Remote Cellular TCP/IP Access to GE Fanuc Ethernet and Serial Devices

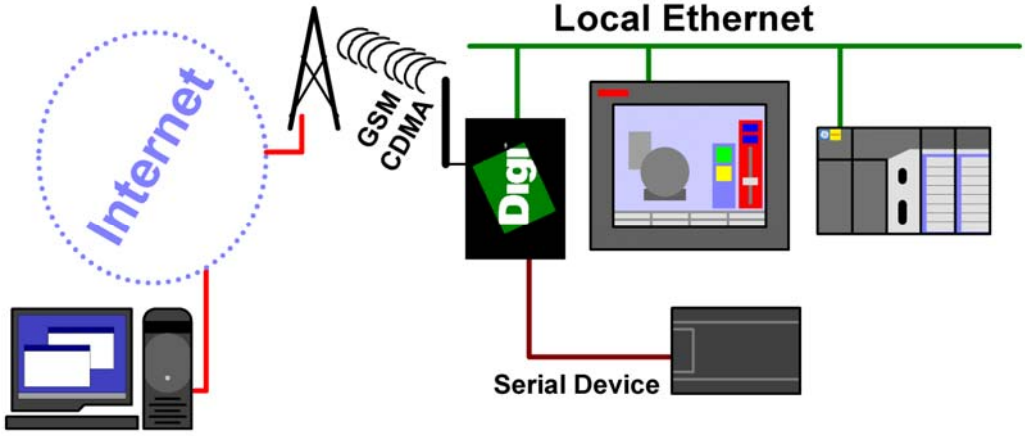
Keywords: Cellular, Quick Panel, PACSystems, VersaMax,

Abstract: This document describes how to set up the Digi Connect® WAN products (Digi Connect WAN, Digi Connect WAN IA, and Digi Connect WAN VPN) for remote cellular TCP/IP access to GE Fanuc equipment, such as Quick Panels, PACSystems, and VersaMax-E05 with Cimplicity Machine Edition. The Digi Connect WAN Family functions much like a home DSL/Cable modem, except the connection is by digital cellular signals such as GSM or CDMA. This enables wireless "Ethernet" solutions on a metro, regional, or global scale.

1 Introduction

1.1 Example Application

To illustrate the use of Digi Connect WAN products with your GE Fanuc equipment, consider the following example:



Key Features:

The Digi Connect WAN product used with your GE Fanuc equipment provides several key features:

- Provides outgoing Network-Address-Translation (NAT) and incoming TCP/UDP port forwarding. Some models act as VPN end-point.
- Maintains an always-up IP connection, either on the public Internet or by customized private networks established through your cellular carrier.
- Being IP-based, all common Ethernet protocols can be used concurrently, including HTTP (Web browsing), SRTP, and Modbus/TCP.

- Existing applications, such as Cimplicity and OPC, can be configured to access the field equipment through existing corporate LAN connections.
- Intelligent field devices can use IP-based protocols to send email, file updates, or report-by-exception notifications.

2 The Digi Cellular Family

The Digi Connect WAN family includes three models to better target your needs. This document (90000XXX) covers use of the Digi Connect WAN VPN and Digi Connect WAN products with general equipment and to pass Modbus protocols through without bridging. Document 90000773 covers in more detail the use of the Digi Connect WAN IA as a Modbus IP to Serial Bridge. *When we mention **Digi Connect WAN** in this document we are describing features than can enabled in all three models.*

Here is a brief comparison of the product features:

Feature *(See the referenced Digi Document for further details)*	**Digi Connect WAN IA** *(9000773)*	**Digi Connect WAN VPN** *(This One)*	**Digi Connect WAN** *(This One)*
1) Modbus/TCP to serial bridge	Yes	No	No
2) Remote TCP/IP connection to local Ethernet-enabled equipment	Yes	Yes	Yes
3) Local Ethernet-enabled equipment can use TCP/IP protocols out to remote servers	Yes	Yes	Yes
4) VPN end-point securely "grafts" local Ethernet onto remote network	Yes	Yes	No
5) Remote access to local serial port by raw TCP, UDP, SSH, or SSL/TLS	If Modbus Bridge is off	Yes	Yes
6) Enables remote console management of routers and servers	If Modbus Bridge is off	Yes	Yes
7) Interacts with standard routers for redundant (backup) paths	Yes	Yes	Yes
8) Digi RealPort® supports legacy serial-only applications	If Modbus Bridge is off	Yes	Yes
9) Digi Configuration by remote, Ethernet, or serial connection	Yes	Yes	Yes
10) Digi acts as a local DHCP server	Yes	Yes	Yes
11) Supports RS-232/422/485	Yes	Yes	RS-232(*)

(*) Note: Digi Connect WAN is migrating to RS-232/422/485 mid-2006. If you need a Digi Connect WAN with RS-422/485, check with Digi to confirm availability.

Following is a detailed discussion of these features:

2.1 Modbus/TCP to Serial Bridge

The Digi Connect WAN IA allows TCP/IP-based masters to query a local serial or Ethernet-based slave. Local Ethernet-based masters can query a local serial slave. Alternatively, a Modbus serial master can access both local Ethernet slaves and remote slaves by the cellular link.

The serial protocols supported are Modbus/RTU and Modbus/ASCII.

The TCP/IP based protocols supported are Modbus/TCP (as TCP or UDP), Modbus/RTU (within TCP or UDP), and Modbus/ASCII (within TCP or UDP).

Note: see document 90000773 for fuller explanation of Modbus Bridge use.

2.2 Remote TCP/IP connection to local Ethernet-enabled equipment

The Digi Connect WAN family allows remote TCP/IP clients to access local Ethernet devices by TCP or UDP port forwarding. Since the Digi Connect WAN is represented externally as a single IP address, this port forwarding limits most protocols to a single local Ethernet device. However, protocols that support configurable port numbers – such as web browsers – allow forwarding to multiple local Ethernet devices. Web browsers routinely are assigned other port numbers, such as 8000 or 8080, which are accessed as http://192.168.1.20:8000 or http://192.168.1.20:8080. A VPN connection overcomes these limitations (see 2.4, “VPN end-point securely ‘grafts’ local Ethernet onto remote network” below).

The Modbus Bridge functionality overcomes this limitation by allowing the Modbus/TCP Unit Id (or Modbus serial slave address) to forward incoming Modbus requests to up to 32 local devices. This forwarding also helps reduce your cellular bills, since the TCP/IP stack of the Digi Connect WAN IA handles the high latency of cellular links much better than most “Ethernet-enabled” products.

2.3 Local Ethernet-enabled equipment can use TCP/IP protocols targeted at remote servers

The Digi Connect WAN family supports Network-Address-Translation (NAT) and thus allows any number of local Ethernet devices to act as outgoing TCP/IP clients to access remote servers. For example, any number of local PLC could use master blocks to send unsolicited or report-by-exception data back to the central site. Since TCP/IP is being used, HMI can send SMTP email, FTP, and even HTTP to push data to other sites.

2.4 VPN end-point securely ‘grafts’ local Ethernet onto remote network

The Digi Connect WAN IA and Digi Connect WAN VPN can establish a secure IPsec (VPN or Virtual Private Network) connection back to a VPN server at your corporate site. Once this is established, the entire local subnet appears to be attached and reachable from your corporate network. This overcomes access limitations mentioned in section 2.2 above.

For example, the Digi Connect WAN IA uses the cellular-assigned IP address to connect and securely authenticate with a central VPN server. The

Digi Connect WAN IA can even have a dynamic IP address. Once connected, the cellular link and the Digi Connect WAN IA disappear from the connection, and the entire local subnet is securely accessible from the central site.

Note: The need to keep the VPN connection active means that you will need a fairly large cellular data plan.

2.5 Remote access to local serial port by raw TCP, UDP, SSH or SSL/TLS

The Digi Connect WAN family allow remote clients to open raw TCP/IP, UDP/IP, SSH or SSL/TLS sockets to access the serial port. By encapsulating a serial protocol into this socket, remote clients can access the attached serial device.

For example, an OPC server can encapsulate DF1 or Omron HostLink into a TCP socket and communicate to an existing serial PLC at site. The OPC server and PLC need to support longer timeouts to accommodate the added latencies in a wide-area network connection.

2.6 Enables remote console management of routers and servers

The Digi Connect WAN family allows remote login on serial console port for routers and servers, offering diverse out-of-band management for land lines.

For example, a Cisco router manages IP traffic over several land lines for an Ethernet subnet at a remote pumping station. If some of the land lines go down, network maintenance people may not be able to access the router by network to troubleshoot. However, the cellular link through the Digi Connect WAN allows them to log into the router and troubleshoot the situation.

2.7 Interacts with standard routers for redundant (backup) paths

The Digi Connect WAN family supports router protocols and can coordinate with traditional land-line routers, including those by Cisco. This allows normal IP traffic to use dedicated land-lines such as frame relay or ADSL links, but to automatically fail over to cellular service when required.

2.8 Digi RealPort® supports legacy serial-only applications

The Digi Connect WAN family supports the Digi RealPort® protocol. A serial-port driver is loaded under Windows, Linux, and most other common operating systems. This driver makes the remote port to appear as a physical serial port on the computer. This allows legacy applications that expect physical serial ports to work with your remote devices. More information on Digi RealPort can be found at http://www.digi.com/pdf/fs_realport.pdf

2.9 Configuration by remote, Ethernet or serial connection

The Digi Connect WAN family can be configured either remotely, by direct Ethernet, or by serial connection.

2.10 Acts as local DHCP server

The Digi Connect WAN family can act as a DHCP server for local Ethernet devices.

2.11 Supports RS-232/422/485

The Digi Connect WAN IA and Digi Conenct WAN VPN have a DIP-switch configured serial port supporting RS-232, RS-422 (four-wire) and RS-485 (two- or four-wire).

(*) Note: Digi Connect WAN is migrating to this RS-232/422/485 design in mid-2006. If you need a Digi Connect WAN with RS-422/485, check with Digi to confirm availability.

3 Performance Expectations

3.1 LAN and WAN Differences

In theory, any TCP/IP-based or UDP/IP-based protocol will work fine over any IP-based Wide Area Network. However, implementers unconsciously build in LAN timing assumptions that prevent their products from running successfully over WAN. In general, satellite and cellular networks require software to be patient. Prematurely timing out and retrying when the network is busy makes matters worse, can actively prevent lost communications from recovering, and can increase your communication costs a hundred-fold.

Here is a brief comparison of differences between “Ethernet” and “WAN”:

	Ethernet (LAN)	Satellite / Cellular (WAN)
1) Connection Delay: how long to “open a socket” or “close a socket”	Normal: less than 0.2 sec. Maximum: assume 5 or 10 seconds is failure.	Normal: 2 to 5 seconds. Maximum: must wait 30 to 60 seconds before assuming failure.
2) Reconnection effort: how hard to “try to reconnect”	Applications try to reconnect either fairly aggressively within seconds, or they just fail and expect user intervention.	Because retries cost money, applications must not retry any harder than the normally budgeted communications.
3) Response Delay: how long to “wait for a response”	Normal: less than 0.2 second. Maximum: assume 1 or 2 seconds is failure.	Normal: 1 to 3 seconds. Maximum: must wait at least 30 seconds before assuming failure.
4) Idle TCP sockets	TCP sockets can sit idle indefinitely; limited only by application protocol expectations.	Varies, but many WAN systems ungracefully interfere with idle TCP sockets; they may stop working without either end seeing a close, abort, or reset.
5) UDP reliability	On modern 100M switched Ethernet, UDP/IP is actually quite reliable with packet loss rare.	Loss of UDP packets is to be expected and can be a sizable percentage of total traffic.
6) Costs to Communicate	Only cost of generating network messages is the impact on other devices and communications.	Most WAN systems include costs based on maximum expected data bytes per month; every message sent potentially costs money.
7) Connection Failure	Loss of a TCP connection is rare and would only occur	TCP connections may fail several time a day – or even dozens of

	during system changes.	times an hour.

3.2 Will my application work?

Unfortunately, most product developers only test on Ethernet/LAN; it is very possible that the first users attempting to use WAN will have to locate and point out the problems for the developers.

3.2.1 Connection Delay

Most applications use the OS defaults – on Windows, this connection delay generally is 5 seconds. Since the application may not even manage an internal setting for connection delay, users won't have any option to change this default behavior. So even if the application allows users to define a 30-second response timeout, the initial socket connection may still time out too fast and prevent successful operation. What does this mean?

- In a best-case scenario the application does not wait long enough to open a socket, making reconnection difficult at times. As long as the application waits at least 30 second before it retries, the connection eventually recovers.
- However, the worst-case scenario occurs if the application not only times out too fast, but retries too fast. In that case, the TCP peers in effect alternate between acting as if they are connected but having to "reset" the connection due to timeouts, and assuming they need to retry the connection. This behavior could continue for as long as the network is congested, and can result in huge overage charges of hundreds or even thousands of dollars in a single month.

3.2.2 Reconnection Effort

Many applications offer users little or no control over the effort spent to reconnect after a TCP/IP connection failure. For example, a customer may configure the application for one remote poll each 5 minutes and assume they will incur less than 5MB of data charges per month. However, this application may very aggressively attempt to reconnect during network failure and literally generate more than 1GB of excess traffic in a few days. This can result in thousands of dollars of overage charges per device when the monthly bill comes.

What does this mean? Although the user "asked" the application to only talk once every 5 minutes, the application isn't really agreeing to this. Users need to confirm their applications can be configured to sharply limit attempts to reconnect. If the user budgets to only talk once every 5 minutes, ideally the application will also attempt to reconnect only once every 5 minutes and generate roughly the same amount of traffic as the normal 5 minute polls.

