A few words about the “environment variables” that tell which port is which.

The internal wifi asset is a network interface (nic) when rc.local is launched at startup, and it runs a script called scanports.sh. This script scans the networking hardware and determines if we have just the two internal devices or if there is also a wifi dongle plugged in. It sets three internal variables called ENET, WIFI, and WAPT to the system names of the associated ports (ie: eth0, wlan0, wlan1). Since there is no pleasant way to pass internal variables UP to their parent process (or even to the shell level itself), we then write the three variables to a file called /usr/Fusion/etc/portassigns. The file will basically contain one of two possible outcomes:

|  |  |
| --- | --- |
| ***No Wifi Dongle*** | ***WiFi Dongle*** |
| **export ENET=eth0** | **export ENET=eth0** |
| **export WAPT=wlan0** | **export WAPT=wlan0** |
| **export WIFI=NULL** | **export WIFI=wlan1** |

This allows us to use the bash statement **source /usr/Fusion/etc/portassigns** in other scripts to fetch a copy of these strings all ready to be used by any child processes they might launch.

Upon return to rc.local, we scan the airwaves using the ${WAPT} device looking for the Community WiFi beacon (ie: a wireless signal with the SSID=”MyBot\_Community\_WiFi”.) If we find that beacon and ${WIFI} == “NULL”, then there is no WiFi Dongle and we’re going to use the Internal wifi asset (aka: wlan0) as a network interface and not as an access point.

We modify the portassigns file so that WAPT=NULL and WIFI=wlan0, set for NIC mode and proceed. In a sense, you can expand the portassigns outcomes table with a third column (and add a row for conditional clarity:

|  |  |  |
| --- | --- | --- |
| ***No Wifi Dongle*** | ***Beacon but No Dongle*** | ***WiFi Dongle*** |
| Standalone Mode –same as today | Community Mode – new | Two-Port Mode – same as today |
| **export ENET=eth0** | **export ENET=eth0** | **export ENET=eth0** |
| **export WAPT=wlan0** | **export WAPT=NULL** | **export WAPT=wlan0** |
| **export WIFI=NULL** | **export WIFI=wlan0** | **export WIFI=wlan1** |

When it launches the FusionServer, rc.local passes the three values stored in the local variables ENET, WAPT, and WIFI to the FusionServer using the names WIRE\_NET, WIFI\_WAP, and WIFI\_NIC respectively:

# Switch to the application server directory

cd /${MAIN\_DIR}/FusionServer/Build

# Launch node.js and the Fusion Server

sudo NODE\_ENV=production WIRE\_NET=${ENET} WIFI\_WAP=${WAPT} WIFI\_NIC=${WIFI} forever start server.js

This supports the current “only one Mimio MicroCloud in the vicinity” demo systems, but will be problematic come multiple MMC environments.