</tr>
<tr>
<td>Gumby5</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Gumby6</td>
<td>0</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>Gumby8</td>
<td>6</td>
<td>0</td>
<td>2</td>
</tr>
</tbody>
</table>

losses was zero for the RoboCup Unicast, as the AP queue was large enough to hold on to the new packets while the old packets were being retransmitted due to the losses.

2. In the second case the TTCP_RATE value is set to 60. The experiments are conducted in the presence of ADDED_TRAFFIC as explained in case 1. The TOTAL_LOSSES observed for each of the three protocols is given in Table 4.6. Observe that the RoboCup Unicast protocol experiences more errors than in the previous case. Also, the Robot which receives the packets last (Gumby8) experiences higher percentage of losses than the one which receives first (Gumby5). The difference in the percentage of losses observed on the robots is analyzed for case 4, when TTCP_RATE is 100. For all the experiments related to the RoboCup Unicast protocol, this phenomenon was observed consistently. This authenticates the statement made in section 3.2.2 that the RoboCup Unicast protocol does not ensure that a packet received by one Robot is also received by all the others robots. The probability that the packet will be received by a Robot varies depending on the order of the robots in which the packets are transmitted by the BS. This can be avoided to a great extent by changing the order of the robots to which the packets are transmitted. This method will decrease the variation among the percentage of losses between the robots but will not reduce the overall packets lost. The order of the robots in which the
Figure 4.5. Percentage Of Losses observed, TTCP\_RATE=40, APs A And B On Channel 2: Results observed at Gumby5. The RWMP and the RoboCup Unicast Protocol observe similar losses and hence the graphs are overlapped packets are transmitted is varied from one experiment to the other in the later sections to emphasize on this point. It should be noted though, that in any single experiment conducted, the order remains same throughout the experiment.

3. In the third case the TTCP\_RATE value is set to 80. The TOTAL LOSSES observed for each of the three protocols is given in Table 4.7. Observe that the TOTAL LOSSES observed in case of the RoboCup Unicast protocol has increased when compared to the previous case.

4. In the fourth case the TTCP\_RATE value is set to 100. The TOTAL LOSSES observed for each of the three protocols is given in Table 4.8.
Figure 4.6. ELAPSED\_TIME Observed, TTCP\_RATE=40, APs A And B On Channel 2: Results observed at Gumby5. The RWMP and the RoboCup Multicast Protocol observed similar ELAPSED\_TIME. Hence the lines green and blue lines are overlapping and not distinguishable.

Table 4.6 TOTAL\_LOSSES Observed, TTCP\_RATE=60, APs A, B On Channel 2

<table>
<thead>
<tr>
<th>Robot</th>
<th>RoboCup Unicast (5,6,8)</th>
<th>RoboCup Multicast</th>
<th>RWMP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Trial 1</td>
<td>Trial 2</td>
<td>Trial 3</td>
</tr>
<tr>
<td>Gumby5</td>
<td>106</td>
<td>1</td>
<td>325</td>
</tr>
<tr>
<td>Gumby6</td>
<td>287</td>
<td>0</td>
<td>780</td>
</tr>
<tr>
<td>Gumby8</td>
<td>542</td>
<td>0</td>
<td>1297</td>
</tr>
</tbody>
</table>
A detailed analysis of the fourth case is given as follows. Figure 4.7 depicts the average ELAPSED_TIME observed at Gumby5 for each of the three protocols. Figure 4.8 depicts the percentage of packet losses at Gumby5 for all the three protocols. The figures are plotted using the values obtained from all the three trials performed. In figure 4.7 observe that the RoboCup Unicast protocol introduces a ELAPSED_TIME of approximately 2 to 3 seconds more than the other two protocols once it transmitted about 400 packets. Such kind of behavior is unacceptable for a real-time application as it introduces high delay in decision making. The RWMP and the RoboCup Multicast protocols differ by about a few milliseconds which is quite acceptable when the percentage of losses is taken into consideration. Figure 4.8 shows that the RWMP performs better than both the RoboCup Unicast and the RoboCup Multicast protocols in terms of percentage of packet losses experienced. The RoboCup Unicast protocol
Figure 4.7. Average ELAPSED TIME Observed, TTCP_RATE=100, APs A And B On Channel 2: Results observed at Gumby5. Only the first 1000 packets of the 5000 are considered. Observe that the RoboCup Unicast protocol had high ELAPSED TIME. The RWMP also experience higher ELAPSED TIME than the RoboCup Multicast Protocol.

does well for the first 500 packets (shown in figure 4.8) before it starts experiencing huge packet errors and experiences error rates of up to 50% by the end of the transmission is reached. The RoboCup Multicast protocol gives a steady performance and yield about 10-15% of packet losses. The RWMP approximately gets all the packets through and at minimal ELAPSED TIME.

By careful analysis of the graphs, it can be concluded that the RWMP yields far better results when compared to the other two protocols. Also it can be observed that both the RWMP and the RoboCup Multicast protocol, produce a steady graph parallel to the X-Axis. This indicates that the performance of the protocols is quiet steady throughout the experiment. Contrary to that, the RoboCup Unicast protocol
Figure 4.8. Percentage Of Losses Observed at Gumby5, TTCP RATE=100, APs A and B on channel 2: The graph is plotted for all three trials. Observe that the RWMP has approximately 0% packet loss. The RoboCup Unicast protocol and the RoboCup Multicast protocol observed approximately 55-75% and 10-15% packet losses respectively.

does not fall into the same category. In figure 4.8 observe that the RoboCup Unicast protocol does well for the first 500 packets. It then starts losing large number of packets as shown by the steep curve in the range of 500 to 1000 packets on the X-axis. And then after the transmission of approximately 1000 packets the curve gets flattened. This phenomenon could be explained by comparing the two graphs 4.7 and 4.8. In figure 4.7, the ELAPSED_TIME curve for the RoboCup Unicast protocol looks more steep at the beginning and then flattens to around 45 degrees angle after approximately 7 seconds. This is approximately the same time when the packet losses are observed in figure 4.8. This is because, the Access Point A, to which the robots are connected to, stores the packets in its output queue and tries to retransmit the
lost packets. As the AP takes care of the packets, it gets successful in retransmitting the packets without errors, but in doing so introduces a huge amount of delay. This happens in the first 7 to 8 seconds period of the experiment, where we observe a huge increase in the ELAPSED\_TIME, but still got 100% throughput. However, once the AP exceeds its maximum capacity of the queue, it starts dropping the packets and attempts lesser number of retransmissions. Because of this packet drops, the Robot starts experiencing packet losses which increase as the time increases.