3.2.3 Response Delay

Many applications default to assume Ethernet/LAN responses occur in 250 milliseconds or less. Fortunately, most applications allow users to change this value. Unfortunately, some applications limit the maximum response delay to 5 or 10 seconds. A WAN-aware application should allow this setting to be at least 30 seconds and preferably at least 60 seconds.

What does this mean? Besides the obvious performance problems when too many timeouts repeatedly puts the remote device "off-line", a more risky problem is how the application handles unexpected responses (technically, "no-longer expected" responses). A simple example is an application that sends a request, then timeouts twice and tries twice. How will the application react when it receives three responses at the same time? Remember, the first two requests were not *lost*; they still reached the remote device. Their responses were just delayed longer than expected.

3.2.4 Idle TCP Sockets

Idle TCP Sockets are related to item #6 (Cost to Communicate). The obvious solution to reducing cost is to slow down data polls. However, at some point, the idle TCP sockets become "unreliable". The sockets are not unreliable in a UDP/IP sense, but in that the application thinks it has a valid TCP socket, but it does not. The application will send a packet, wait, and see no ACK or other indication the socket is closed. So it will follow the normal TCP rules of back-off and retry. But this activity is in vain, as the only solution will be to abort (not close) and then reopen the socket.

This issue varies based on WAN technology, but a good rule of thumb at present is that you must either send data or a TCP keep-alive every 4-5 minutes to keep the TCP socket healthy.

3.2.5 UDP Reliability

UDP reliability may seem like a moot point; by definition UDP/IP is unreliable. An application using one or two UDP packets per transaction will likely handle WAN fine. The big problem arises with applications that require tens of thousands of sequential UDP packets to complete a single transaction, such as TFTP for file transfer. The longer response lags and higher probability of UDP packet loss may prevent the application from ever completing the transaction.

However actual cellular tests with UDP/IP show it to be very reliable compared to traditional analog modems. Users can expect 1 or 2 lost UDP packets per 10,000 packets sent. This compares very favorably to tradition analog modems where errors where expected every few hundred packets.

3.2.6 Cost to Communicate

Few applications are written to optimize network traffic; after all, the end devices themselves are usually the limiting factor. But put such applications across a WAN, and you may discover that 99% of the data you are paying for is either protocol overhead or data updates without any change in value.

Here are ball-park monthly data costs (as of mid-2006) based on ***polling 10 words or registers of data*** – about 35 bytes of user protocol data.

Poll rate	TCP/IP	UDP/IP	Monthly
Once per second	500 MB	230 MB	$250 or more
Once per 5 seconds	100 MB	46 MB	$75 - $100
Once per 1 minute	9 MB	4 MB	$20 - $30
Once per 5 minutes	3 MB	1 MB	$15 - $20
Once per 15 minutes	1.5 MB	0.3 MB	$10 - $15
Once per 1 hour	0.9 MB	0.1 MB	$5 - $10

Ultimately, to minimize cost applications may need to be rewritten to implement Report-By-Exception or Change-of-State – preferably by UDP/IP. This allows a system to **operate** with a once-per-hour budget yet have the **responsiveness** of a once-per-5-seconds poll rate.

3.2.7 TCP Connection Failure

The interruption of the TCP connection has creates 3 specific risks:

First, many applications have a connection start-up phase which could involve higher than normal traffic. For example, the client may attempt to upload a majority of the device status and configuration. This means repeatedly losing and re-establishing the connection could greatly increase monthly data usage.

Second, some application programmers haven't expected failure so soon after restart. We have seen some applications "hang" if the initial connection phase is interrupted during certain critical steps. In other words, during connection startup there may be a small window-of-opportunity where poorly written or poorly tested code causes unrecoverable failure.

Third, given the high latency of the system and the fact that during network recovery a higher-than-normal percentage of packets are lost, we have seen resource deadlocks develop. For example, a client connects to a remote device, the remote device accepts the connection, but the acceptance response is lost. TCP/IP promises eventually this situation will recover, but the client and remote device will handle this recovery at different rates. So there is risk that the client will abort the first connection and attempt to reconnect a second time; while the remote may reject the second connection since it still considers the first connection viable and pending.

3.3 IP Address Considerations

In general there are three types of "service plans" for IP address assignment that you can contract with your carrier.

3.3.1 Proxy or Private (Hidden) IP address

The lowest-cost service plan will be a Proxy plan, where the Digi device is assigned a private, non-routable IP address, such as 10.x.x.x. Your service provider appears to be a huge "home network" that allows outgoing connections but prevents all incoming connections. This service plan only works if your field device initiates all communications to your central server. Since the IP address is unreachable from your central server, even attempting to 'send' the IP address to your server will not enable it to initiate a connection.

3.3.2 Internet or Public (Exposed) IP address

In an Internet or public (exposed) IP address plan, the Digi device is assigned a dynamic public IP address, such as 166.x.x.x, plus the service provider usually maintains a DDNS server allowing you to locate the Digi device by a DNS lookup. Your field device can initiate communications to your central server. Your central server can use DNS lookup to initiate communications to your field device. Since the IP address is fully exposed as public, others are free to probe and attempt to connect to your field device.

3.3.3 Custom plan with fixed IP address and other options

In a custom plan, you can arrange IP addresses with your service provider as required. Most large users will arrange a 100% private and hidden network based on fixed IP addresses. However, custom plans generally cost extra, or are reserved for larger customers with hundreds of cellular devices.

3.4 What about the advertised "Unlimited Data Plans?"

Unfortunately, the "unlimited data plans" are not for you. Cellular carriers split data plans into two types of service:

- The largest group of data users consists of a mobile phone, PDA, or notebook computer in the hands of a human user. The mobile device is connecting out to the Internet; in fact it is likely impossible for a remote server to ever connect to the mobile device. Carriers know that the human user driving these devices normally use no data at all, and only use large amounts of data for short bursts of time, so the notion of "unlimited data" is tolerated.
- In contrast, the other group of data users can be referred to as "machine-to-machine" or M2M. Such a telemetry system can easily consume its full bandwidth 100% of the time forever. In these situations, a central server or "the Internet" is connecting out to the remote mobile device. Cellular carriers require M2M users to sign up for "Telemetry Data Plans" – none of which offer unlimited data once you read the fine print.

3.5 Costs of continuous versus occasional access

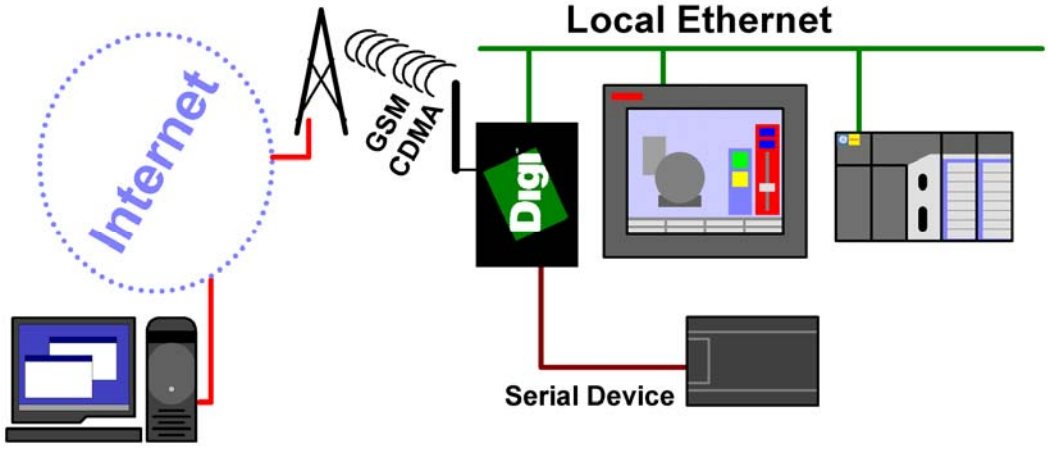
For continuous access, the number and frequency of polls determines if your monthly bill will be $20 or $2000. You need to run some carefully controlled pilot tests to confirm whether your existing software tools are compatible with a high-latency system like cellular. Some software tools work fine when the network is up, but have recovery behavior that multiplies the data moved by 100 or more times. Therefore, make sure you test the data moved during system failures. Remember, you are charged for the data * **SENT** * to your cellular device – even if it is powered off. Therefore an application that tries too hard to stay connected or reconnect is not suitable for use with a cellular network.

Your task is simpler if you plan on occasional access only. You can view the costs much like long-distance telephone costs. Real-world PLC tests show that connecting with programming tools causes from 5k to 25k of data to move per minute. For your average cell plans – assuming you have already used up your "free data bytes" – this works out to be from $1 to $12 per hour to connect. While you would not want to pay $12 per hour to connect for 72 hours (that is, $864), troubleshooting a PLC for an hour or two at $12 per hour is cheaper than either sending an engineer to site or dialing up to an analog modem with normal business-to-business long distance charges.

4 Cellular-Enabling Ethernet Devices

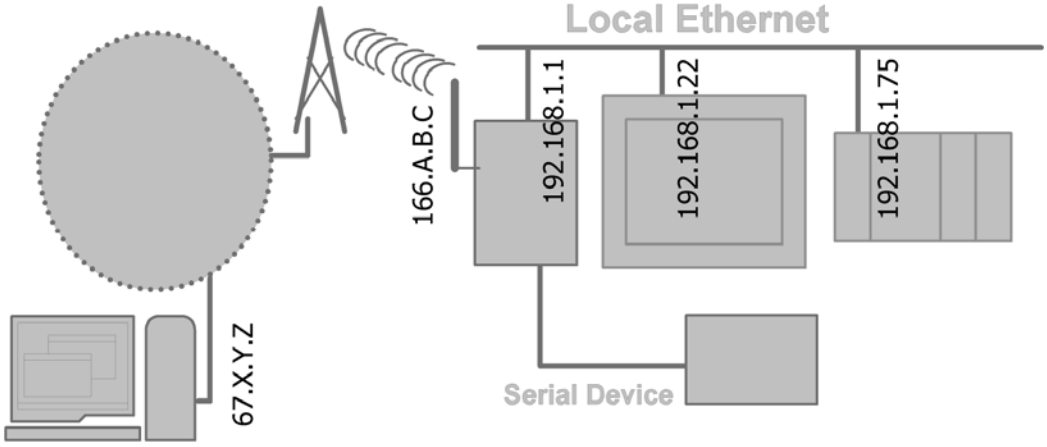
4.1 Overview

The Digi Connect WAN family act much like your home DSL/Cable router. It is assigned an IP address by your service provider (your "cellular ISP"). Outgoing TCP/IP connections are handled with Network Address Translation (NAT), just as your home DSL/Cable router does. This allows any number of local Ethernet devices to connect out into the Public Internet. However, to the Public Internet, the Digi Connect WAN appears as just a single IP address. Therefore, incoming connections must be manually forwarded based on TCP port number to one and only one of the local Ethernet devices. That restriction means it is not possible to have more than one Modbus/TCP device expecting connections on TCP port 502 or web server expecting connections on TCP port 80.



4.2 IP Address Design

Here is an example of IP address assignment in a sample system, followed by a discussion of each set of addresses:



4.2.1 Local Ethernet Subnet (192.168.1.X)

In this example, all devices on the local Ethernet subnet are assigned an IP address in the range 192.168.1.1 to 192.168.1.254 with a subnet mask of 255.255.255.0. This range is defined for "private" use, meaning you do not need to ask permission or pay anyone money to use this range.

The Ethernet port of the Digi Connect WAN/Digi Connect WAN VPN is assigned the IP of 192.168.1.1, and acts as the router/gateway for the subnet. Set your other devices to any ***IP within the range 192.168.1.2 to 192.168.1.254*** and set their ***router/gateway IP to 192.168.1.1***. Unless you are an expert at manual IP route table configuration, you cannot have any other routers on the local subnet.

A Digi Connect WAN family ***can be configured to act as a DHCP server*** for the local subnet. While you likely want to use fixed IP addresses for your field devices, your mobile field technicians will enjoy having this server enabled to allow painless connection of a portable computer when on-site.

4.2.2 Public/WAN IP Addresses

The cellular port of the Digi Connect WAN is assigned an IP address by your service provider, such as 166.213.2.99 or 67.48.210.20 – which are both public IP addresses in this example.

An application on the host computer (IP 67.X.Y.Z) can open a Modbus/TCP connection to the Digi Connect WAN device at IP 166.A.B.C. The Digi Connect WAN device must be configured to forward each desired TCP or UDP protocol to one (and only one) of the local Ethernet devices.

