Wireless Human-Machine Interfaces

Background

Wireless networks in yesterday's automation world were limited to a serial protocol. This limited your ability to add Human Machine Interfaces (HMI). For the HMI to gather data from all over the network it would need to be connected to the systems master PLC and the systems master PLC would read/write data via message instructions to the slave PLC's.

Today's wireless networks feature Ethernet protocols, supporting EtherNet/IP, making it is easier to deploy HMI's. However, it's much easier to get yourself into trouble when designing a wireless network with HMI's.

Problem

When integrating an HMI into an application using EtherNet/IP, whether it's a PC running RSLinx or a PanelView HMI panel, an Ethernet driver needs to be setup and, unlike a serial driver, the Ethernet driver can be setup to communicate with any device on the network as long as it wirelessly has access to it. This clearly has its advantages but it also has its problems. When using FreeWave's FGR and/or HTPlus series of Ethernet radios, the network works on a "Master/Slave" basis meaning that basically any data that is received via the Ethernet port of an Endpoint (slave) radio is sent to the Gateway (master) radio and with any given network there can only be one Gateway radio. Because of basic radio physics, only one Endpoint radio can communicate to the Gateway radio at any given time. When you're dealing with a protocol that does polling this generally doesn't represent any problems, however when dealing with a protocol that can operate on a broadcast basis, this can create problems.

Problem Details: Simultaneous Responses

For the purpose of this paper I'm going to use a PanelView Plus to illustrate the point but the concept is the same for any Allen Bradley HMI and/or any HMI that uses EtherNet/IP. One of the advantages of setting up a PanelView Plus is that linking a tag to a data point in any given device in the network is a very simple process. As long as you have a path to the device you have access to its data. This level of simplicity can also get you into trouble if you don't pay close attention to the details. When setting up a PanelView Plus screen, objects on the screen are linked to data tags and each screen has a "Maximum Tag Update Rate". This update rate dictates how frequently the PanelView Plus requests the data from each tag with the default being once per second. However, you can't select different rates for different tags on any give screen you can only set the refresh rate for the entire screen. The problem comes in when you have multiple tags from multiple PLC's. When the PanelView Plus updates its screen, it send out a "kick timer" for each tag and when these tags are in different PLC's this request goes to each PLC at the same time (about one kick timer per millisecond). With this, each PLC will respond at virtually the same time. The problem is that only one Endpoint radio can connect to the Gateway at any given time and depending on the radio and the RF environment the amount of time it takes for the data to get through will vary. If the amount of time it takes to get the requested data back is longer than the screen refresh rate the application will eventually experience problems.

To illustrate the point, let's look at an example in detail. Let's create an application using FGR2-PE Ethernet radios and a PanelView Plus gathering data from five remote PLC's with each PLC having four data points. With the PanelView Plus set at its default, once a second, it will send out 20 kick timers within about 25ms and all five PLC's will start responding in about the same amount of time. Because the radios operate on a "Master/Slave" basis only the first radio to make a connection will get to send its data while the other four will start to buffer their incoming data. It's worth noting that while the tags from any given PLC are requested separately, once a given PLC gets connected to the PanelView Plus it will send all of its tag data during that connection. With four tags coming back from each PLC, the first PLC takes about 280ms to get its data through. When the first PLC is done the second one gets through and the remaining three PLC's continue to buffer their data as well as the first PLC is now buffering any data that it has coming into its Ethernet port. This continues until all five PLC's have finished sending their data. With each PLC taking approximately 280ms to send its data, the total elapsed time is 1400ms or 1.4 seconds. The problem is that the PanelView Plus, once a second, sends out another 20 kick timers and all of the PLC's either started to reply (if they had already finished with the first request) or buffered the request to respond once it was done with the first request. This will continue and as time progresses so will the amount of data being buffered until the data coming back is so old that it is no longer considered valid by the PanelView Plus and/or the PanelView Plus times out the data. It's worth noting that this is simply an example application to illustrate the point. Any given application will obviously have different amount of data points coming back from a different amount of PLC's. Unfortunately this means that it can be hard to predict how long it will take for each PLC to respond with its data. In addition, while the FGR2-PE's are very capable of dealing harsh RF environments, different RF

environment can produce different data throughput speeds which in turn will change how long it takes for any given PLC to get its data through the link (changing the above calculations).

Solution 1: Maximum Tag Update Rate

So, what's the fix? There are a couple of ways of approaching this all of which involve pre-planning at the system design stage. One approach is to calculate the amount of data coming back to the HMI and set the "Maximum Tag Update Rate" accordingly. In the above example this setting would need to be set to two seconds and as long as there wasn't any time critical data. With that said, if the application used HTPlus radios instead of the FGR2-PE radios the overall response would be 500ms as the HTPlus runs at a much fast RF speed. In fact if the application supports the HTPlus' specifications (mainly range) it should be the radio of choice because of its higher speed capability but it's not the fix in all applications. While it is faster than the FGR2-PE it still operates on a "Master/Slave" basis (only one Endpoint can connect to a Gateway at any given time) so there is still a limit to the number of remote tags that can answer back in the allotted time. With this in mind, another approach is to have multiple Gateway radios each with their own set of Endpoint radios. The concept is that with multiple Gateways their Endpoints will be able to send data back at the same time (one per Gateway). With this model all you have to do is some basic math to figure out how many Gateways you need and connect them via a level 2 or level 3 managed switch to the HMI as well as make sure that you practice good RF management with regards to the radios configuration and antenna installation (if this were done in the above application example, two Gateways would be needed to keep the response down below the one second refresh rate).

Solution 2: Master PLC

With all of that said, the best solution, regardless of what radios you're using, is a Master PLC. The basic issue that leads to potential problems is the lack of control of the EtherNet/IP protocol. It operates in a broadcast format rather than a polling one and as a result all requested data comes back at once. When you're connected to a hardwired network, the network infrastructure can handle the traffic but when you need to move that traffic wirelessly (as you can see from the above examples) you can quickly get into trouble. With a Master PLC, you have much better control of the traffic because you can "poll" the needed data from each of the remote PLC's. Another advantage is that you can program the PLC's (Master and Remote) to react to different situations not the least of which is a loss of communications. With just the HMI, when a link is lost, the HMI will notify you of the problem but with a Master PLC both the Master and the Remote PLC can be configured to react depending on the needs of the application. It's worth noting that adding an HMI to an application using a Master PLC doesn't mean that the HMI has to be hardwired connected to the Master PLC. The HMI (or multiple HMI's) can be connected via Endpoint radios or because FreeWave's radios support multiple networks in a single location you can connect the HMI via a separate radio network.

Conclusion

In short, years ago when HMI's were added to a wireless application it was connected to the application's network Master and the master gathered all of the needed information. Today, that's still a great way to add HMI's to your automation application especially when dealing with EtherNet/IP. When it was your only option it was easy to lose sight of the advantages it brought but now these advantages sometimes become requirements in order to make the best use of your equipment.