Figure 4.9 gives the percentage of losses observed for the RoboCup Unicast protocol on all the three robots for all the three trials. As indicated in Table 4.8, the BS transmitted packets to Gumby6, Gumby8 and Gumby5 in that order (as indicated in the Table by RoboCup Unicast (6,8,5)). Observe that the percentage of losses observed increases from Gumby6 to Gumby8, and then from Gumby8 to Gumby5. This validates the point that, for the RoboCup Unicast protocol, the order of the destination robots effects the amount of losses observed on them. The Robot that receives the packet first experiences few losses than the one which receives last. In this case, the final percentage of losses observed at Gumby6, Gumby8 and Gumby5 are approximately 25%, 50% and 65% respectively (taking the average of the three trials). Considering the above trend, the percentage of losses would increase further for the fourth and fifth robots, had they been used. This pattern is observed in all the experiments conducted and is one of the major drawbacks of the RoboCup Unicast protocol at the competition.

From the four cases discussed in this section, we observed that the Robust Wireless Multicast Protocol does extremely well, both in terms of percentage of losses observed as well as the ELAPSED\_TIME observed at the destination. The RoboCup Multicast Protocol also does extremely well in terms of avoiding ELAPSED\_TIME at the receiver end. But it experiences error rates of around 10 to 15% which is not acceptable. The RoboCup Unicast Protocol fails miserably both in terms of percentage of packet losses experienced and the high delay that it introduces. This phenomenon
Figure 4.9. Percentage Of Losses Observed For The RoboCup Unicast, TTCP RATE=100, APs A And B On Channel 2: Observe that the percentage of losses increases as we move from Gumby6 to Gumby8 to Gumby5, in that order for the RoboCup Unicast protocol alone.

holds true for the experiments explained in the following sections. The same argument holds for all the experiments followed. Any additional variations seen are pointed out along the graphs and tables given.

4.5.2 Access Points Placed On Adjacent Channels

In the second set of experiments, the APs are configured to run on adjacent channels instead of same channel. AP A is unchanged and allowed to run on Channel 2. AP B is configured to run on Channel 3 of the 802.11b spectrum. The experiments conducted are similar to that performed in section 4.5.1. The results observed in each of the four experiments are as follows:
Table 4.9 TOTAL LOSSES Observed, TTCP RATE=40, APs A, B On Channels 2 And 3 Respectively.

<table>
<thead>
<tr>
<th>Robot</th>
<th>RoboCup Unicast(5,6,8)</th>
<th>RoboCup Multicast</th>
<th>RWMP(6)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Trial 1</td>
<td>Trial 2</td>
<td>Trial 3</td>
</tr>
<tr>
<td>Gumby5</td>
<td>718</td>
<td>12</td>
<td>53</td>
</tr>
<tr>
<td>Gumby6</td>
<td>1936</td>
<td>14</td>
<td>133</td>
</tr>
<tr>
<td>Gumby8</td>
<td>2883</td>
<td>14</td>
<td>296</td>
</tr>
</tbody>
</table>

Table 4.10 TOTAL LOSSES Observed, TTCP RATE=60, APs A, B On Channels 2 And 3 Respectively.

<table>
<thead>
<tr>
<th>Robot</th>
<th>RoboCup Unicast(8,5,6)</th>
<th>RoboCup Multicast</th>
<th>RWMP(8)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Trial 1</td>
<td>Trial 2</td>
<td>Trial 3</td>
</tr>
<tr>
<td>Gumby5</td>
<td>2202</td>
<td>2119</td>
<td>2098</td>
</tr>
<tr>
<td>Gumby6</td>
<td>3158</td>
<td>3151</td>
<td>3117</td>
</tr>
<tr>
<td>Gumby8</td>
<td>748</td>
<td>779</td>
<td>778</td>
</tr>
</tbody>
</table>

1. In the first case the TTCP_RATE value is set to 40. The ADDED TRAFFIC generated is same as explained in section 4.5, and is kept constant for all the three protocols. The TOTAL LOSSES observed for each of the three protocols is given in Table 4.9.

2. In the second case the TTCP_RATE value is set to 60. The experiments are conducted in the presence of ADDED TRAFFIC as explained in case 1. The TOTAL LOSSES observed for each of the three protocols is given in Table 4.10.

3. In the third case the TTCP_RATE value is set to 80. The TOTAL LOSSES observed for each of the three protocols is given in Table 4.11.
Table 4.11 TOTAL LOSSES Observed, TTCP RATE=80, APs A, B On Channels 2 And 3 Respectively.

<table>
<thead>
<tr>
<th>Robot</th>
<th>RoboCup Unicast(5,6,8)</th>
<th>RoboCup Multicast</th>
<th>RWMP(6)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Trial 1</td>
<td>Trial 2</td>
<td>Trial 3</td>
</tr>
<tr>
<td>Gumby5</td>
<td>1702</td>
<td>1483</td>
<td>1551</td>
</tr>
<tr>
<td>Gumby6</td>
<td>3541</td>
<td>3763</td>
<td>3756</td>
</tr>
<tr>
<td>Gumby8</td>
<td>4002</td>
<td>4237</td>
<td>4289</td>
</tr>
</tbody>
</table>

Table 4.12 TOTAL LOSSES Observed, TTCP RATE=100, APs A, B On Channels 2 And 3 Respectively.