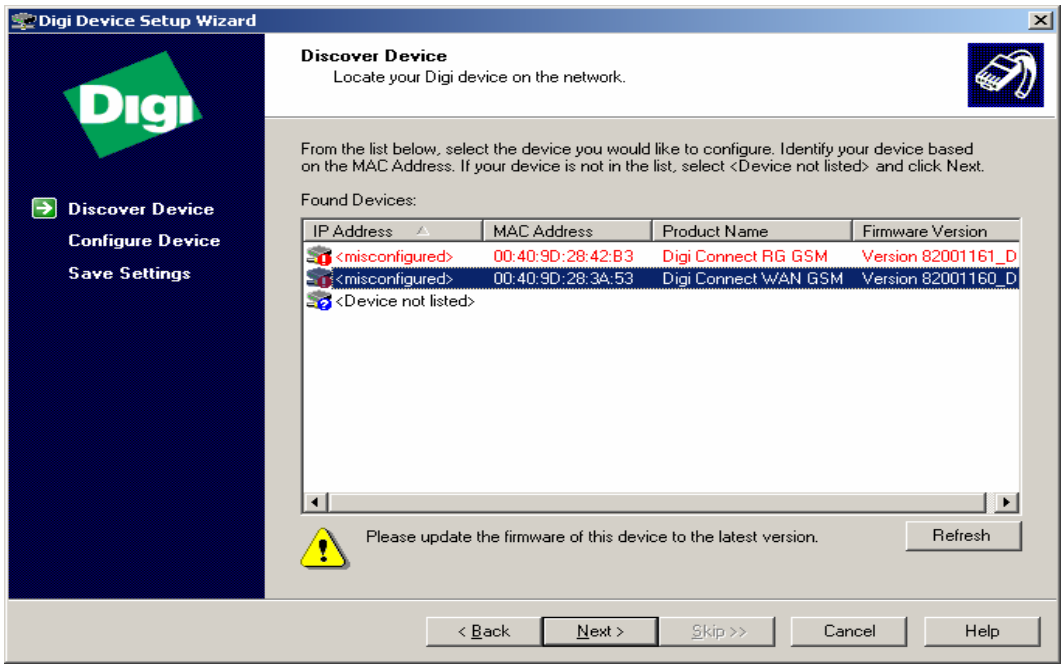
4.3 Configuring the Digi Connect WAN Device

To configure your Digi Connect WAN or Digi Connect WAN VPN, attach both your Digi device and computer to the same Ethernet hub or switch.

4.3.1 Device Discovery and IP settings

Install the Digi Device Discovery tool that is included on the CD with your Digi device on your computer.

The Digi Device Discovery tool uses IP multicast to locate any Digi products connected to your local subnet. There are several factors that may block or affect the device-discovery operation. Some "Personal Firewalls" block this discovery, and some combinations of Ethernet hardware under Windows and "cross-cables" do not allow proper device discovery. If you cannot see your Digi device within the device-discovery results after a few minutes and after pressing **Refresh**, try using an external switch (not a cross-cable) and disable any personal firewall to allow full network access.

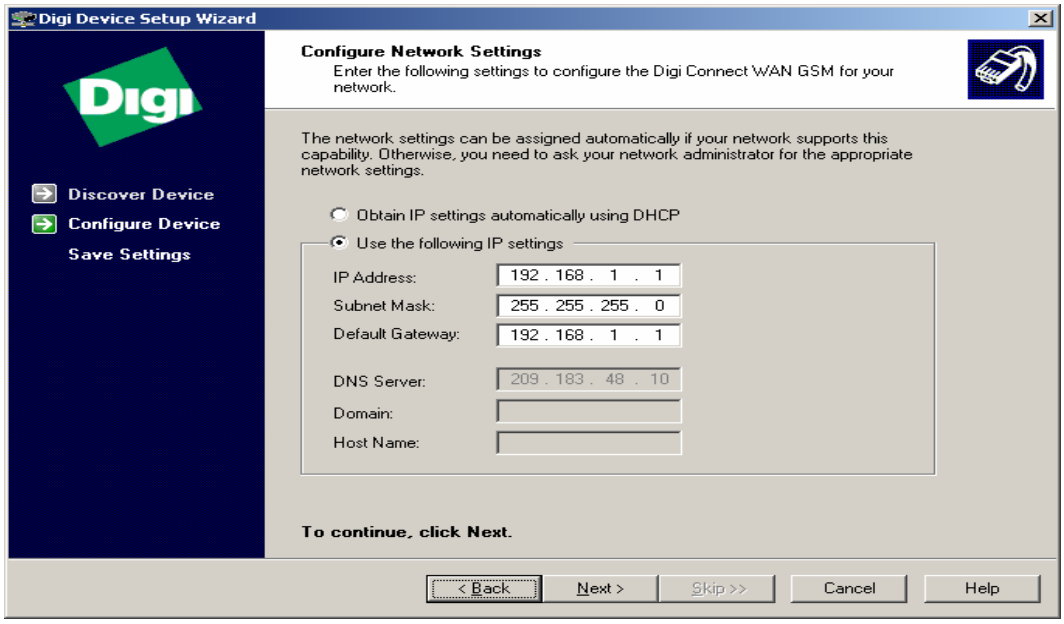


The "<misconfigured>" warning in the device discovery results above is caused by the Digi device having an existing IP address assigned on a different subnet. For example, your PC may have the IP address 192.168.1.201 and the Digi Connect WAN an IP address of 192.168.20.1. This does not prevent the tool from changing the Digi device's IP address information.

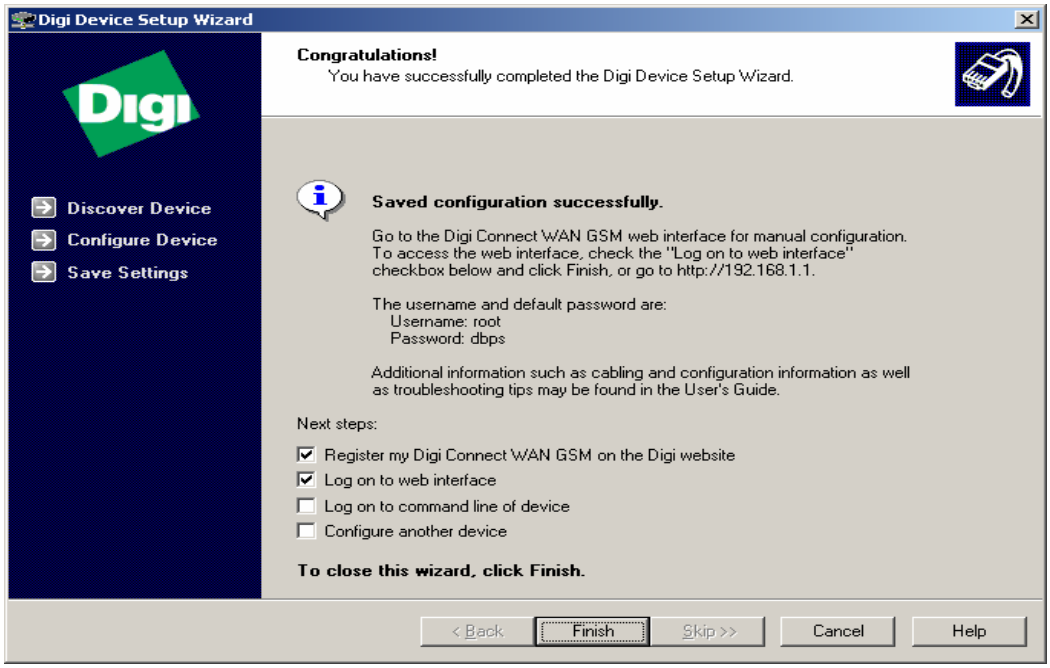
Remember: at this point we are just assigning the IP address used by the *Ethernet port* of the Digi device. The IP address used by the *cellular port* will be assigned remotely by your cellular service provider. Even if you have arranged for a fixed IP address to be used, the ISP will "dynamically" reassign the same IP address every time to your cellular connection.

Select your device to configure – the Ethernet MAC address is shown by each entry – and click **Next**. The Digi Device Setup Wizard is launched. On the Configure Network Settings screen, enter the desired information, such as IP address 192.168.1.1

You can skip the Scenario Settings wizard screen, and continue to click **Next** until the wizard screen titled **Saving Settings** is displayed.

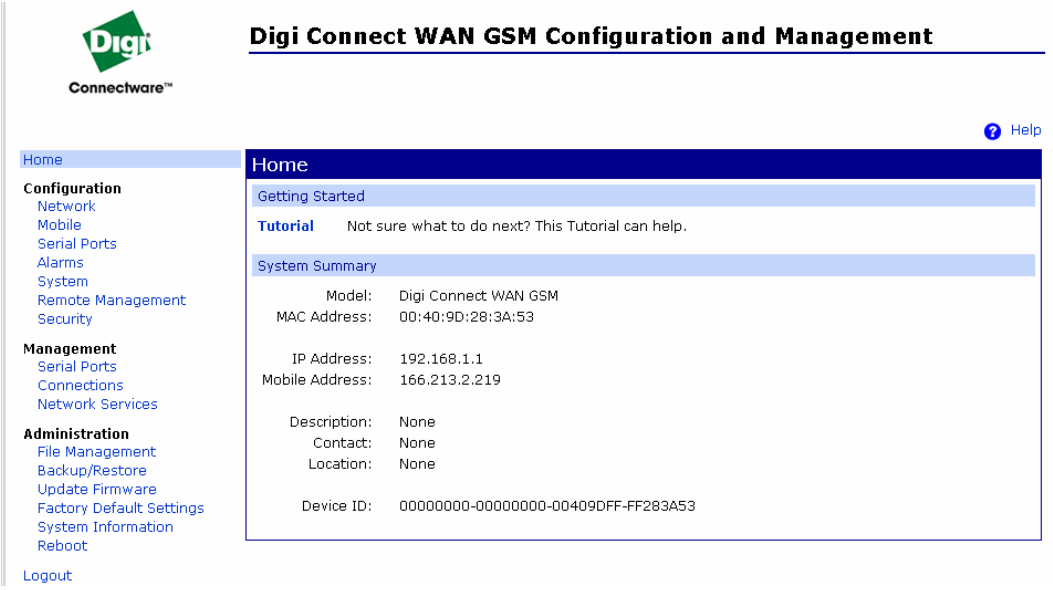


After a minute or two, you should see the **Congratulations** screen below.

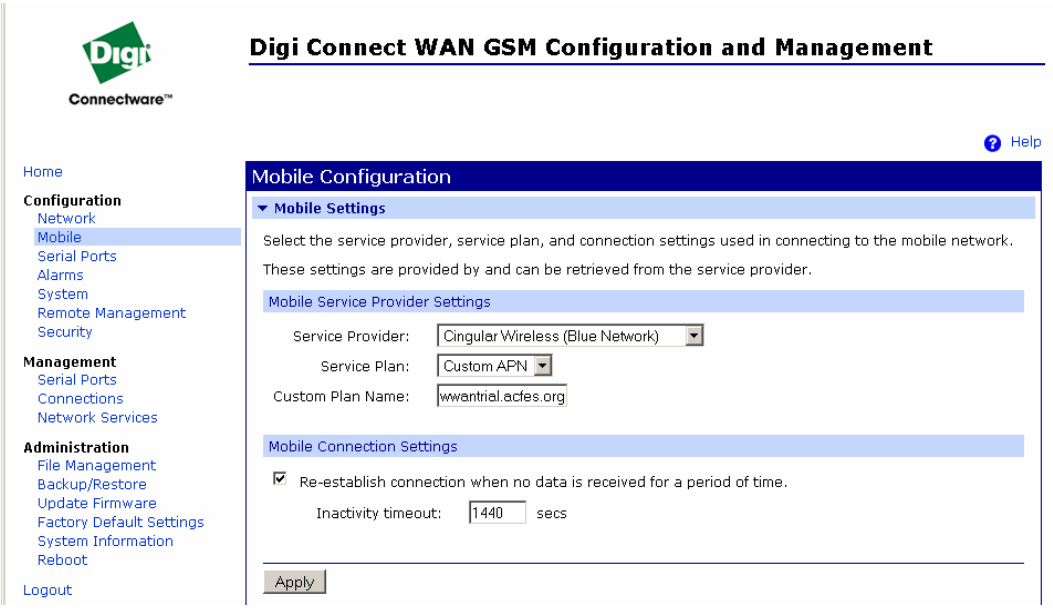


4.3.2 Web Interface and Service Plan Settings

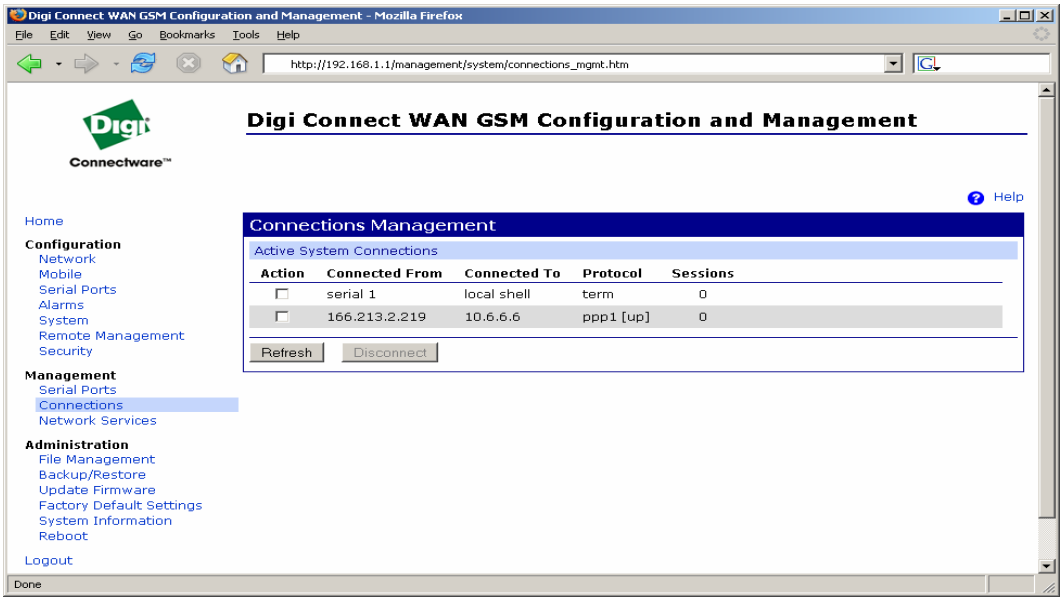
Next, open the Web user interface for you newly installed Digi Connect WAN product. You can either open the Web user interface from the last screen of the Digi Device Setup Wizard, as shown above, or launch your desired Web browser and specify the address of the Digi device. The home page is shown below.



