WARP WIRELESS OPEN-ACCESS RESEARCH PLATFORM

WARP MAC: halfMac (with Software ACK Transmission)

Chris Hunter & Patrick Murphy

Rice University

WARP Project

April 27, 2011

1 Introduction

In the previous lab, WARP nodes had no concept of addresses. The "MAC" was little more than software that transferred packets between the wired and wireless network interfaces. In this lab, we will implement a "true" medium-access layer to control nodes in the topology shown in Figure 1.

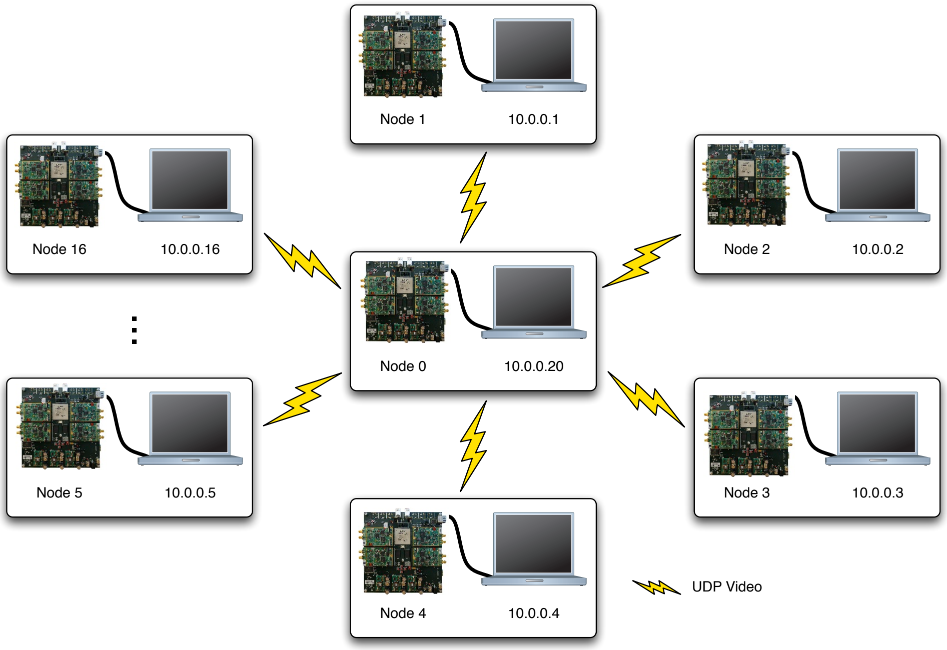


Figure 1: *Lab Topology*

In this experiment, a central transmitter will stream UDP video to each PC. Each wireless node has a unique MAC address, allowing the node to filter and process only its own packets. In this exercise, the wireless MAC address is set using the DIP switch on the WARP FPGA board; each node is already configured with the appropriate DIP switch value.

The central node implements an ARP table, mapping WARP MAC addresses to the IP address of the attached PC. The central node also implements a modified version of a CSMA MAC. This MAC implementation expects to receive an ACK for each DATA packet it sends to a client. It will re-transmit unacknowledged packets 8 times before dropping the packet entirely.

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

In this lab, you need to implement code which realizes the following behavior:

- Check the destination address of each received wireless packet. If it is addressed to your node, send an ACK to node 0.
- The sequence number of received DATA packets should be checked against previously received sequences. If the packet is unique, send it via Ethernet to your PC.
- If a packet is received via Ethernet, the MAC should send it wirelessly to node 0. Your node should not expect an ACK.

This behavior is illustrated in Figure 2. In effect, your code will implement the receive half of of a full CSMA protocol (hence the name “halfMac”).