<table>
<thead>
<tr>
<th>Robot</th>
<th>RoboCup Unicast(5,6,8)</th>
<th>RoboCup Multicast</th>
<th>RWMP(8)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Trial 1</td>
<td>Trial 2</td>
<td>Trial 3</td>
</tr>
<tr>
<td>Gumby5</td>
<td>1673</td>
<td>2382</td>
<td>1842</td>
</tr>
<tr>
<td>Gumby6</td>
<td>3207</td>
<td>4188</td>
<td>3956</td>
</tr>
<tr>
<td>Gumby8</td>
<td>3864</td>
<td>4352</td>
<td>4232</td>
</tr>
</tbody>
</table>

4. In the fourth case the TTCP RATE value is set to 100. The TOTAL LOSSES observed for each of the three protocols is given in Table 4.12. Figures 4.10 and 4.11 compare the ELAPSED TIME and percentage of losses observed in all the three protocols respectively.

4.5.3 Access Points A And B Placed One Channel Apart

In the third set of experiments, the Access Points are configured to run on alternate channels. The first AP is configured to run on Channel 2 and the second AP is configured to run on Channel 4 of the 802.11b spectrum. The experiments conducted are similar to that performed in section 4.5.1. The results observed in each of the four experiments are as follows:
Figure 4.10. Average ELAPSED TIME Observed, TTCP RATE=100, APs A And B On Channels 2 And 3 Respectively: Only first 2000 packets of the 5000 are considered. Observe that the RoboCup Unicast protocol faced huge packet delay. The RWMP also introduced higher ELAPSED TIME compared to the RoboCup Multicast.

1. In the first case the TTCP RATE value is set to 40. The ADDED TRAFFIC applied is the same as one explained at the start of section 4.5 and is kept constant for all the three protocols. The TOTAL LOSSES observed for each of the three protocols is given in Tables 4.13.

2. In the second case the TTCP RATE value is set to 60. The experiments are conducted in the presence of ADDED TRAFFIC as explained in case 1. The TOTAL LOSSES observed for each of the three protocols is given in Table 4.14.
Figure 4.11. Percentage of Losses Observed, TTCP_RATE=100, APs A And B On Channels 2 And 3 Respectively : Observe that the RWMP experiences approximately 2-3% packet loss. The RoboCup Unicast yields 35 to 45% packet losses and the RoboCup Multicast gives approximately 12 to 15% packet losses.

Table 4.13 TOTAL_LOSSES Observed, TTCP_RATE=40, APs A And B On Channels 2 And 4 Respectively.

<table>
<thead>
<tr>
<th>Machine Name</th>
<th>RoboCup Unicast(5,6,8)</th>
<th>RoboCup Multicast</th>
<th>RWMP(6)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gumby5</td>
<td>9</td>
<td>23</td>
<td>3</td>
</tr>
<tr>
<td>Gumby6</td>
<td>127</td>
<td>869</td>
<td>194</td>
</tr>
<tr>
<td>Gumby8</td>
<td>302</td>
<td>1059</td>
<td>1228</td>
</tr>
</tbody>
</table>
Table 4.14 TOTAL LOSSES Observed, TTCP RATE=60, APs A And B On Channels 2 And 4 Respectively.

<table>
<thead>
<tr>
<th>Machine Name</th>
<th>RoboCup Unicast(8,5,6)</th>
<th>RoboCup Multicast</th>
<th>RWMP(8)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gumby5</td>
<td>0</td>
<td>14</td>
<td>0</td>
</tr>
<tr>
<td>Gumby6</td>
<td>156</td>
<td>748</td>
<td>405</td>
</tr>
<tr>
<td>Gumby8</td>
<td>289</td>
<td>755</td>
<td>142</td>
</tr>
</tbody>
</table>

Table 4.15 TOTAL LOSSES Observed, TTCP RATE=80, APs A And B On Channels 2 And 4 Respectively.

<table>
<thead>
<tr>
<th>Machine Name</th>
<th>RoboCup Unicast(6,8,5)</th>
<th>RoboCup Multicast</th>
<th>RWMP(6)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gumby5</td>
<td>0</td>
<td>51</td>
<td>0</td>
</tr>
<tr>
<td>Gumby6</td>
<td>51</td>
<td>927</td>
<td>153</td>
</tr>
<tr>
<td>Gumby8</td>
<td>123</td>
<td>1270</td>
<td>1174</td>
</tr>
</tbody>
</table>

3. In the third case the TTCP RATE value is set to 80. The TOTAL LOSSES observed for each of the three protocols is given in Table 4.15.

4. In the fourth case the TTCP RATE value is set to 100. The TOTAL LOSSES observed for each of the three protocols is given in Table 4.16.

Comparing the results from sections 4.5.1, 4.5.2 and 4.5.3, it can be concluded that as the AP B moves more than one channel away from AP A, the TOTAL LOSSES observed reduces considerably. It should be noted that, the percentage of losses observed when AP B operated in channel 5 were approximately 0% (see appendix A). This clearly indicates that the losses observed before were due to high collisions between the packets transmitted by the BS and the Laptop X PC, when operating
Table 4.16 TOTAL_LOSSES Observed, TTCP_RATE=100, APs A And B on Channels 2 And 4 Respectively.

<table>
<thead>
<tr>
<th>Machine Name</th>
<th>RoboCup Unicast(5,6,8)</th>
<th>RoboCup Multicast</th>
<th>RWMP(8)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gumby5</td>
<td>286</td>
<td>46</td>
<td>10</td>
</tr>
<tr>
<td>Gumby6</td>
<td>473</td>
<td>883</td>
<td>695</td>
</tr>
<tr>
<td>Gumby8</td>
<td>718</td>
<td>1168</td>
<td>97</td>
</tr>
</tbody>
</table>

in overlapping channels. These collisions cause heavy packet losses. In case of the RoboCup Unicast protocol, these packet losses triggered huge packet retransmissions, which ultimately resulted in increased losses of more than 50% and also added packet delay of approximately 2 to 3 seconds. The RoboCup Multicast observed losses of approximately 15% in the previous two cases. The loss rate can be further increased by introducing more collisions using additional external devices. Section 4.6 tries to generate higher collision rates by adding AP C into the network and generating additional ADDED.TRAFFIC.