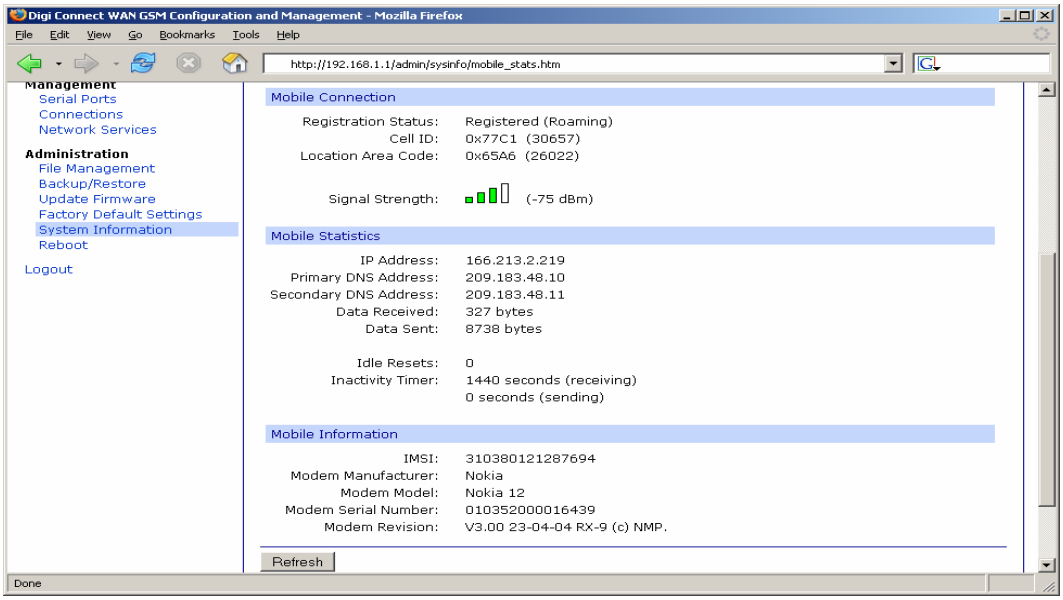
At present, your Digi Connect WAN product is not likely connected to the Internet. Under the **Configuration** menu, click **Mobile.** In the **Mobile Service Provider Settings**, enter the ***information provided by your service provider*** – the information shown below is an example. Pressing **Apply** initiates your Internet connection. Some carriers also require you to access a web site or telephone directly to activate your assigned data plan.



To see your IP status, under **Management**, click **Connections**, and look for the PPP status. **[up]** means you are connected and ready to go. If you see the status cycling between **[init]** and **[connecting]**, this usually means that even though you may have a good cellular signal, you are connected to a roaming partner that does not support the data service required for IP traffic.



Another useful status display is **Administration > System Information > Mobile**. The Mobile page shows your cellular signal, status of the cellular link, and the appropriate IP details if PPP has successfully connected. On this page, you will see the IP address assigned to your Digi device, as well as the DNS address(es) for your field devices to use.



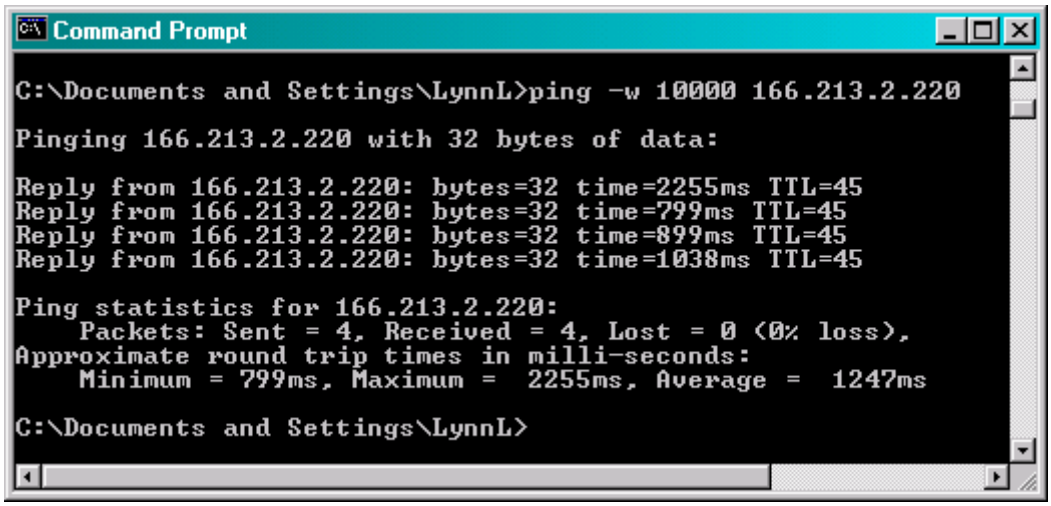
4.3.3 Your devices can connect out to the Internet

As configured, your local field devices can now initiate outgoing connections to the Internet or central servers you maintain. They use the Digi device's IP address as the router to forward the connection; the Digi device uses Network-Address-Translation (NAT) to access the remote resource. For example, if a PLC with local Ethernet address 192.168.1.10 connected to a remote server at IP address 67.43.210.56, the remote server would see it as a connection from the Digi device's IP address (for example 166.213.2.219) and *not* the IP address of your field device.

At this point you could use master blocks in one, two, or more PLC to write data back to a central server or PLC with a public IP address. An HMI panel could initiate SMTP email messages or use FTP to upload files to a central server. No one on the Internet would be able to connect to or bother any of your PLC.

4.3.4 Using the "ping" Command to Test Access

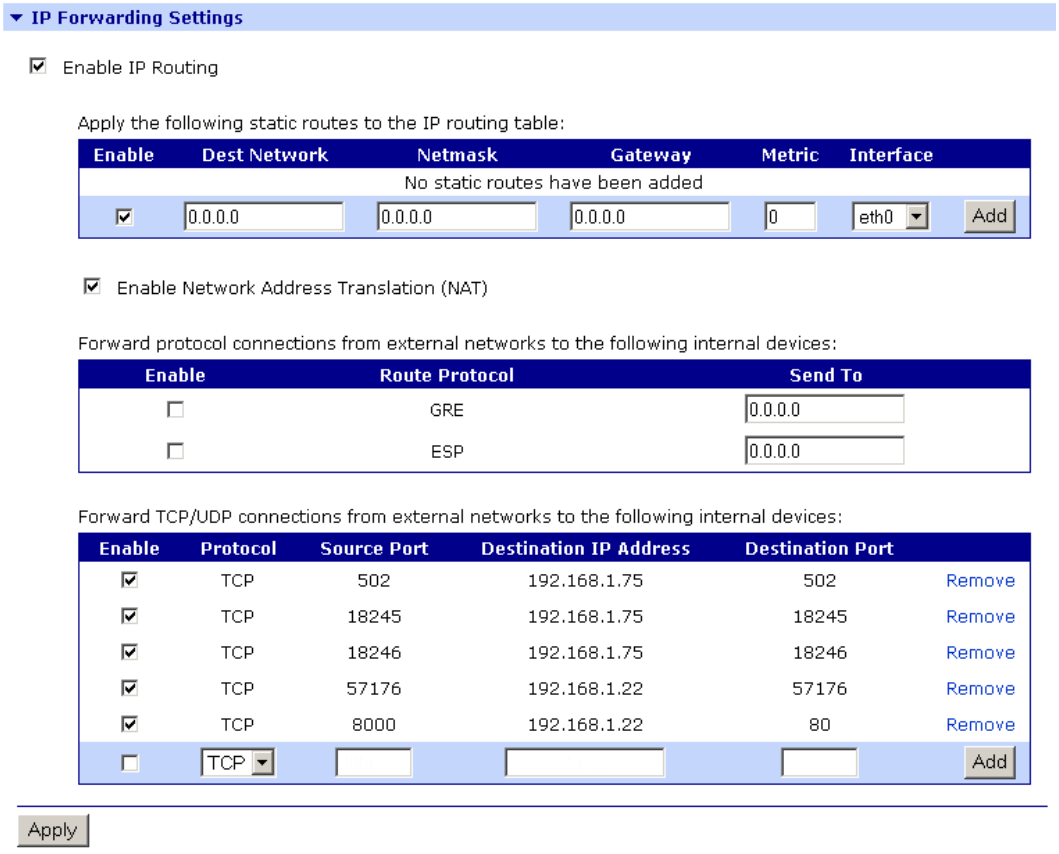
Just like all IP devices, you can use the **ping** command to test access. However, many **ping** utilities assume a short 1-second or 2-second timeout. So use the "-w" option to inform the **ping** command to wait longer for a response; below, "-w 10000" is used to set a 10-second timeout. Notice how the first response is considerably slower than subsequent responses.



4.3.5 Enabling Incoming Access

While outgoing connections to the Internet work with no direct configuration, to enable incoming connections from the Internet requires explicate configuration.

Select **Configuration > Network > IP Forwarding Settings**. This example assumes you have a GE Fanuc PLC at IP address 192.168.1.75 and a Quick Panel at 192.168.1.22. Modbus/TCP uses TCP port 502. GE/SRTP uses TCP ports 18245 and 18246. The Quick Panel configuration uses TCP port 57176 and its web server is on TCP port 80. Note how we "move" incoming web traffic from 8000 to 80 within the Digi Connect WAN. When all entries are added, remember to click **Apply** or you will lose your new settings.

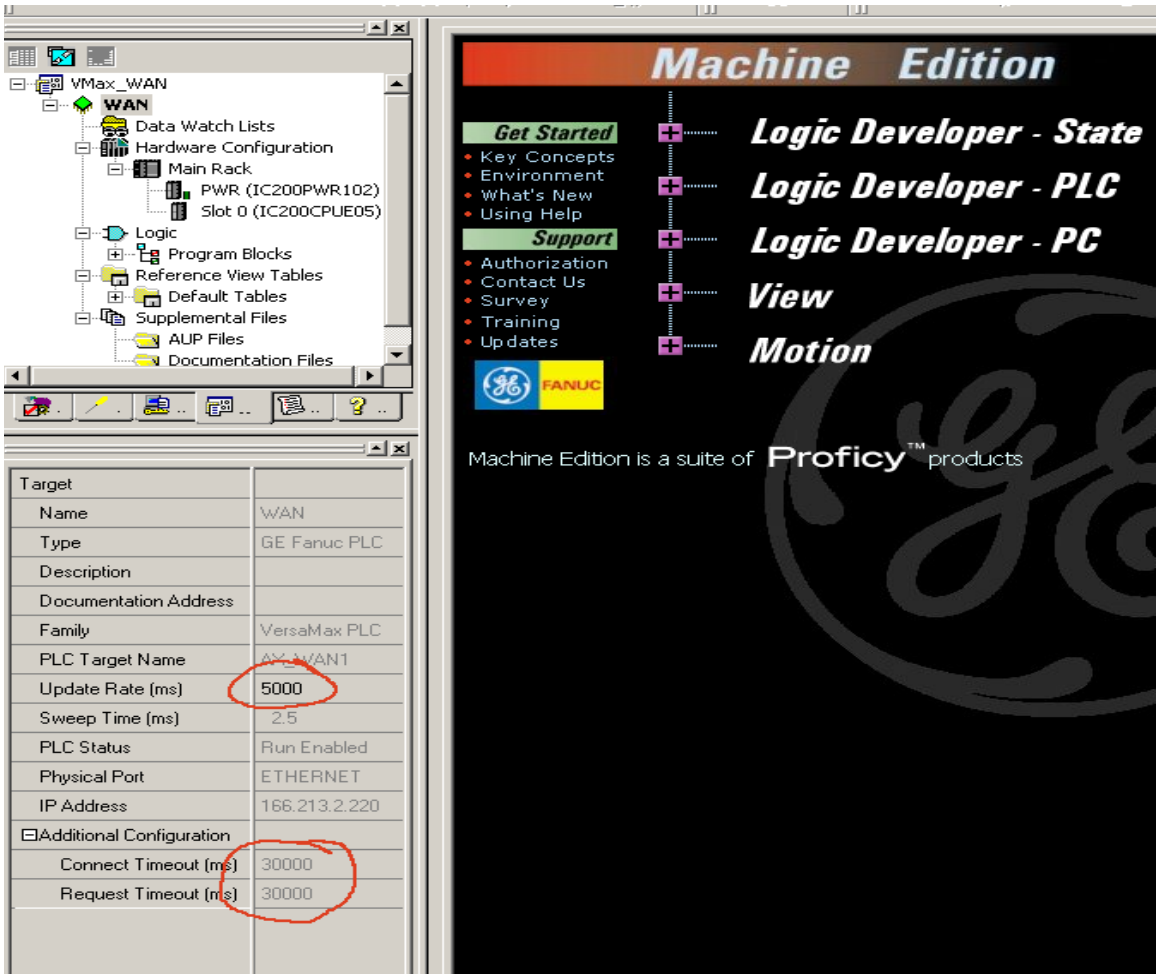


This completes the configuration steps for your Digi Connect WAN product. Now, any incoming Modbus/TCP or GE/SRTP protocols packets received by the Digi will be forwarded to the GE Fanuc PLC. Likewise the Quick Panel can be accessed as a web site on TCP port 8000. Access to port 80 brings up the Digi Connect WAN's web site or nothing if the Digi Connect WAN's web server has been disabled or moved to another TCP port.