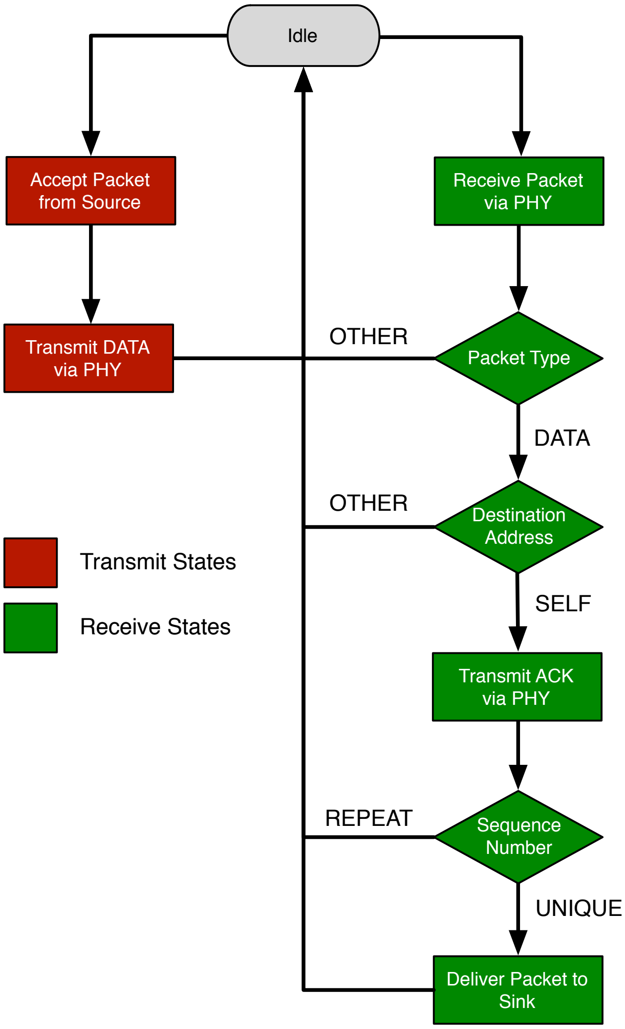


Figure 2: *halfMac State Diagram*

It is important to note that, while the central node implements retransmissions for unacknowledged packets, your MAC implementation will not. Because of the unidirectional nature of UDP, the only Ethernet traffic your node will forward to node 0 is an ARP reply to establish the Ethernet MAC address lookup tables on both PCs.

The WARP API will be required throughout this lab exercise. Skeleton code is provided that should compile without user modification.

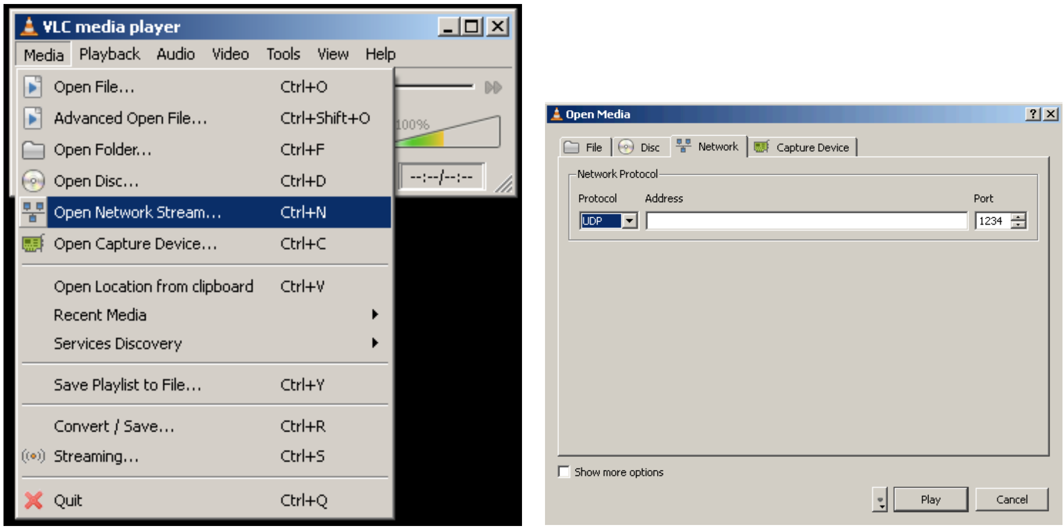
2 Instructions

1. Open the XPS project at `.\WARP_MAC_LABS\system.xmp`.
2. You should modify "HALFMAC_CLIENT_SW" software application.[1] You can select which project is used to program the FPGA by right-clicking and checking "Mark to Initialize BRAMs."
3. Within the "Sources" and "Headers" hierarchies, you will see the source code for this project:
    - *halfmac_client_sw.c* - The main file for this project; 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 *halfmac_client_sw.c* file and modify the skeleton code to implement the functionality specified in the comments.

3 Testing your MAC

3.1 Video Reception

Launch VLC Media Player, open the network stream as shown in Figure 3, select "UDP" and click "play." If your MAC is successfully receiving wireless packets and sending them via Ethernet, VLC should open a player window showing the video streaming from the central node.



Protocol: UDP
Port: 1234

(a) Open the network stream

(b) UDP on port 1234

Figure 3: VLC configuration

[1] The "HALFMAC_SERVER" project is provided for reference; this is the code running on the central node.