4.6 Performance Of The Protocols In Presence Of ADDED.TRAFFIC Generated By Two Additional Wireless Networks

For the last set of experiments all three APs are used. The setup on the robots, BS and AP A is left unchanged. Laptop X_PC is connected to AP B and Laptop Y_PC is connected to AP C. All the machines involved were in vicinity of each other. All the three APs are configured to be on different IP subnetworks. As in the previous cases, TTCP used to generate the ADDED_TRAFFIC. However, TTCP is now run on both X_PC and Y_PC to transmit udp packets of size 1KB at 1000 packets per second.

APs B and C are connected to the same wired network. Two machines named Indigo and Elephus are also connected to the same wired network as that of the APs
B and C. *Indigo* is used as the target for X_{PC} and *Elephus* is used as the target for Y_{PC} for their respective UDP packet transmissions. The packets generated by X_{PC} and Y_{PC} are destined to port 6001 of their target machines. In all the experiments illustrated in the following subsections, the above overhead is applied on the network continuously during the experiments by running TTCP.

Figure 4.12 depicts the experimental setup for the experiments discussed in this section. It should be noted though, that the position of the machines in the figure is not relevant to the results shown.

![Experiment Setup With Three APs](image)

Figure 4.12. Experiment Setup With Three APs: Note that the BSAs of APs A, B And C, indicated by the dotted circles, are imaginary and have no effect on the results.
Table 4.17 TOTAL LOSSES Observed, TTCP_RATE=40, APs A, B And C On Channel 2.

<table>
<thead>
<tr>
<th>Robot</th>
<th>RoboCup Unicast(5,6,8)</th>
<th>RoboCup Multicast</th>
<th>RWMP(5)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gumby5</td>
<td>0</td>
<td>407</td>
<td>0</td>
</tr>
<tr>
<td>Gumby6</td>
<td>0</td>
<td>818</td>
<td>54</td>
</tr>
<tr>
<td>Gumby8</td>
<td>2</td>
<td>1317</td>
<td>573</td>
</tr>
</tbody>
</table>

The experiments are mainly divided into three subsections. The experiments from any two subsections differ in terms of the Channel in which the APs B and C operate in. AP A always runs in Channel 2 of the 802.11b spectrum for all the experiments. Sections 4.6.1, 4.6.2, and 4.6.3 describe the experiments conducted in each of the three situations.

4.6.1 All Three APs On The Same Channel

In the first set of experiments all the three APs A, B and C are configured to run on the same channel (Channel 2) of the 802.11b spectrum. The experiments conducted are similar to those performed in section 4.4 but in presence of ADDED_TRAFFIC explained before. The results observed in each of the four experiments are as follows:

1. In the first case the TTCP_RATE value is set to 40. The experiments are conducted in the presence of ADDED_TRAFFIC as explained above. This ADDED_TRAFFIC is applied during the testing of all the three protocols. The TOTAL_LOSSES observed for each of the three protocols is given in Table 4.17.

2. In the second case the TTCP_RATE value is set to 60. The experiments are conducted in the presence of ADDED_TRAFFIC as explained in case 1. The TOTAL_LOSSES observed for each of the three protocols is given in Table 4.18.
Table 4.18 TOTAL LOSSES Observed, TTCP RATE=60, APs A, B And C On Channel 2.

<table>
<thead>
<tr>
<th>Robot</th>
<th>RoboCup Unicast(6,8,5)</th>
<th>RoboCup Multicast</th>
<th>RWMP(6)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gumby5</td>
<td>1942</td>
<td>616</td>
<td>9</td>
</tr>
<tr>
<td>Gumby6</td>
<td>376</td>
<td>850</td>
<td>1</td>
</tr>
<tr>
<td>Gumby8</td>
<td>1112</td>
<td>1236</td>
<td>138</td>
</tr>
</tbody>
</table>

Table 4.19 TOTAL LOSSES Observed, TTCP RATE=80, APs A, B And C On Channel 2.

<table>
<thead>
<tr>
<th>Robot</th>
<th>RoboCup Unicast(8,5,6)</th>
<th>RoboCup Multicast</th>
<th>RWMP(8)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gumby5</td>
<td>1662</td>
<td>473</td>
<td>4</td>
</tr>
<tr>
<td>Gumby6</td>
<td>2261</td>
<td>660</td>
<td>12</td>
</tr>
<tr>
<td>Gumby8</td>
<td>524</td>
<td>1634</td>
<td>3</td>
</tr>
</tbody>
</table>

3. In the third case the TTCP RATE value is set to 80. The TOTAL LOSSES observed for each of the three protocols is given in Table 4.19.

4. In the fourth case the TTCP RATE value is set to 100. The TOTAL LOSSES observed for each of the three protocols is given in Table 4.20. Figures 4.13 and 4.14 compare the three protocols for the ELAPSED TIME and percentage of losses observed at Gumby5 respectively.

4.6.2 Access Points B And C Placed On Adjacent Channel

In the second set of experiments involving all the three APs, B and C are configured to run on channel 3. AP A is still configured to run on Channel 2 of the 802.11b spectrum. The experiments conducted are similar to that performed in section 4.6.1.
Table 4.20 TOTAL_LOSSES Observed, TTCP_RATE=100, APs A, B And C On Channel 2.

<table>
<thead>
<tr>
<th>Robot</th>
<th>RoboCup Unicast(5,6,8)</th>
<th>RoboCup Multicast</th>
<th>RWMP(5)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gumby5</td>
<td>689</td>
<td>620</td>
<td>1</td>
</tr>
<tr>
<td>Gumby6</td>
<td>1241</td>
<td>772</td>
<td>186</td>
</tr>
<tr>
<td>Gumby8</td>
<td>2302</td>
<td>1030</td>
<td>411</td>
</tr>
</tbody>
</table>

Figure 4.13. ELAPSED_TIME Observed, TTCP_RATE=100, APs A, B And C On Channel 2: Only first 1000 packets of the 5000 are considered. Observe that the RoboCup Unicast protocol faced huge packet delay. The RWMP also had slight delay compared to the RoboCup Multicast but it is far less when compared to that of the RoboCup Unicast.
Figure 4.14. Percentage Of Losses Observed, TTCP_RATE=100, APs A, B And C On Channel 2: Observe that the RWMP has 0% packet loss (line is along the X-Axis and not visible). The RoboCup Unicast protocol yields approximately 14% packet lost. The RoboCup Multicast also experiences similar losses of approximately 12 to 13%. However for the RoboCup Unicast, the percentage of packet loss varies between robots and is approximately 45% for Gumby8 (given in Table 4.20).