4.4 Cimplicity Machine Edition access to the remote Ethernet PLC

Below is a screen shot of ME accessing a VersaMax-E05 via a Digi Connect WAN. You must increase the circled parameters from their defaults. During best-case behavior, cellular networks average round-trip response times in the 1 to 2 second range with a few percent being in the 3 to 5 second range. However, you will see responses above 10 seconds a few times each day, so 30 seconds is a good general response timeout. Setting the timeout too low risks not only doubling your data costs, but is similar to the proverbial "throwing fuel on the fire and hoping it goes out". Higher than normal latency is often a sign of network or cell-tower congestion and attempting to push extra packets through does not help the situation.

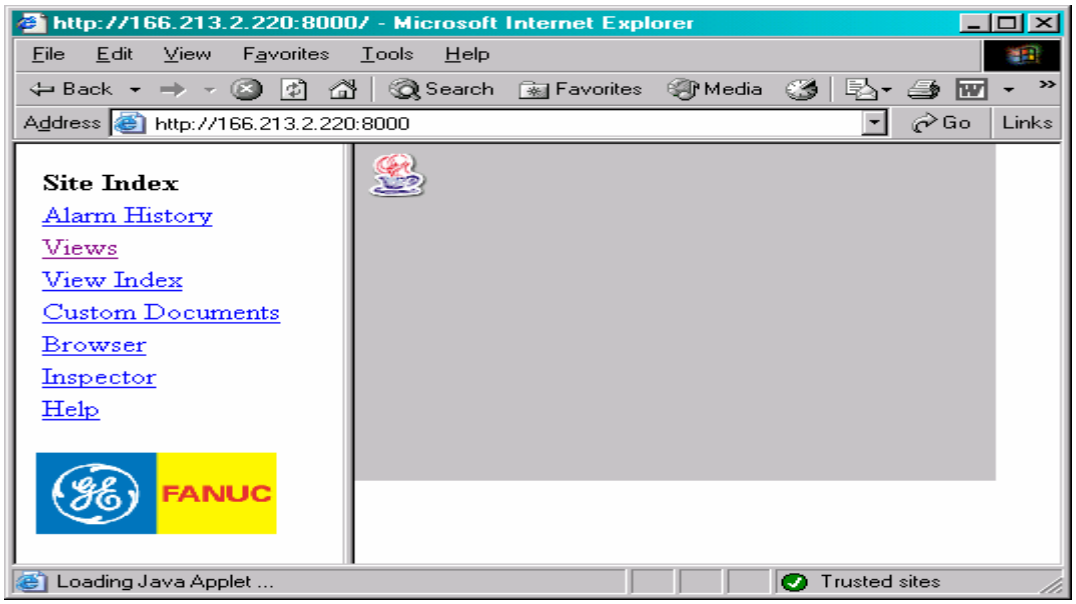
Uploading or downloading a full program will take considerably longer than on direct Ethernet due to the end-to-end latency. While you may have a 100,000 to 3,000,000 bits-per-second connection to the cellular device, the half-duplex poll-response paradigm of most PLC protocols combines with the high end-to-end latency to limit you to a few thousand bits per second of effective throughput.

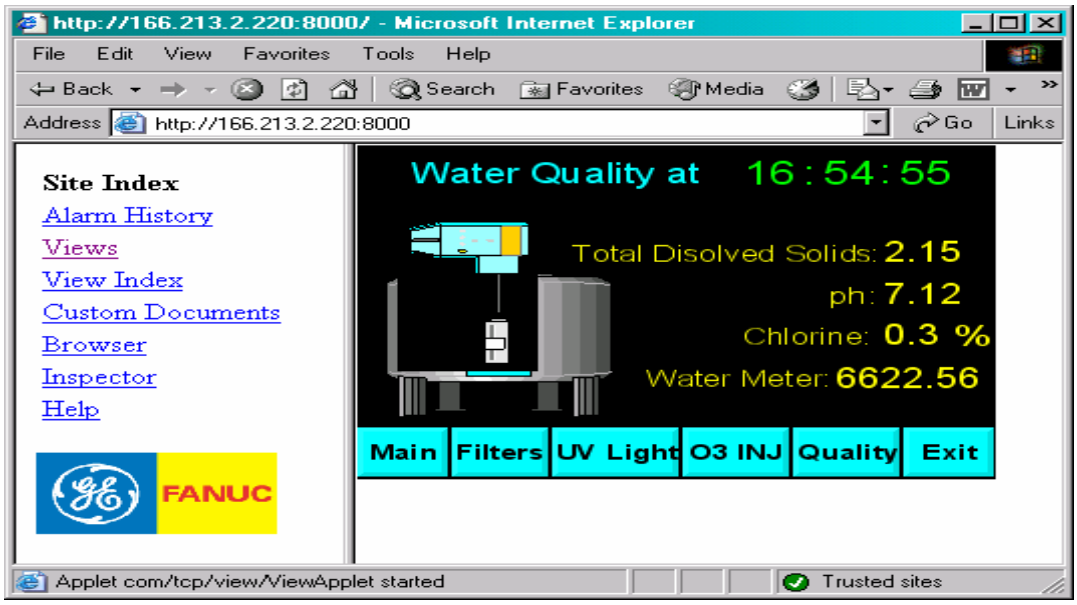


4.5 Web Browser access to a Quick Panel or View Station

In the example configuration above we forwarded TCP port 8000 to a Quick Panel or View Station's web browser. Picking a port other than 80 is wise given that you don't want Google, Yahoo, or other portions of the general "Internet Infrastructure" to treat your web site as a public resource. You must add the ":8000" to the web address.

When first connecting and after selecting "Views", you should see this image as the multi-megabyte Java applet loads – it could take several minutes. ***Do NOT impatiently hit the refresh button*** – this is NOT an elevator! Prematurely hitting the refresh button just restarts the large Java Applet download again – plus you are paying for all of those retransferred bytes.





5 Cellular-Enabling Serial Devices

5.1 Overview

The Digi Connect WAN family works much like a DSL/Cable router for cellular with a built in serial Device Server. This allows both local Ethernet and remote IP-based devices to access a serial device. The Digi Connect WAN offers 3 ways to encapsulate serial data for transmission over an IP-network such as cellular:

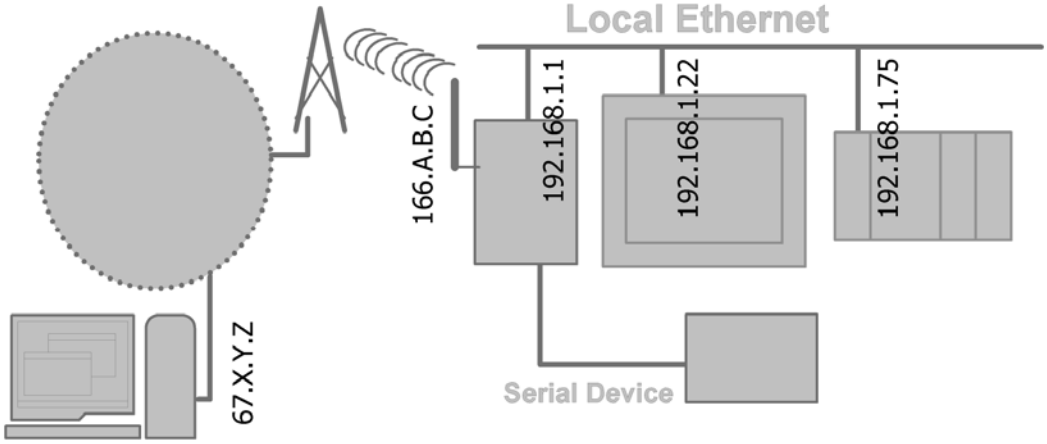
- The raw serial protocol sent within TCP/IP (Modbus or SNPX)
- The raw serial protocol sent within UDP/IP (Modbus or SNPX)
- The serial protocol by Digi RealPort (Modbus, SNP, SNPX)

The first 2 methods require an application which can directly send serial protocol bytes over a TCP or UDP socket or connection. Since only data can be moved, this is limited to Modbus/RTU, Modbus/ASCII, or breakless SNPX. SNP cannot be supported because there is no way to encapsulate the <Long Break> event as raw bytes of data.

The third method uses a special Digi utility called RealPort which mimics a serial comm card for your application. So the application just opens a special COM port (like COM5 or COM24) and Digi RealPort moves the serial data over TCP/IP. Since Digi RealPort wraps the serial protocol within a special protocol, non-data events like <Long Break> can be sent, as well as changes in control signals such as DTR or RTS. Digi RealPort allows use of Modbus/RTU, Modbus/ASCII, SNP (with break), and SNPX (without break).

5.2 IP Address Design

Here is an example of IP address assignment in a sample system:



5.2.1 Local Ethernet Subnet (192.168.1.X)

In this example: to local Ethernet devices, the Digi Connect WAN product appears like a serial Device Server at IP address 192.168.1.1.

5.2.2 Public/WAN IP Addresses

In this example: to remote IP devices the Digi Connect WAN product appears like a serial Device Server at IP address 166.A.B.C.

5.3 Configuring the Digi Connect WAN

Attach both your Digi Connect WAN device and computer to the same Ethernet hub or switch.

5.3.1 Device Discovery and IP settings

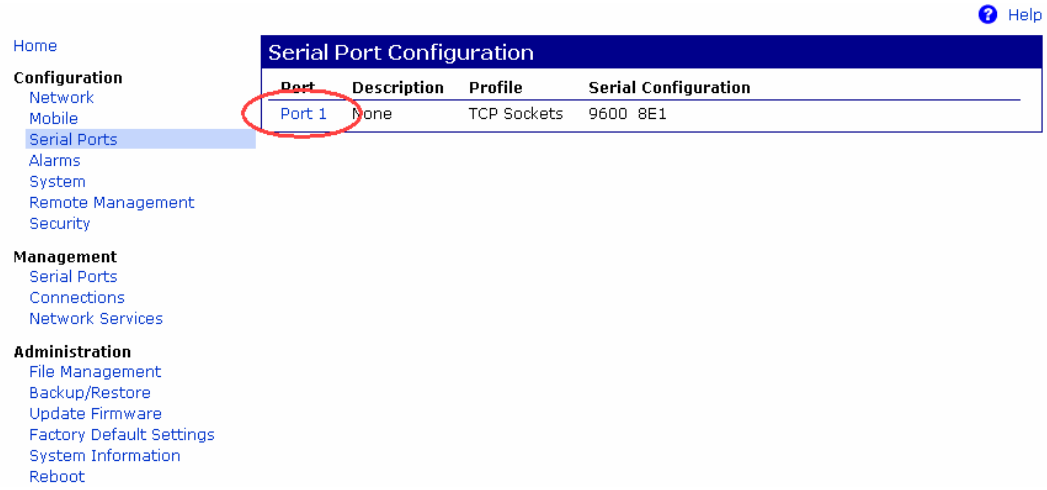
Follow the procedure to set the IP address assigned to the Ethernet port of the Digi Connect WAN, outlined in section 4.3 above. This enables you to use a Web browser to fully configure the Digi device.

5.3.2 Web Interface and Service Plan Settings

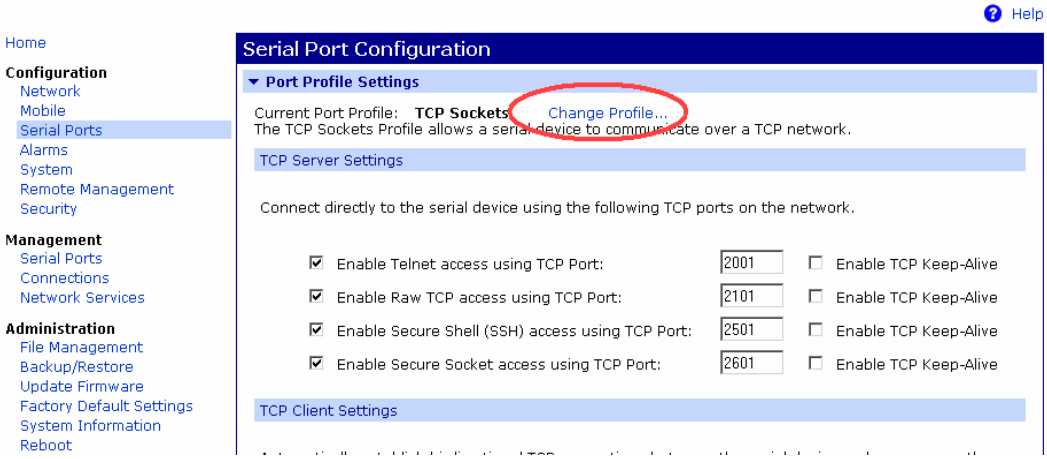
Follow the procedure outlined in section 4.3 above to configure your Service Plan settings for the Digi Connect WAN.

5.3.3 Configure the Serial Port

In the Web user interface for the Digi Connect WAN RG/WAN VPN, click **Configuration > Serial Ports** to see the following display. Click **Port 1** to open the Port Profile Settings page for the serial port.



Click the **Change Profile** link.



