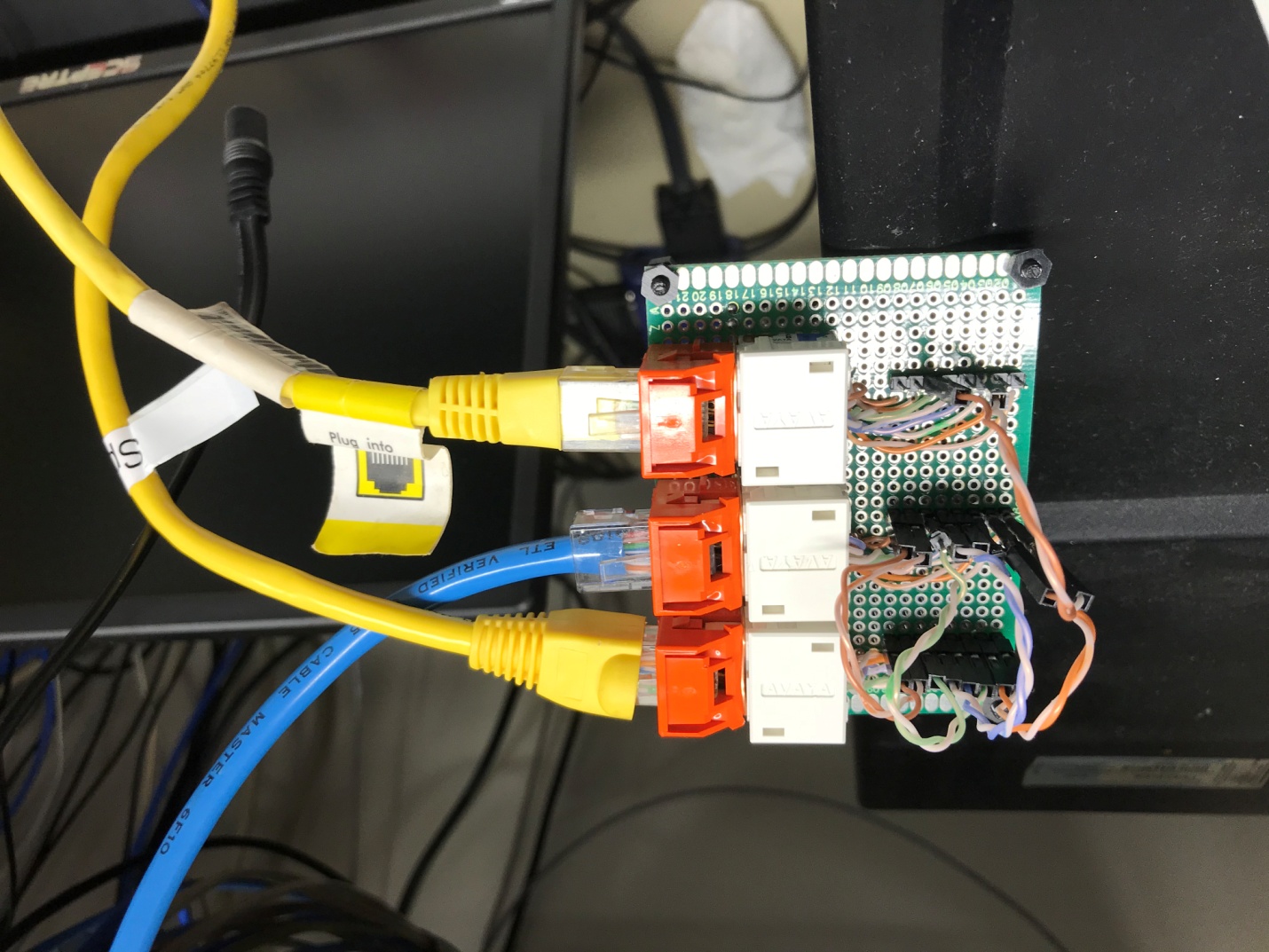
Using wireshark to capture packets between a Niles GXR2 and an ICS Controller

These notes are undergoing construction. Updates will be provided daily.

One caveat before we get started. The devices and procedures described in this paper are experimental and should not be a permanent part of your Niles configuration. No attempt to correct for impedance mismatch is made, nor is there any effort to isolate the laptop from the other network components. In short, what I did was crude, but effective.

The problem with Niles equipment is that it is proprietary (well, somewhat proprietary.) It contains a 12 port built-in switch with a standard Ethernet configuration for the send and receive pairs (pins 1 and 2 for send, controller-side, and pins 3 and 6 receive.) Unfortunately, the other four pins (4, 5, 7, and 8) are not standard. Pins 4 and 5 are used for Infrared signals and pins 7 and 8 have 12VDC on them. Therefore, it could be a problem if a standard network connection was made between one of the GRR2 controller ports and, say, a laptop. Because I wanted to look at the network traffic between the GXR2 and a controller, I decided to try bridging the connection to see what would happen. The rules for the bridge were simple: it had to provide a straight-through connection from the GXR2 to the controller for all eight wires. It had to provide the capability to separately patch either the send or receive pair to a laptop or other device running wireshark. The photo below shows my prototype bridge.



The bottom two connectors are jumpered together as straight through (pin 1 to pin 1, pin 2 to pin 2, etc) The network connection to the controller is plugged into the bottom connector and the connection to the Niles GXR2 is plugged into the center connector. The connection to the laptop is plugged into the top connector. The send pair from the botton connector is jumpered to pins 3 and 6 of the top connector. That way, the laptop will be listening to traffic that is being sent from the controller to the GXR2. If you want to listen to traffic being sent from the GXR2 to the controller, then you would move the jumper to pins 3 and 6 on the bottom connector.

Since the laptop is only connected to the network bridge by its receive pair. It has no way of asking for a DHCP lease. That is why it is necessary manually configure the wired connection on the laptop for a static IP. You want to give it one in the range of 10.100.x.x/16 because that is within the range of the Niles GXR2’s internal network. I used 10.100.123.123 with a subnet mask of 255.255.0.0.

I used the following filter for wireshark:

ip.dst==10.100.0.1

That wil filter out all traffic except for packets that have a destination address of the GXR2. Then, I press buttons on the controller and watch what happens.

When looking at packets from the GXR2 to the controller, I used the following rule for wireshark:

ip.src==10.100.0.1

When you do this, you’ll notice that the destination IP’s are all multicast addresses in the range of 132.0.0.1–6 and 132.0.0.