The only difference is that the experiments are conducted for only two values of TTCP_RATE, 40 and 60. The experiments with TTCP_RATE values 80 and 100 were not considered as the RoboCup Unicast protocol experienced 90% packet losses when TTCP_RATE was increased to 60. The results observed in the two cases are as follows:

1. In the first case the TTCP_RATE value is set to 40. The ADDED_TRAFFIC generated is the same as explained at the start of the section 4.6, and is kept constant for all the three protocols. The TOTAL_LOSSES observed for each of the three protocols is given in Table 4.21. Figures 4.15 and 4.16 compare
Table 4.21 TOTAL LOSSES Observed, TTCP RATE=40, AP A On Channel 2, B And C On Channel 3.

<table>
<thead>
<tr>
<th>Robot</th>
<th>RoboCup Unicast(5,6,8)</th>
<th>RoboCup Multicast</th>
<th>RWMP(5)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Trial 1</td>
<td>Trial 2</td>
<td>Trial 3</td>
</tr>
<tr>
<td>Gumby5</td>
<td>1322</td>
<td>1264</td>
<td>1581</td>
</tr>
<tr>
<td>Gumby6</td>
<td>3115</td>
<td>3060</td>
<td>3291</td>
</tr>
<tr>
<td>Gumby8</td>
<td>4429</td>
<td>4325</td>
<td>4530</td>
</tr>
</tbody>
</table>

Table 4.22 TOTAL LOSSES Observed, TTCP RATE=60, AP A On Channel 2, B And C On Channel 3.

<table>
<thead>
<tr>
<th>Robot</th>
<th>RoboCup Unicast(6,8,5)</th>
<th>RoboCup Multicast</th>
<th>RWMP(6)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Trial 1</td>
<td>Trial 2</td>
<td>Trial 3</td>
</tr>
<tr>
<td>Gumby5</td>
<td>4568</td>
<td>4538</td>
<td>4564</td>
</tr>
<tr>
<td>Gumby6</td>
<td>2168</td>
<td>2670</td>
<td>1738</td>
</tr>
<tr>
<td>Gumby8</td>
<td>3958</td>
<td>4239</td>
<td>4009</td>
</tr>
</tbody>
</table>

the three protocols for the ELAPSED TIME and percentage of losses observed respectively.

2. In the second case the TTCP RATE value is set to 60. The experiments are conducted in the presence of ADDED TRAFFIC as explained in case 1. The TOTAL LOSSES observed for each of the three protocols is given in Table 4.22. Figures 4.17 and 4.18 compare the three protocols for the packet ELAPSED TIME and percentage of losses observed respectively.

4.6.3 APs B And C Are Placed On Channels 1 And 3 Respectively

In the last set of experiments, AP B and AP C are configured to run on channels 1 and 3 respectively. AP A is still configured to run On Channel 2 of the 802.11b
Figure 4.15. Average ELAPSED_TIME Observed, TTCP_RATE=40; AP A On Channel 2, B And C On Channel 3: Only first 1000 packets of the 5000 are considered. Observe that the RoboCup Unicast protocol faced huge packet delay when compared to the other two protocols.

The experiments conducted are similar to that performed in section 4.6.2. The results observed in each of the four experiments are as follows:

1. In the first case the TTCP_RATE value is set to 40. The ADDED_TRAFFIC generated is the same as in the previous two subsections. This ADDED_TRAFFIC is kept constant for all the three protocols. The TOTAL_LOSSES observed for each of the three protocols is given in Tables 4.23. Figures 4.19 and 4.20 compare the three protocols for the ELAPSED_TIME and percentage of losses observed respectively.

2. In the second case the TTCP_RATE value is set to 60. The experiments are conducted in the presence of ADDED_TRAFFIC as explained in case 1.
Figure 4.16. Percentage of Losses Observed, TTCP RATE=40, AP A On Channel 2, B And C On Channel 3: Observe that the RWMP has approximately 0% packet loss. The RoboCup Unicast yields approximately 25% packet loss which is similar to that of the RoboCup Multicast. However for the RoboCup Unicast, the percentage of packet loss varies between robots and is approximately 90% for Gumby8.

Table 4.23 TOTAL LOSSES Observed, TTCP RATE=40, APs A, B And C On Channels 2, 1 And 3 Respectively.

<table>
<thead>
<tr>
<th>Robot</th>
<th>RoboCup Unicast (5,6,8)</th>
<th>RoboCup Multicast</th>
<th>RWMP (5)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Trial 1</td>
<td>Trial 2</td>
<td>Trial 3</td>
</tr>
<tr>
<td>Gumby5</td>
<td>907</td>
<td>1108</td>
<td>1358</td>
</tr>
<tr>
<td>Gumby6</td>
<td>2939</td>
<td>3024</td>
<td>3417</td>
</tr>
<tr>
<td>Gumby8</td>
<td>4356</td>
<td>4447</td>
<td>4588</td>
</tr>
</tbody>
</table>
Figure 4.17. Average ELAPSED\_TIME Observed, TTCP\_RATE=60, AP A On Channel 2, B And C On Channel 3: Only first 1000 packets of the 5000 are considered. Observe that the RoboCup Unicast protocol faced huge packet delay. The RWMP also introduced packet delay but it is far less compared to that of the RoboCup Unicast.

The TOTAL\_LOSSES Observed for each of the three protocols is given in Table 4.24. Figures 4.21 and 4.22 compare the three protocols for the packet ELAPSED\_TIME and percentage of losses observed respectively.

### 4.7 Queue Model Simulation

This section describes the expected behavior of the protocols by simulating the queue model and average transmission time for packet transfer over the wireless network.

