WARP WIRELESS OPEN-ACCESS RESEARCH PLATFORM

WARP MAC Lab: noMac

Chris Hunter & Patrick Murphy

Rice University

WARP Project

April 27, 2011

1 Introduction

This exercise is intended to introduce the tools required for MAC development on WARP. We will make extensive use of the WARPMAC framework (http://warp.rice.edu/WARP_API). The framework handles most of the details in setting up the WARP hardware and OFDM PHY, allowing us to focus on the higher-level MAC code.

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

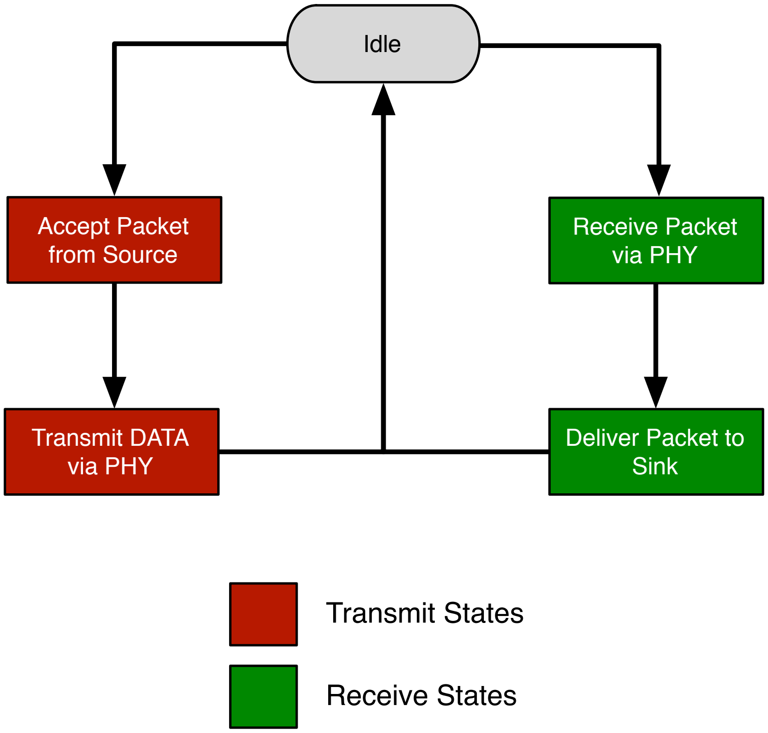

Figure 1: *Ethernet MAC - Wireless PHY Hub algorithm*

The WARP 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. Within the skeleton code are comments describing the code that needs to be written. You can find these sections by searching for "WORKSHOP PSEUDOCODE."

Note: All lab files are stored locally on workshop machines. This location will be referred to as .\ throughout the lab materials.

2 Instructions

1. Open the *system.xmp* in `.\WARP_MAC_LABS\`
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

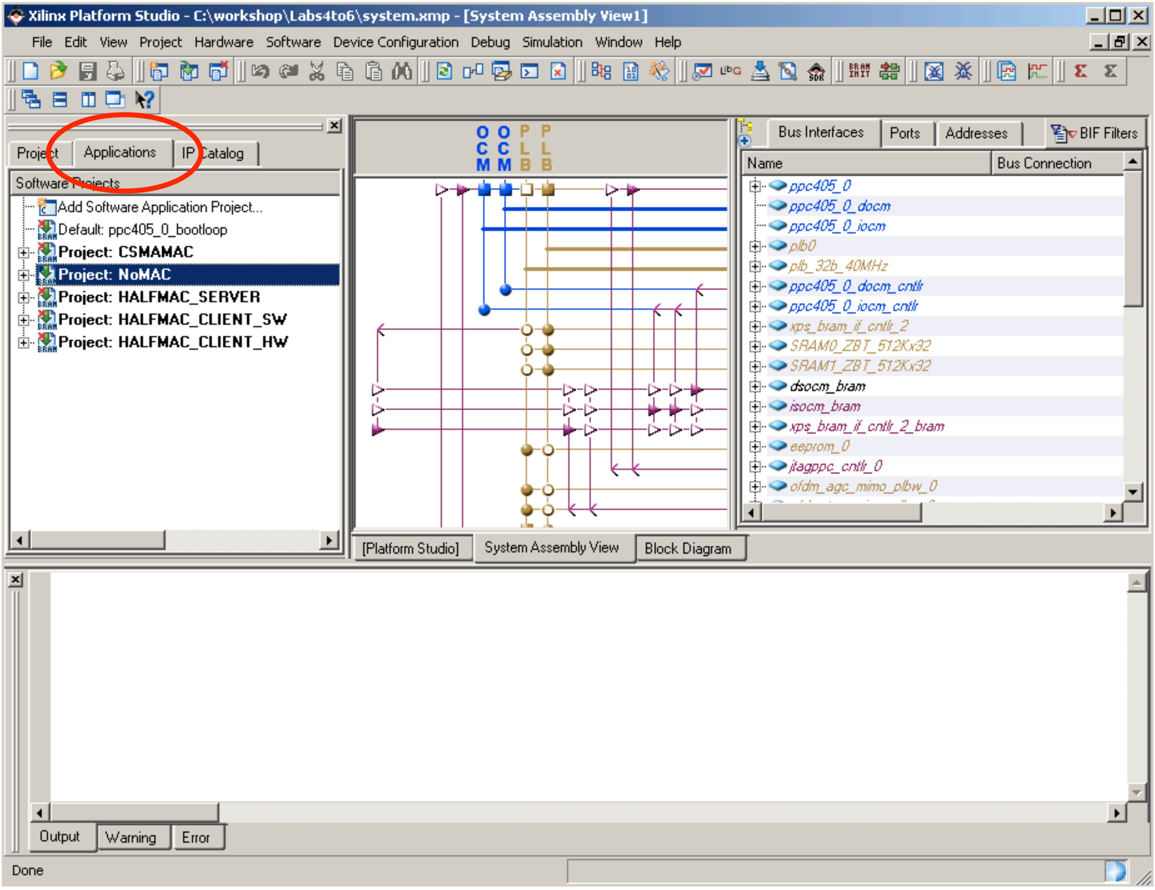

Figure 2: *Application Tab*

3. Expand the **NOMAC** application. Within the "Sources" and "Headers" hierarchies, you will see the source code for the noMac project:
   - *noMac.c* - The main file for noMac; all of your code changes will be in this file.
   - *warpmac.c* and *warpphy.c* - These files contain all of the MAC development and PHY interface frameworks. 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 comments in the code to guide your implementation.
5. Choose Device Configuration → Download Bitstream to compile your code and download it to the FPGA.

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. We will post the IP address of a machine you should be able to ping.

Take note that the wireless network performance will change as more users complete the lab. As more groups finish, there will be more contention for access to the wireless medium. This MAC implementation does not attempt to mitigate contention, so performance will degrade as more users test their code. We will see the gains of a true MAC protocol in the next lab.