Remember to hit the Apply *button always to save your configuration changes!*

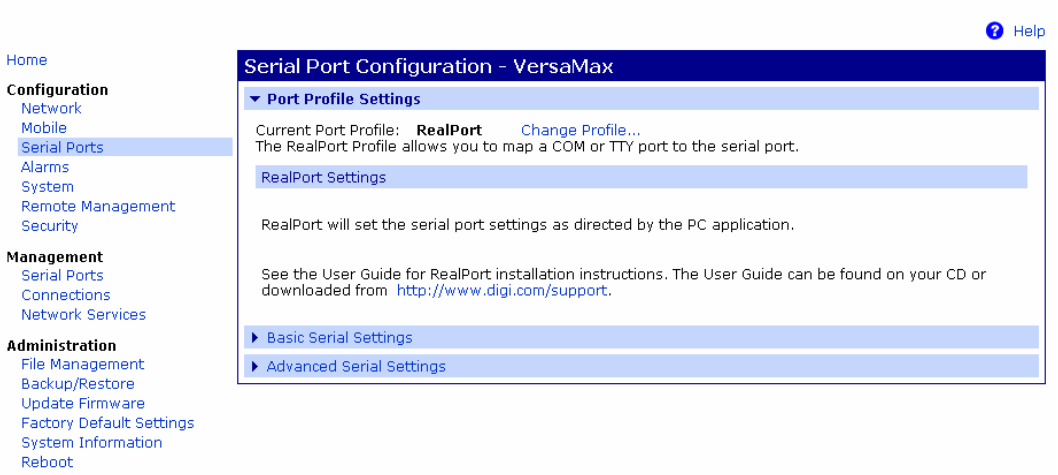
5.3.4 Configure the Serial Port – RealPort Port Profile

If you desire the serial port of the Digi Connect WAN to appear to a remote computer as a physical serial port; select **RealPort** and press the Apply button at the bottom of the screen.

If you do select RealPort, you also need to install the appropriate Digi RealPort driver for your computer's operating system (Windows, Linux, AIX, etc.). There is nothing else you need to configure – settings such as baud rate are automatically forwarded directly from your remote application.

Under the **Basic Serial Settings** section you can add a descriptive text name.

To use Cimplicty Machine Edition, you'll need to select the RealPort Port Profile; this enables serial protocols and long breaks to be transparently encapsulated with TCP/IP to a remote serial port.

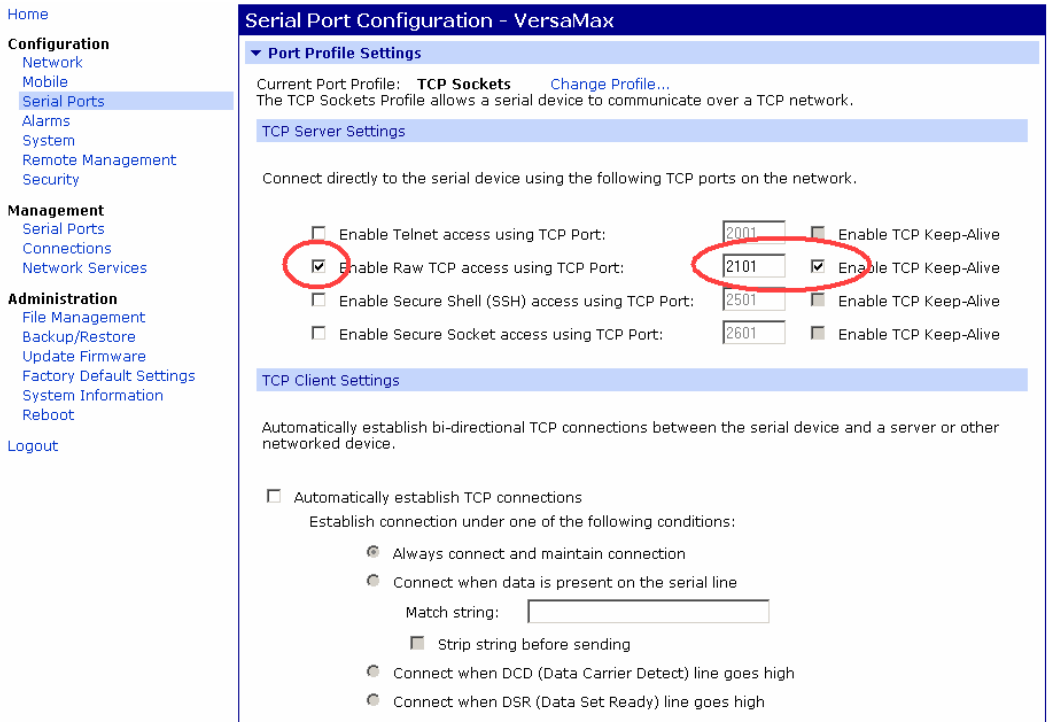


5.3.5 Configure the Serial Port – TCP Sockets Port Profile

The TCP Sockets port profile allows your remote computer to open a TCP socket to carry serial protocol data. For example, some applications can read and write serial Modbus/RTU messages directly into a TCP socket.

Generally, you'll use only the **raw TCP port 2101** service – you can the TCP port number to any available number. A growing number of OPC servers support this and call it "TCP Encapsulation". Modbus/RTU and SNP-X (breakless) could be supported by raw TCP sockets; however SNP (with breaks) cannot be sent by a raw TCP socket. Generally you should disable the other services here.

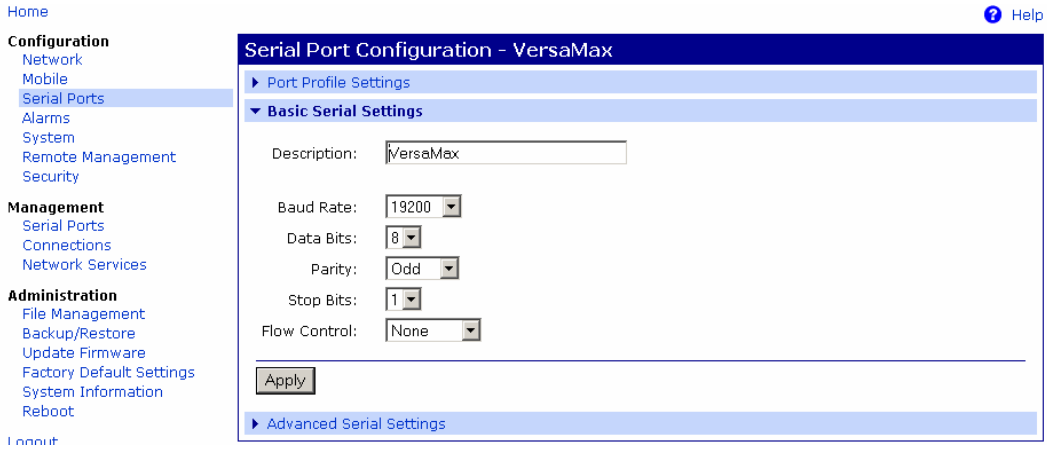
Cimplicity Machine-Edition does not support the TCP Sockets Port Profile – use RealPort instead.



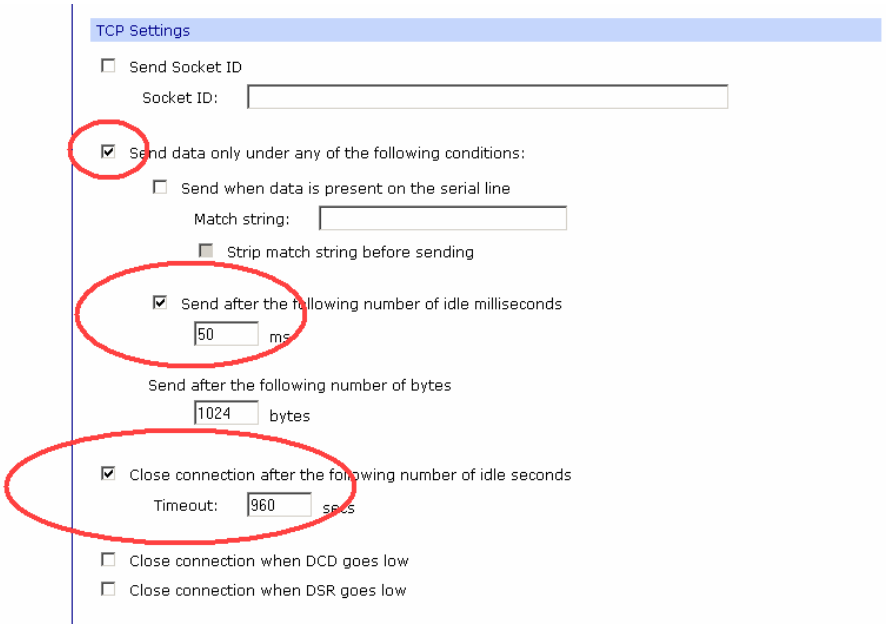
There are many options you can use, including automatically connecting out to remote TCP servers. However, explaining all options is beyond the scope of this basic application note. Select the online help link in the upper right of every page for more information.

Telnet Access (default TCP port 2001) requires your application to handle Telnet commands and handle the duplication of 0xFF bytes. **Secure Shell** (default TCP port 2501) enables your application to use SSH to open an encrypted channel to the Digi Connect WAN. **Secure Socket** (default TCP port 2601) enables your application to use SSLv3/TLSv1 to open an encrypted channel to the Digi Connect WAN.

Click **Basic Serial Settings**. Set the appropriate values for your device; most GE PLC default to **19200,8,O,1** and **Flow Control** must be set to **None**. In the **Description** field, you can enter a useful description of the attached device.



Click **Advanced Serial Settings** and scroll down to the **TCP Settings** section shown. Enable **Send data only under any of the following conditions**. Also enable **Send after the following number of idle milliseconds** and enter the **value 50**. This prevents the Digi device from breaking serial responses into multiple TCP packets and trades off a small amount of speed for fewer data packets and therefore fewer monthly charges. The **Close connection after the following number of idle seconds** is useful to help close broken TCP sockets. This is NOT a TCP Keepalive, but a hard disconnect time. 960 seconds is 16 minutes, which means if no new requests come within 16 minutes, the idle connection is killed as gracefully as possible and a new client can connect freely.



5.3.6 Configure the Serial Port – UDP Sockets Port Profile

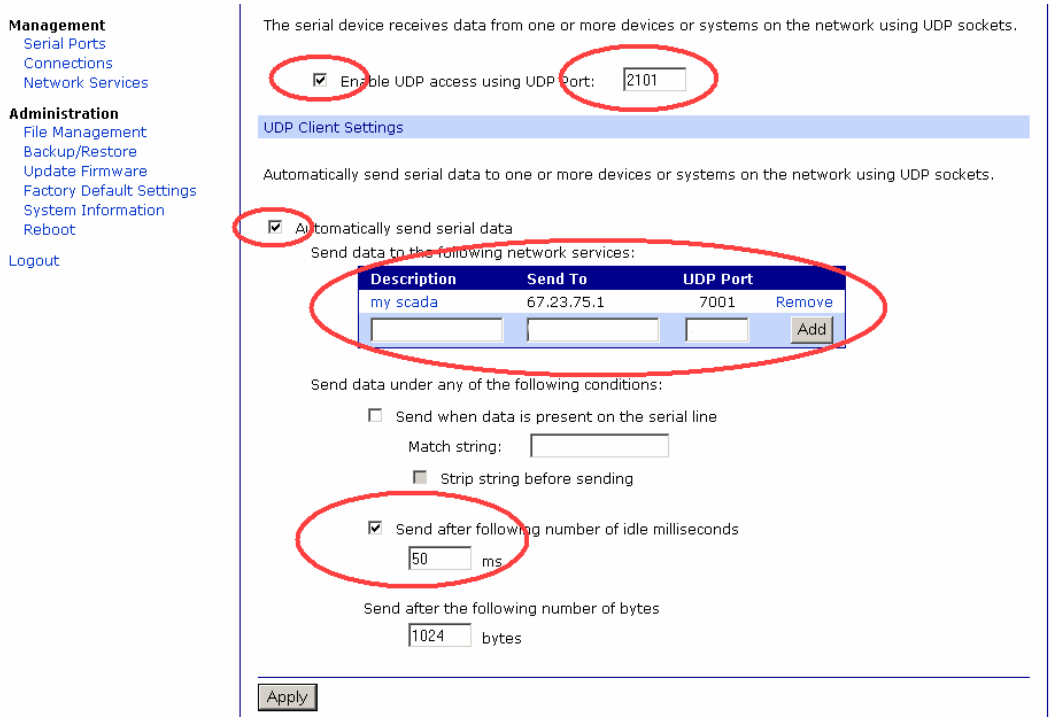
The UDP Sockets port profile allows your remote computer to send UDP packets to carry serial data. It also informs the Digi Connect WAN to which remote IP address any responses should be sent. You need to manually configure the return path for UDP data.

Confirm that the Digi Connect WAN will receive UDP on the correct **UDP port**, which defaults to 2101.

Enable **Automatic send serial data** and add the IP address of your central server, as shown below. More than one server can be defined, which allows for redundant data collection and all server will see a copy of any data received. Also enable **Send after the following number of idle milliseconds** and enter the **value 50**. This helps keep serial responses packaged as single UDP packets.

Note that use of UDP with corporate firewalls is not for the "faint-of-heart". Unlike TCP/IP - which tends to automatically exit and indirectly return transparently through your corporate firewall, ***UDP/IP only works*** *if you have received approval and the firewall-manager manually enables forwarding incoming UDP responses to your host computer*. Some users will find this just a formality; while others will find this an impossible task.

Cimplicity Machine-Edition does not support the UDP Sockets port profile – use RealPort instead.