The queue size of AP A, which is connected to the BS, is considered as 400. The TTCP\_RATE is considered to be 100. Therefore, the packets arrive at the AP once
Figure 4.18. Percentage of Losses Observed, TTCP_RATE=60, AP A On Channel 2, B And C On Channel 3: Observe that the RWMP has approximately 2 to 3% packet loss. The RoboCup Unicast yields heavy packet losses of approximately 90% and the RoboCup Multicast experiences approximately 20 to 30% packet losses.

Table 4.24 TOTAL_LOSSES Observed, TTCP_RATE=60, APs A, B And C On Channels 2, 1 And 3 Respectively.

<table>
<thead>
<tr>
<th>Robot</th>
<th>RoboCup Unicast (6,8,5)</th>
<th>RoboCup Multicast</th>
<th>RWMP (5)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Trial 1</td>
<td>Trial 2</td>
<td>Trial 3</td>
</tr>
<tr>
<td>Gumby5</td>
<td>4629</td>
<td>4532</td>
<td>4616</td>
</tr>
<tr>
<td>Gumby6</td>
<td>2348</td>
<td>2063</td>
<td>1764</td>
</tr>
<tr>
<td>Gumby8</td>
<td>4040</td>
<td>4192</td>
<td>3925</td>
</tr>
</tbody>
</table>
Figure 4.19. Average ELAPSED TIME Observed, TTCP RATE=40, APs A, B And C On Channels 2, 1 And 3 Respectively: Only first 1000 packets of the 5000 are considered. Observe that the RoboCup Unicast protocol faced huge packet delay.

every 10000 microseconds. The probability of packet loss is taken as 30% (this is the maximum loss observed for the RoboCup Multicast Protocol).

The time taken to transmit a packet successfully over a 1Mbps 802.11b wireless network ($T_t$) is calculated using the below equation [3]:

$$T_t = 2(P_r + Hdr) + ((len + UDP_h + IP_h + Ack) * 8 * 10^6 / r) + SIFS + DIFS + (B_o * s_t / 2)$$

where,

$P_r$ is the time to transmit the preamble before transmitting the actual data (72 microseconds)

$Hdr$ is the time to transmit the 802.11 header (24 microseconds)

$len$ is the length of packet being transmitted (250 bytes)
Figure 4.20. Percentage of Losses Observed, TTCP RATE=40, APs A, B And C On Channels 2, 1 And 3 Respectively: Observe that the RWMP has approximately 0 to 1% packet loss. The RoboCup Unicast yields packet losses of approximately 20 to 25% for Gumby5 and even more for Gumby6 and Gumby8 which come later in the order. The RoboCup Multicast protocol experiences packet losses of around 20%.

- $UDP_h$ is the length of the UDP header (8 bytes)
- $IP_h$ is the length of the IP header (20 bytes)
- $r$ is the speed of the 802.11 wireless network (1 Mbps)
- $SIFS$ is the Short Inter Frame Space mentioned in section 2.4.3.1 (10 microseconds)
- $Ack$ is the length of the 802.11 Acknowledgement (14 bytes)
- $DIFS$ is the Distributed Inter Frame Space mentioned in section 2.4.3.1 (50 microseconds)
- $B_o$ is the random backoff mentioned in section 2.4.4 (32 bytes)
- $s_t$ is the slotTime mentioned in section 2.4.4 (20 microseconds)
Figure 4.21. Average ELAPSED\_TIME Observed, TTCP\_RATE=60, APs A, B And C On Channels 2, 1 And 3 Respectively: Only first 1000 packets of the 5000 are considered. Observe that the RoboCup Unicast protocol faced huge packet delay.

From the above equation and values, the time taken to transmit 250 bytes successfully over a 1 Mbps 802.11b wireless network is calculated to be 2799.9 microseconds. Note that a packet is said to be transmitted successfully if it receives an error free acknowledgement. Assuming a 30\% loss probability and the retransmission limit to be 5, the average transmission time for the same 250 bytes is calculated to be 3996.7 microseconds.

Figure 4.23 gives the estimated ELAPSED\_TIME for the RoboCup Unicast protocol and the RWMP protocol for the above network model. The queue size of the AP A is considered to be 400 and total packets transmitted are 5000 for the RWMP and 15000 for the RoboCup Unicast.
Figure 4.22. Percentage of Losses Observed, TTCP RATE=60, APs A, B and C On Channels 2, 1 and 3 Respectively: Observe that the RWMP has approximately 0% packet loss. The RoboCup Unicast yields packet losses of approximately 90%, though this value decreases for Gumby6 and Gumby8. The RoboCup Multicast experiences losses of approximately 20%.

As given in figure 4.23, the RoboCup Unicast protocol observes a steady increase in the ELAPSED_TIME for the first 800 packets (approximately). And then the curve bends back to a 45 degree angle parallel to the RWMP curve indicating that the increase in the ELAPSED_TIME has stopped. This is because, the queue of the AP A gets filled as the AP receives packets at a rate faster than it can transmit. Observe that for the RoboCup Unicast protocol, the AP receives 3 packets every 10000 microseconds, whereas it takes 11990.3 microseconds to transmit them out. Thus, for every 3 packets transmitted by the BS, the AP gets behind by approximately 1990.3 microseconds. This time gap increases as the packets queue up. But once the queue of the AP gets filled, the amount of delay being introduced reaches to a steady state.
Figure 4.23. Estimated ELAPSED\_TIME, TTCP\_RATE=100, Loss Probability = 30% : Observe that the RoboCup Unicast protocol experienced a delay of approximately 2 seconds when compared to the RWMP. (only first 1500 packets of the 5000 are shown in the graph)

We observed similar results in all our graphs depicting the ELAPSED\_TIME for the RoboCup Unicast protocol. The reason for such a behavior of all the graphs for ELAPSED\_TIME for the RoboCup Unicast protocol is clearly explained by this simulation model.

