WIRELESS OPEN-ACCESS
RESEARCH PLATFORM

Lab 4: noMac

Chris Hunter

Rice University

WARP Project

Document Revision 6

November 1, 2007

1 Introduction

The WARP networking lab is intended to introduce the tools required for MAC development on WARP. In previous labs, code on the PowerPC was used to drive cores running in the FPGA fabric. We will make extensive use of the WARPMAC framework (http://warp.rice.edu/WARP_API). The framework handles the hardware interaction, and we will program a basic MAC.

In this particular lab, we will implement the simplest MAC: no MAC at all. We will use the functions of WARPMAC to create a true bridge between Ethernet and the wireless PHY. Any packet that is received on one medium will be forwarded to the other medium as-is; there will be no addressing and no retransmissions. This behavior is shown in Figure 1.

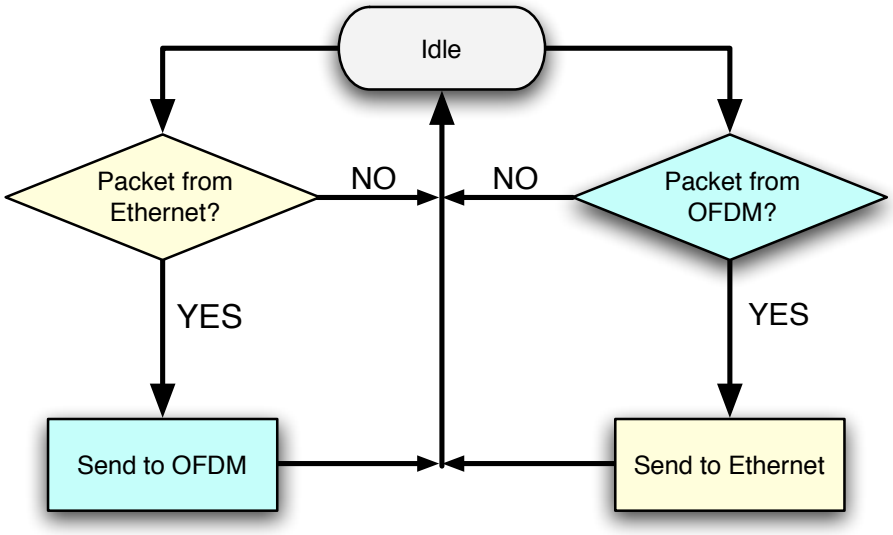

Figure 1: *Ethernet MAC - Wireless PHY Hub algorithm*

The WARPMAC API will be required throughout this lab exercise. The full API documentation is available online (http://warp.rice.edu/WARP_API). Skeleton code is provided which compiles without any modification. This code monitors the PHY for good and bad packets. It blinks LEDs on the FPGA board to indicate reception of both. Within the skeleton code are comments describing the code that needs to be written.

Note: All files are stored in `C:\workshop\userN\` where `userN` is your user login location. This location will be referred to as `.\` for the rest of the lab.

2 Instructions

1. Open the *system.xmp* in `.\Labs4_6_MAC\`
2. Because we will only be dealing with the software project in this lab, we can ignore everything in the "System Assembly View." Click on the "Applications" tab as shown in Figure 2

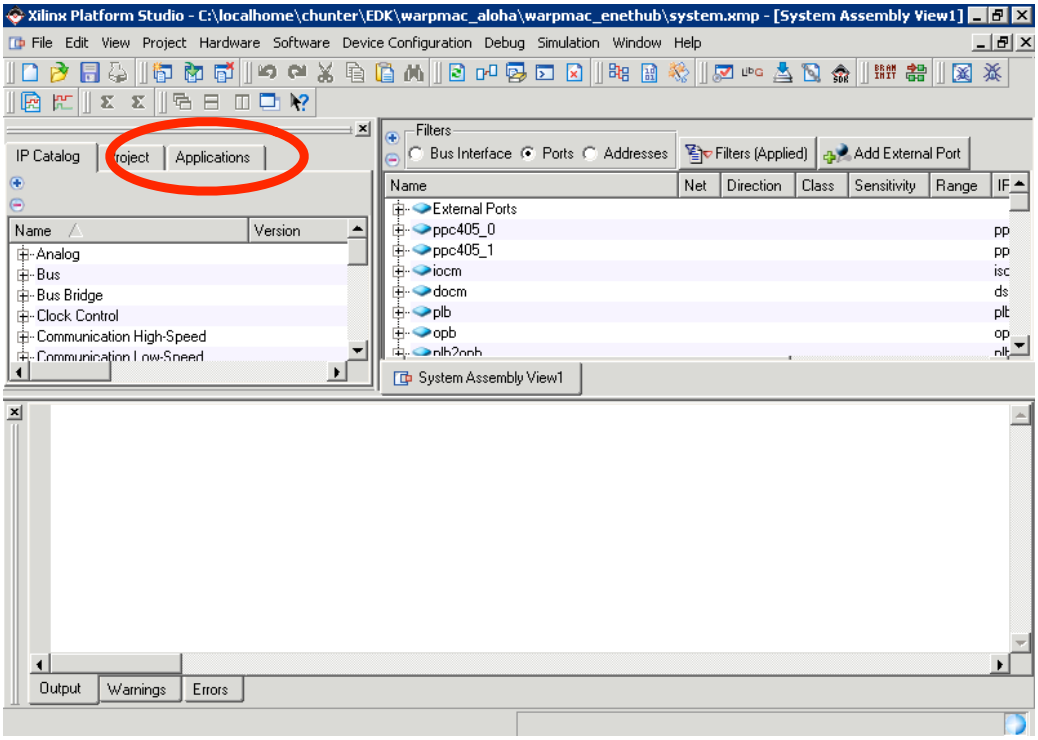

Figure 2: *Application Tab*

3. Expand the **NOMAC** application. Within the "Sources" and "Headers" hierarchies, you will be greeted by a number of files required for the project to build:
   - *noMac.c* - This file is the where all of the user's modifications will take place.
   - *warpmac.c* and *warpphy.c* - These file contains all of the MAC development and PHY interface framework. These frameworks provide the user high-level functions for abstracting interactions with the wireless and wired network interfaces.
4. Open the *noMac.c* file and modify the skeleton code to implement the functionality described in the state machine above. There are helpful comments included in the skeleton code to guide your implementation.
5. Choose Device Configuration → Update Bitstream to compile your code and create the FPGA programming file. Use iMPACT on your local PC to download the bitstream to your FPGA board to test your design.

3 Testing your MAC

If the MAC works as expected, the wireless network in the room should be equivalent to a wired Ethernet hub connecting every computer. To make sure this is the case, try this experiment:

- Use *ping* from the command-line on your local PC to see the effective round-trip times over the wireless network. A machine with IP address 192.168.1.20 is available to respond to pings.

Notice the performance of the link as a function of other groups completing the assignment. As more groups finish, the wireless medium will become more contended. The MAC we have built has no means of dealing with this contention, so performance will be quite poor. We will see the gains of a true MAC in the next lab.