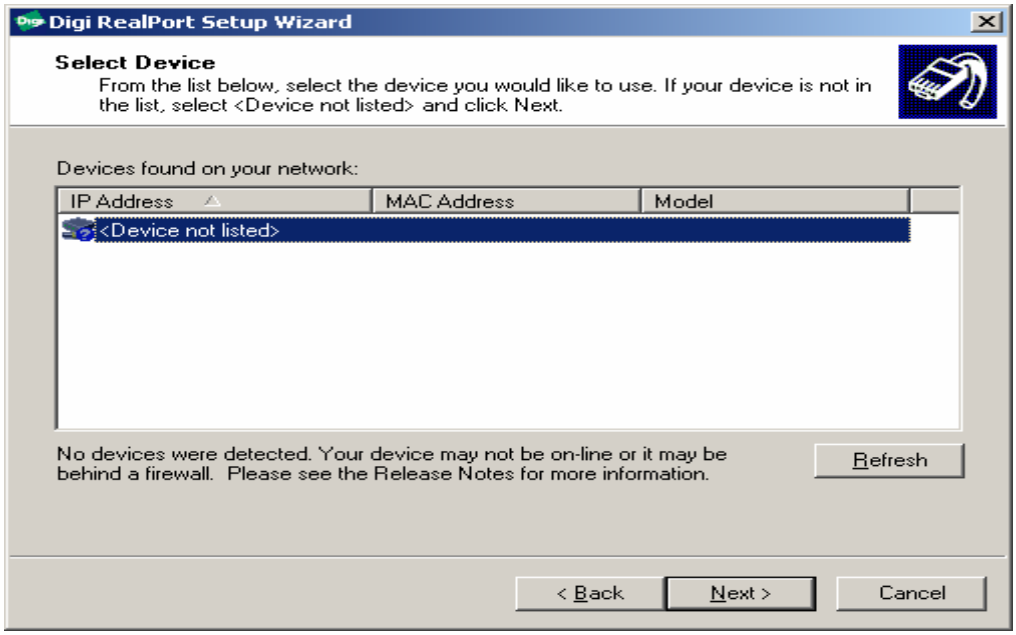
5.4 Install RealPort for the first time

If you have installed a previous version of RealPort, go to the Windows Device Manager and uninstall it. You should use the latest version downloadable from the Digi support web site.

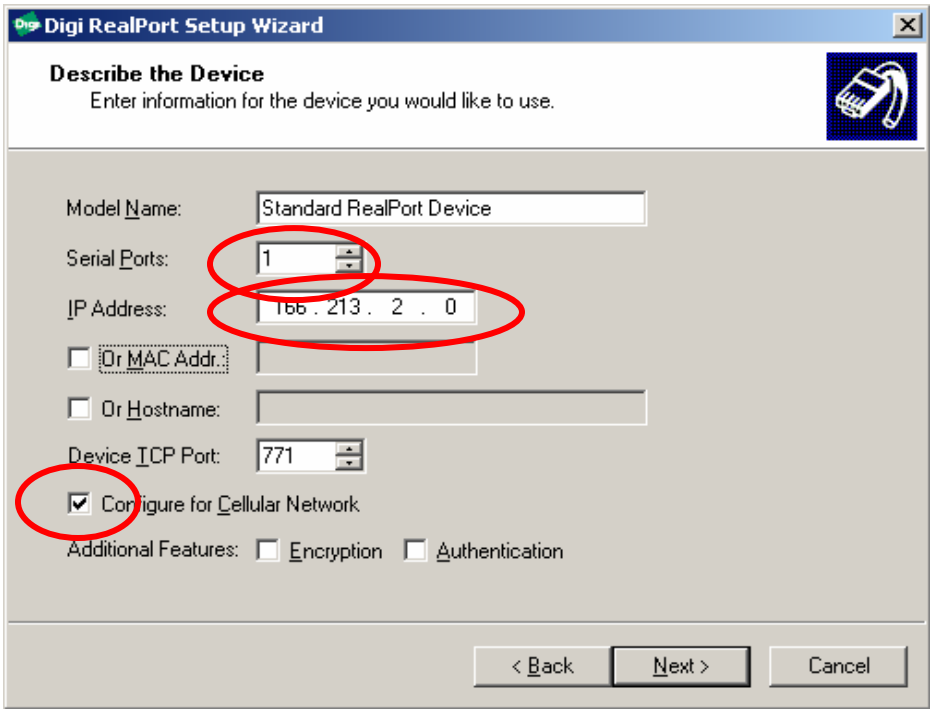
When you run Setup.exe for the first time, the Welcome screen for the Digi RealPort Setup Wizard is displayed. Click **Next**.

By default, RealPort probes your local subnet and finds attached Digi devices. Since the Digi Connect WAN device is remote, it will not be detected.

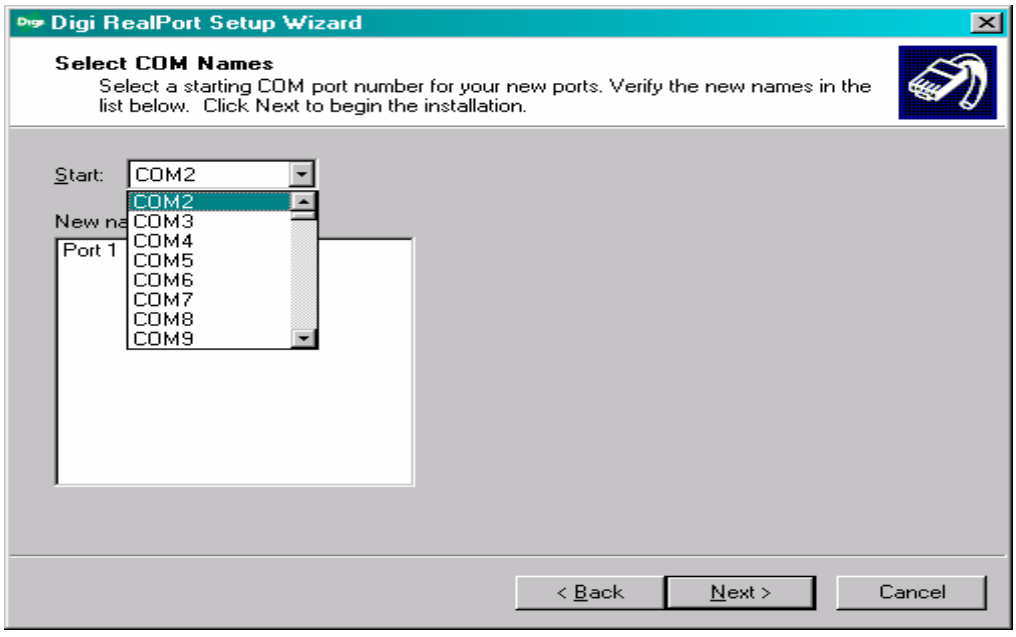
Select **<Device not listed>** and click **Next**.



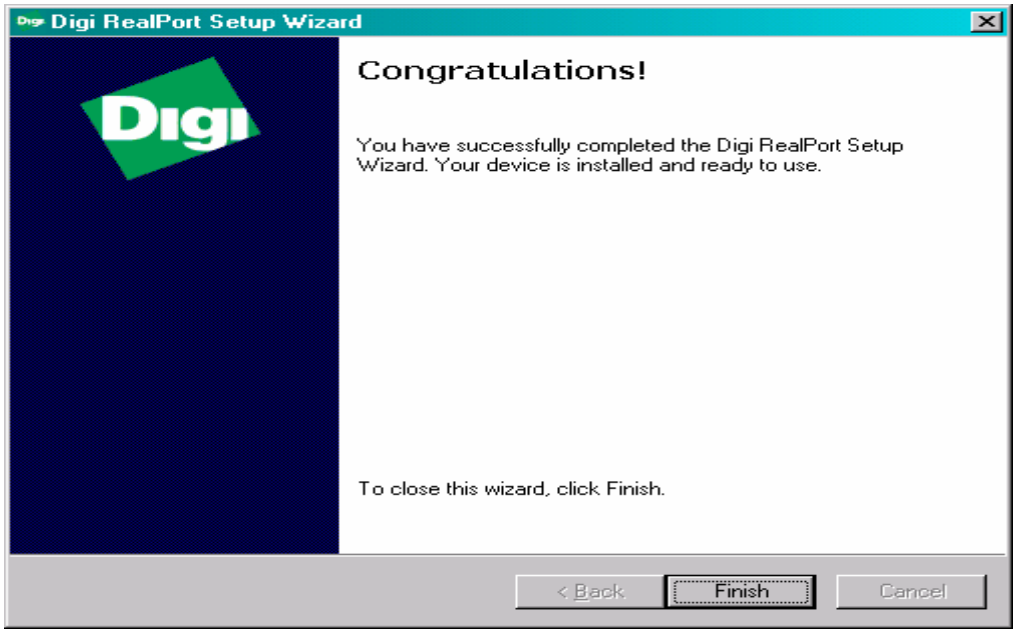
Set the **Serial Ports** to one (1) and enter the **IP Address** or DNS hostname assigned to the Digi device. Leave the **TCP Port** set at 771. Select to **Configure for Cellular the Networks**.



Associate this RealPort connection to any available COM port name, such as COM2 or COM31. The rest of this document assumes that you select COM2. You will only see available names reported by Windows. RealPort *cannot* "replace" an existing COM port; so, for example, COM1 will only be shown if you remove or disable the computer's built-in serial port. Click **Next** to continue.



The RealPort drivers are now installed. Since RealPort will be connecting via the Public Internet to the Digi device, the actual installation may take several minutes. When installation completes, the following wizard screen is displayed.

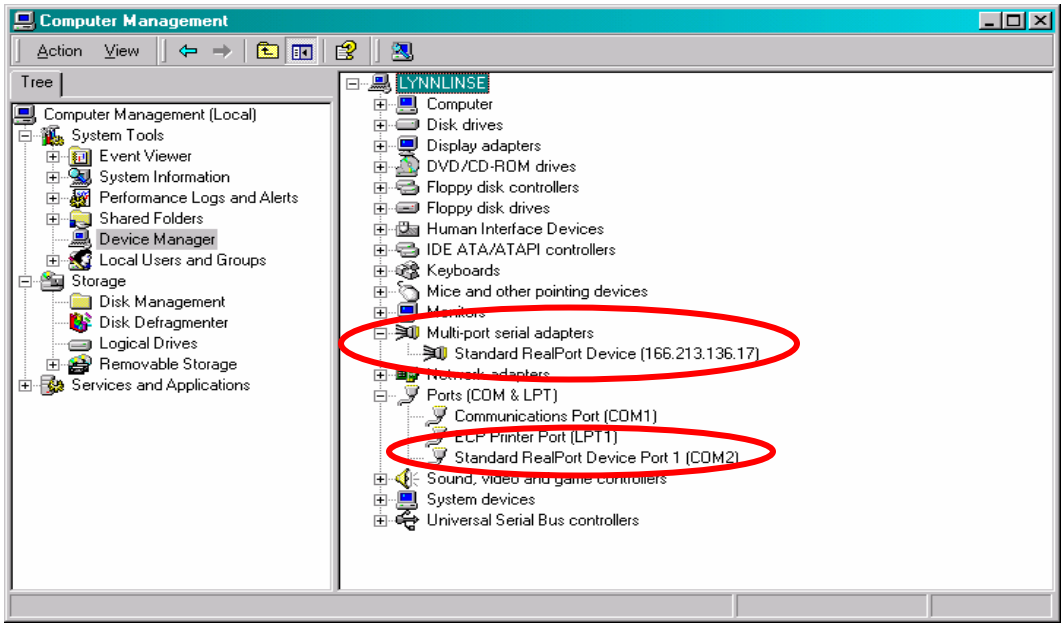


After the wizard completes, you need to configure RealPort to tolerate the wide-area network conditions of a cellular data link. Without these special settings, your application will talk but will frequently go offline.

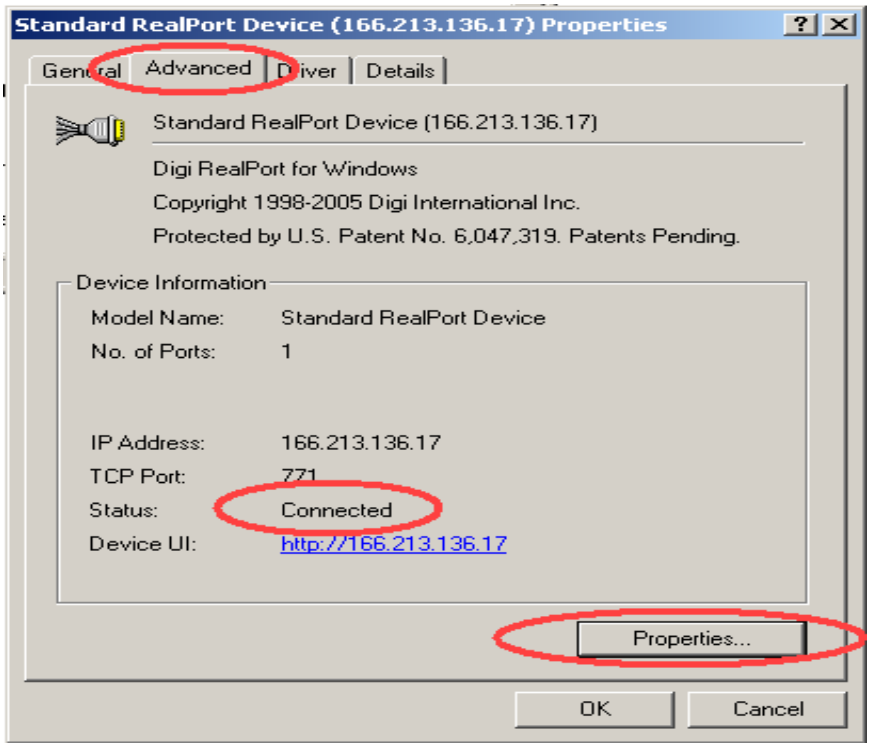
5.5 Configuring RealPort for Wide-Area Network conditions