4.8 Experiment Spanning Over Longer Time

This section is similar to the one explained in section 4.5.2. The same experiment setup is used as in the previous case. This experiment is conducted to show that the Robust Wireless Multicast Protocol (RWMP) experiences minimal losses even when the experiment is run for longer time period. In the previous case, only 5000 packets were transmitted by the BS. When the TTCP\_RATE is 100, the experiment spanned
Table 4.25 TOTAL LOSSES Observed On CAPTAIN for the RWMP, TTCP_RATE=100, APs A, B On channels 2, 3 Respectively.

<table>
<thead>
<tr>
<th></th>
<th>Chunk 1</th>
<th>Chunk 2</th>
<th>Chunk 3</th>
<th>Chunk 4</th>
<th>Chunk 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trial 1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>7</td>
<td>0</td>
</tr>
<tr>
<td>Trial 2</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Trial 3</td>
<td>1</td>
<td>0</td>
<td>3</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Average</td>
<td>1.6</td>
<td>0.3</td>
<td>1.3</td>
<td>2.6</td>
<td>0.6</td>
</tr>
</tbody>
</table>

for only 50 seconds. This experiment is to observe the behavior of the RWMP over longer periods of time.

The total packets captured at the Robot is still 5000. However, the packets are captured during various stages of the experiment. The BS transmits packets continuously at 100 packets every second. The first 1000 packets are captured as in the previous case. This group of first 1000 packets is identified as “Chunk 1”. The second chunk of 1000 packets is captured 5 minutes after the arrival of the 1000th packet. The third chunk of 1000 packets is captured 5 minutes after capturing the second chunk of 1000 packets and so on. A total of 5 chunks, each of 1000 packets, are captured spanning over approximately 21 minutes (including the time to transfer the 5000 packets captured).

Only two robots were used for this experiment. Tables 4.25, 4.26 give the TOTAL LOSSES observed on both the CAPTAIN and the Robot listening in promiscuous mode respectively. The percentage of losses observed on both the robots are given in figure 4.24 and figure 4.25. The graphs are plotted using the average of the values obtained from the three trials.

In figure 4.24, the average percentage of losses varied from 0 to 0.5% (5 out of 1000 packets). The variation in the number of packets lost is very less and hence it
Table 4.26 TOTAL LOSSES Observed On Non-CAPTAIN for the RWMP, TTCP_RATE=100, APs A, B On Channels 2, 3 Respectively.

<table>
<thead>
<tr>
<th></th>
<th>Chunk 1</th>
<th>Chunk 2</th>
<th>Chunk 3</th>
<th>Chunk 4</th>
<th>Chunk 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trial 1</td>
<td>7</td>
<td>5</td>
<td>17</td>
<td>12</td>
<td>21</td>
</tr>
<tr>
<td>Trial 2</td>
<td>4</td>
<td>5</td>
<td>14</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>Trial 3</td>
<td>21</td>
<td>7</td>
<td>13</td>
<td>15</td>
<td>32</td>
</tr>
<tr>
<td>Average</td>
<td>10.6</td>
<td>5.6</td>
<td>14.6</td>
<td>11.3</td>
<td>20.3</td>
</tr>
</tbody>
</table>

Figure 4.24. Average Percentage of Losses Observed On CAPTAIN, TTCP_RATE=100, APs A On Channel 2 And B On Channel 3 : Observe that the percentage of losses range from 0 to 0.5% for all the five chunks indicating that the performance is same throughout the experiment.
Figure 4.25. Average Percentage of Losses Observed On non-CAPTAIN, TTCP_RATE=100, APs A On Channel 2 And B On Channel 3: Observe that the percentage of losses vary from 1 to 2% for all the five chunks indicating that the percentage of loss is not high even after longer time periods.

can be concluded that the CAPTAIN experiences constant and negligible losses even at the later stages of the experiment. The percentage of losses for the non-CAPTAIN, as given in figure 4.25, is slightly higher and varies from 1 to 2%. However, even this loss is very less when compared to the percentage of losses observed by the RoboCup Unicast and the RoboCup Multicast protocols in earlier experiments.

Thus from the above results, it is concluded that the Robust Wireless Multicast protocol performs well even for experiments spanning over longer time periods.
5. Conclusion

In this chapter, a summary of the RoboCup competition and the two protocols, the RoboCup Unicast and the RoboCup Multicast, used by the Ohio University (OU) RoboCup team is presented. A brief overview of the proposed the Robust Wireless Multicast Protocol (RWMP) is then presented, followed by the advantages and limitations of the RWMP and some recommendations for future work.

5.1 Summary

The RoboCup is an international soccer competition between teams comprising of five autonomous mobile robots. Each team communicates with its robots using a dedicated Access Point (AP) and a Base Station (BS). The BS interprets the position of the robots using a camera placed on top of the field. It uses this information to send strategic information in the form of packet updates of approximately 250 bytes to its team of robots over its Wireless LAN (WLAN). Many teams share the 2.4GHz - 2.4835GHz frequency spectrum, which is also used by the 802.11b standard.

The extreme network conditions observed at the competition, combined with the characteristics of 802.11b network devices, caused huge error rates during the competition. The RoboCup Unicast protocol sends packet updates to every Robot individually. Due to this over utilization of the available bandwidth, the RoboCup Unicast protocol did not yield good results at the competition. As an improvement, the RoboCup Multicast protocol was used to send packet updates. The RoboCup Multicast protocol sends a single packet to a multicast address and makes all the robots listen to packets destined to this multicast address. Though the RoboCup
Multicast protocol reduced the amount of overhead by five times, it did not yield satisfactory performance due to the lack of reliability and robustness.

Two factors, ELAPSED TIME and TOTAL LOSSES, were considered while measuring the performance of the protocols mentioned. The behavior of the RoboCup Unicast and the RoboCup Multicast protocols was studied using the 802.11b standard to figure out the exact reasons for the failure of the protocols. Considering these issues, a new protocol Robust Wireless Multicast Protocol (RWMP) was designed and implemented. RWMP sends each packet to one of the robots (CAPTAIN), and all of the other robots capture the packets by listening in promiscuous mode. The solution tries to reduce network overhead, ensure reliability, avoid packet delay, avoid redundant re-transmissions of old data at the expense of new data, transmit minimal packets and still be able to send data to all the robots and give precedence to current packet over obsolete packet.