Run the Windows **Device Manager** by right-clicking the **My Computer** icon on your desktop and selecting **Manage**. In the Device Manager display, you will notice two new pieces of "hardware:"

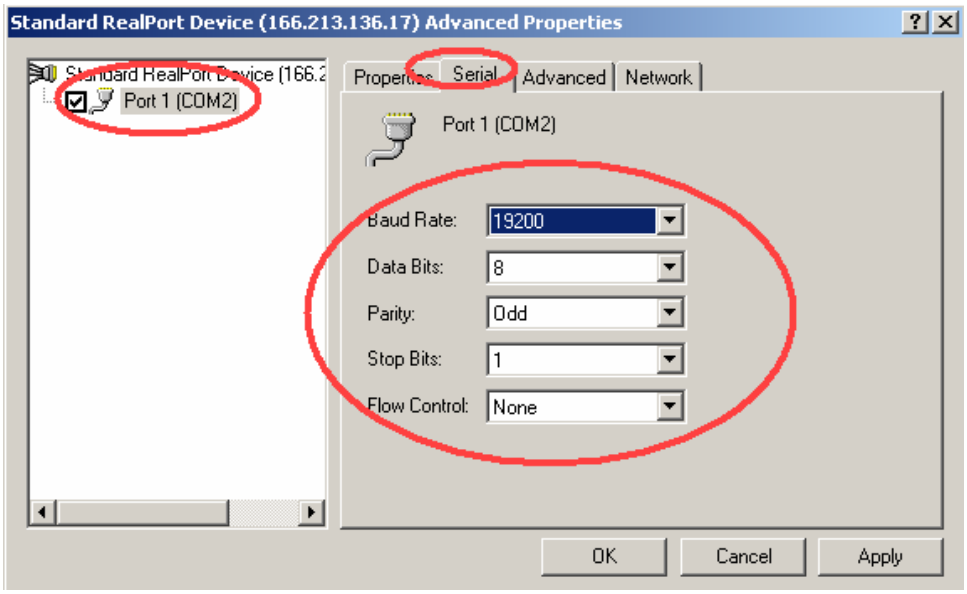
- Each device RealPort connects to has an entry under **Multi-port serial adapters**; you adjust RealPort settings here.
- Each RealPort "COM port" has an entry under **Ports (COM & LPT)**. Applications treat the RealPort (COM2 in this case) as any other serial port.



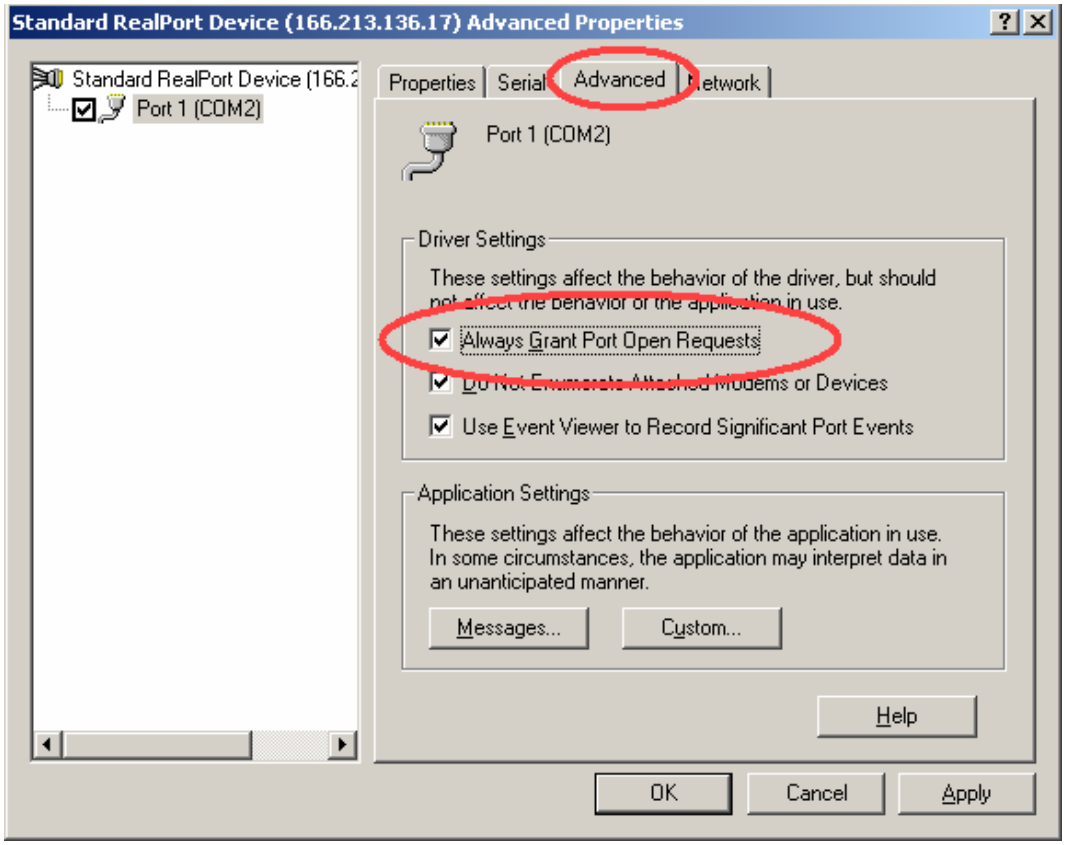
Right-click **Standard RealPort Device** and select **Properties** in the drop-down menu. Notice the **Status** line; this says *Connected* if a standard TCP/IP socket has been established to the remote Digi Connect WAN. If it says *Reconnecting* or *Disconnected* then you have a configuration problem or something is wrong with the cellular link. On the **Advanced** tab, click **Properties**.



Select Port 1 and the Serial tab to set your default port settings – these may be automatically forwarded to the Digi Connect WAN. However, most applications will override these values when they open the port.

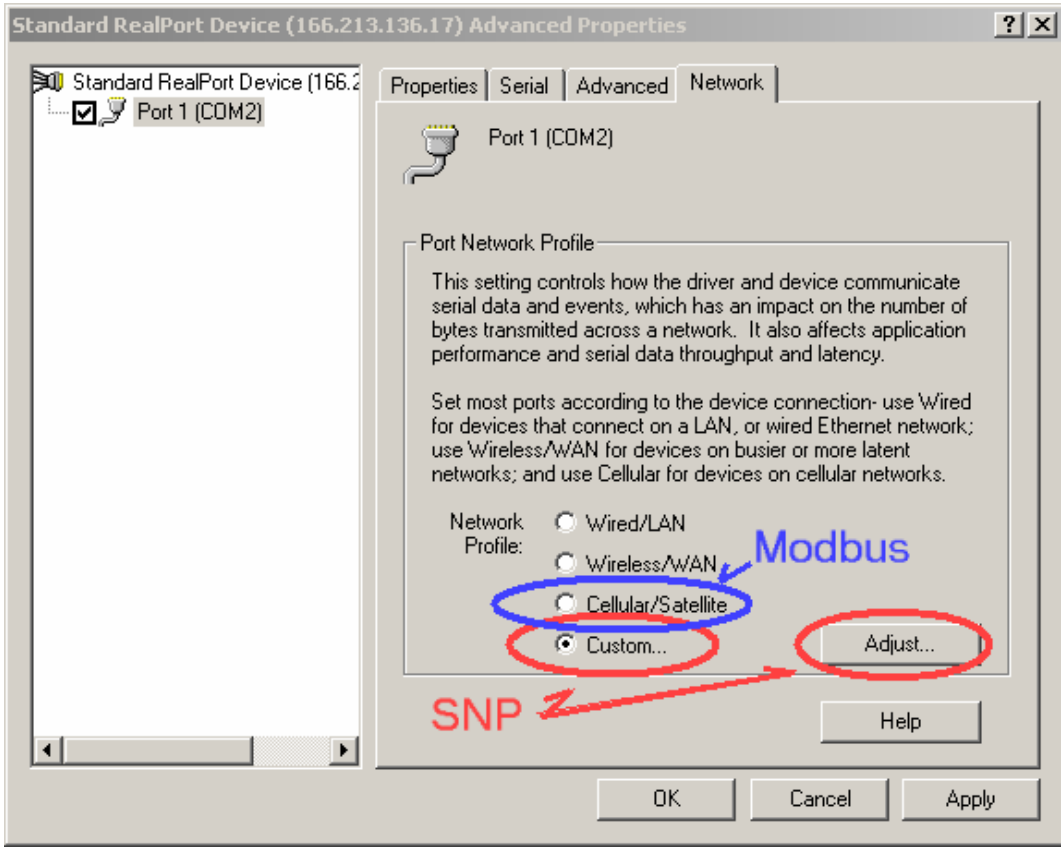


On the **Port 1** and **Advanced** tab, you may want to try enabling **Always Grant Port Open Requests**; this allows your application to always open the RealPort COM2 regardless of the availability of cellular access to the Digi Connect WAN. By default this is off and the serial port can ONLY be opened if the TCP/IP connection is active to the Digi Connect WAN. Some applications do not retry if the serial port fails to open.



On this final Network Tab, the 3 main network profiles can be summarized as:

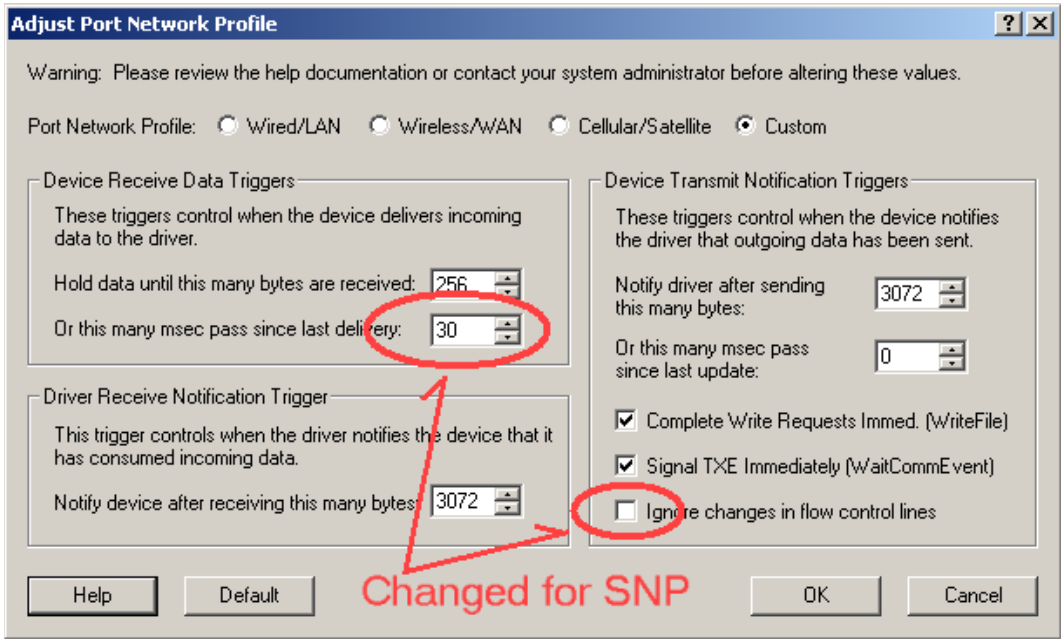
- **Wired/LAN** means direct Ethernet where packet count is less important than raw performance. RealPort will move many small data packets to minimize end-to-end latency. With a system like cellular where you pay for your traffic and all TCP/IP overhead this is a very bad setting.
- **Wireless/WAN** means systems with moderate latency (50 to 100 msec). RealPort attempts to reduce packet count and tolerate more delays. The "raw performance" assumptions of Wired/LAN just won't accomplish the same result with these routed networks and is wasted effort.
- **Cellular/Satellite** means systems with high latency (800 to 30,000 msec) where packet reduction is critical. This setting also disables control signal information (RTS, CTS, DTR, etc) and sets data writes to return immediately to your application and not wait for confirmation from the Digi Connect WAN that all bytes have been transmitted.



For **Modbus** select the default **Cellular/Satellite** setting; this combines settings shown to work best at lowest cost for most poll-response protocols.

For **SNP** and **PLC programming**, select the **Custom** setting; we need to tweak some settings that will increase your cellular costs but are required to – for example – handle SNP's <Long Break> event and the SNP <ACK>.

For **SNP** and **Cimplicity Machine Edition** we need to make 2 changes. First, reduce the **many msec pass since last delivery** value from the default to only 30. This improves handling of the SNP <ACK>, but causes larger SNP responses to be fragmented into multiple TCP packets. Second, we need to uncheck the **Ignore changes in flow control lines** to enable support for the <Long Break> event. This also causes RTS/CTS and other control signal information to be moved by across the cellular link regardless of value.



When you press Ok on all open dialogs and return to Windows Device Manager, ***it may appear to "hang"***. Just be patient, and in a few minutes, it will refresh its screen and can be closed. This delay occurs because RealPort needs to close the old cellular connection, reconfigure, and reopen a new cellular connection. If RealPort is active on the port, you may be asked to reboot the computer.

6 Removing RealPort

You do not want to leave RealPort active when you no longer desire access to the remote device. Because Digi RealPort will automatically attempt to connect to the Digi device every time you run Windows, it will impact your computer needlessly if left active. It will impact your cellular data bill also.

6.1 Removing specific "adapters" from with Windows Device Manager

Right-click the device and select to **uninstall**. Windows asks you to confirm the uninstall, then removes the selected Digi device. In our example above this was called **Standard RealPort Device (166.213.136.17)**.

6.2 Removing all from the Command Prompt

Alternatively, you can run the original **Setup.exe** command line using the option **/removeall** to remove all Digi RealPort drivers installed.