Several experiments were conducted to test the performance of the protocols, and a detailed analysis of the results was performed. The experiments differed in terms of the rate at which packet updates were sent by the BS and network interference generated (similar to that observed at the competition).

5.2 Advantages And Disadvantages

In all of the experiments conducted, the RWMP experienced losses varying from 2% to 3%. The RoboCup Unicast protocol observed losses varying from 10% to 90% depending on various factors such as the rate at which the packets were sent, the order of robots and the interference generated. The RoboCup Multicast protocol observed losses varying from 15% to 25% due to the lack of reliability involving transmission of multicast packets.

Moreover, the RoboCup Unicast protocol experienced high ELAPSED TIME. It introduced a delay of few seconds more than the RoboCup Multicast protocol, within the transmission time of a few hundred packet updates. The delay observed increased
as the time increased, and such behavior is unacceptable in a real time environment like that of the RoboCup. The RWMP experienced an increase of a few milliseconds in its ELAPSED_TIME when compared to the RoboCup Multicast protocol. This delay in packet transmission was minimal and was acceptable given the small number of losses experienced by Robust Wireless Multicast protocol when compared to the other two protocols.

RWMP has the limitation that it only works when all the machines are in vicinity of each other. Also the robots listening in promiscuous mode tended to lose a few packets more than the CAPTAIN. This is because of difference in noise experienced by the robots and the lack of retransmissions for packets lost by the robots listening in promiscuous mode. The effect of this could be largely minimized by randomly selecting the CAPTAIN for each packet transmission.

5.3 Future Work

The RWMP makes the basic assumption that all of the wireless devices within the vicinity of the AP can receive all the packets transmitted by the AP. It would be interesting to see the behavior of the protocol in situations where the mobile devices move in and out of the visibility range of the AP. The protocol could be made more robust in such situations by sending updates to more robots rather than sending them to just the CAPTAIN. Selecting different channels when experiencing heavy packet losses on channels could also be an alternative.

The protocol is based on the assumption that all the packets sent are of a fixed size of 250 bytes. It would be interesting to see the results when the packet sizes are increased, which would result in MAC layer fragmentation. This thesis does not consider the fragmentation technique and the use of the RequestToSend and ClearToSend packets. It would be interesting to study the effect of these factors on the three protocols.
It would also be interesting to see how the protocols work using the 802.11a wireless standard instead of the 802.11b standard. The 802.11a standard provides a greater number of channels, with each channel having a higher bandwidth than the 802.11b standard and also operating at a higher frequency range (5 GHz).
BIBLIOGRAPHY


APPENDIX
A. Results When Access Points A And B Are Placed 2 Channels Apart

This appendix gives the values skipped in chapter 4.

In the second set of experiments involving the study of the performance of the protocols in added ADDED_TRAFFIC, the Access Points (APs) are configured to run two channels apart. AP A is configured to run on Channel 2 and AP B is configured to run on Channel 5 of the 802.11b spectrum. The experiments conducted are similar to that performed in section 4.5.1. The results observed in each of the four experiments are as follows:

1. In the first case the TTCP_RATE value is set to 40. The network overhead applied is kept constant for all the three protocols. The TOTAL_LOSSES observed for each of the three protocols is given in Table A.1.

Table A.1 TOTAL_LOSSES Observed, TTCP_RATE=40, APs A And B On Channels 2 And 5 Respectively.

<table>
<thead>
<tr>
<th>Machine Name</th>
<th>RoboCup Unicast(5,6,8)</th>
<th>RoboCup Multicast</th>
<th>RWMP(5)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gumby5</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Gumby6</td>
<td>0</td>
<td>80</td>
<td>50</td>
</tr>
<tr>
<td>Gumby8</td>
<td>0</td>
<td>31</td>
<td>70</td>
</tr>
</tbody>
</table>
Table A.2 TOTAL LOSSES Observed, TTCP RATE=60, APs A And B On Channels 2 And 5 Respectively.

<table>
<thead>
<tr>
<th>Machine Name</th>
<th>RoboCup Unicast(5,6,8)</th>
<th>RoboCup Multicast</th>
<th>RWMP(8)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gumby5</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Gumby6</td>
<td>0</td>
<td>95</td>
<td>67</td>
</tr>
<tr>
<td>Gumby8</td>
<td>2</td>
<td>41</td>
<td>0</td>
</tr>
</tbody>
</table>

Table A.3 TOTAL LOSSES Observed, TTCP RATE=80, APs A And B On Channels 2 And 5 Respectively.

<table>
<thead>
<tr>
<th>Machine Name</th>
<th>RoboCup Unicast(8,5,6)</th>
<th>RoboCup Multicast</th>
<th>RWMP(6)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gumby5</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Gumby6</td>
<td>7</td>
<td>38</td>
<td>0</td>
</tr>
<tr>
<td>Gumby8</td>
<td>0</td>
<td>43</td>
<td>59</td>
</tr>
</tbody>
</table>

2. In the second case the TTCP RATE value is set to 60. The experiments are conducted in the presence of ADDED TRAFFIC as explained in case 1. The TOTAL LOSSES observed for each of the three protocols is given in Table A.2.

3. In the third case the TTCP RATE value is set to 80. The TOTAL LOSSES observed for each of the three protocols is given in Table A.3.

4. In the fourth case the TTCP RATE value is set to 100. The TOTAL LOSSES observed for each of the three protocols is given in Table A.4.
Table A.4 TOTAL LOSSES Observed, TTCP RATE=100, APs A And B On Channels 2 And 5 Respectively.

<table>
<thead>
<tr>
<th>Machine Name</th>
<th>RoboCup Unicast(6,8,5)</th>
<th>RoboCup Multicast</th>
<th>RWMP(8)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gumby5</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Gumby6</td>
<td>0</td>
<td>48</td>
<td>55</td>
</tr>
<tr>
<td>Gumby8</td>
<td>0</td>
<td>40</td>
<td>0</td>
</tr>
</tbody>
</table>