Remote Cellular TCP/IP Access to Modbus Ethernet and Serial Devices

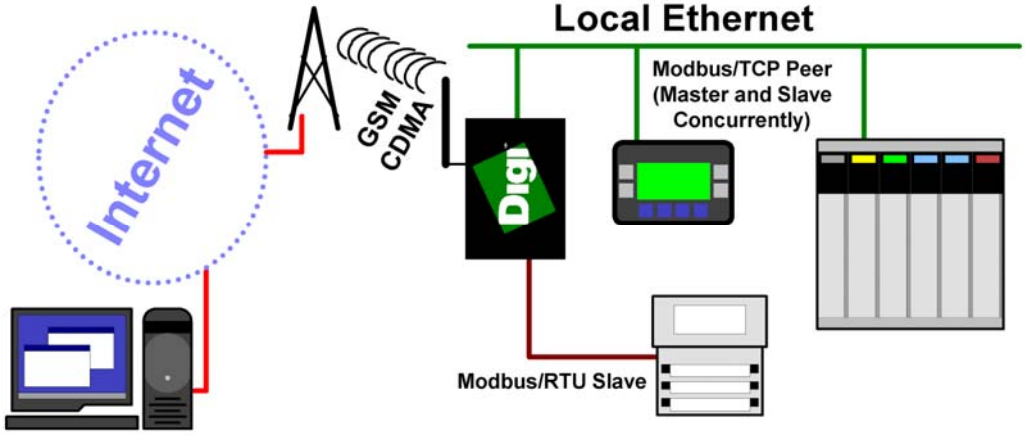
Keywords: Cellular, Modbus

Abstract: This document describes how to setup the Digi Connect WAN IA Modbus Bridge for remote cellular TCP/IP access to Modbus serial and Ethernet devices. The Digi Connect WAN IA functions much like a home DSL/Cable modem, except the connection is by 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 features of the Digi Connect WAN IA, consider the following example:



Key Features

The Digi Connect WAN IA has several key features:

- Accepts incoming Modbus/TCP, Modbus/RTU, and Modbus/ASCII encapsulated within either TCP/IP or UDP/IP, and bridges this to local devices supporting Modbus/TCP, Modbus/RTU, and Modbus/ASCII.
- Like a home router, provides outgoing Network-Address-Translation (NAT), incoming TCP and UDP port forwarding, and can act as a 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 on the Digi Connect WAN IA, including HTTP (web browsing), ODVA Ethernet/IP, and Modbus/TCP.

- Existing applications such as 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 Features Supported by Digi Connect WAN Family Products

The Digi Connect WAN family includes four models to better target your needs. This document (90000773) covers use of the Digi Connect WAN IA as a Modbus Bridge. Document 90000774 covers use of the Digi Connect WAN VPN, Digi Connect WAN RG, and Digi Connect WAN products with non-Modbus equipment. Here is a brief comparison of the product features:

Feature ***(See the referenced Digi Document for further details)***	**Digi Connect WAN IA** ***This One***	**Digi Connect WAN VPN** ***90000774***	**Digi Connect WAN RG** ***90000774***	**Digi Connect WAN** ***90000774***
1) Modbus/TCP to serial bridge	Yes	No	No	No
2) Remote TCP/IP connection to local Ethernet-enabled equipment	Yes	Yes	No	Yes
3) Local Ethernet-enabled equipment can use TCP/IP protocols out to remote servers	Yes	Yes	No	Yes
4) VPN end-point securely "grafts" local Ethernet onto remote network	Yes	Yes	No	No
5) Remote access to local serial port by raw TCP, UDP, SSH, or SSL/TLS	If Modbus Bridge is off	Yes	Yes	Yes
6) Enables remote console management of routers and servers	If Modbus Bridge is off	Yes	Yes	Yes
7) Interacts with standard routers for redundant (backup) paths	Yes	Yes	No	Yes
8) Digi RealPort® supports legacy serial-only applications	If Modbus Bridge is off	Yes	Yes	No
9) Digi Configuration by remote, Ethernet, or serial connection	Yes	Yes	Yes	Yes
10) Digi acts as a local DHCP server	Yes	Yes	No	Yes
11) Supports EIA-232/422/485	Yes	Yes	Yes	EIA-232

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: The first release of Digi Connect WAN IA firmware does NOT support incoming Modbus/RTU or Modbus/ASCII by Digi RealPort when the Modbus bridging engine is enabled.

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

The Digi Connect WAN IA allows remote TCP/IP clients to access local Ethernet devices by TCP or UDP port forwarding. Since the Digi Connect WAN IA 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 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 IA 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 MAST 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 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 the security and 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

If the Modbus Bridge function is disabled, the Digi Connect WAN IA allows 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, the 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 would 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

If the Modbus Bridge function is disabled, the Digi Connect WAN IA 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 one of the land lines goes down, network maintenance people cannot access the router by network to troubleshoot. However, the cellular link through the Digi Connect WAN IA 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 IA 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

If the Modbus Bridge function is disabled, the Digi Connect WAN IA 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 IA can be configured either remotely, by direct Ethernet, or by RS-232 connection.

2.10 Acts as local DHCP server

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

2.11 Supports EIA-232/422/485

The Digi Connect WAN IA has a DIP-switch configured MEI (multi-electrical interface) serial port that supports EIA-232, EIA-422 (four-wire) and EIA-485 (two or four-wire).

3 Performance Expectations

3.1 LAN and WAN Differences

In theory, any TCP/IP- 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) Response Delay: how long to "wait for a response"	Normal: less than 0.2 sec. Maximum: assume 1 or 2 seconds is failure.	Normal: 1 to 3 seconds. Maximum: must wait at least 30 seconds before assuming failure.
3) Idle TCP sockets	TCP sockets can sit idle indefinitely; limited only by application protocol expectations.	Many WAN systems ungracefully interfere with idle TCP sockets; they stop working without either end seeing a close, abort, or reset.
4) 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; however real world tests show loss is less than 1% over cellular networks.
5) 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.

3.2 Will my application work?

Unfortunately, most product developers only test on Ethernet/LAN. It has been Digi's experience that the first users attempting to use WAN with a vendor's tools and products will have to locate and point out the problems for the vendor.

3.2.1 Connection Delay

Connection delay is likely the largest problem you will have. Most applications use the OS defaults – on Windows, this connection delay generally is 2 or 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 open may still time out too fast.

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 will eventually recover.
- 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 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 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, which timeouts twice and is retried 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.

3.2.3 Idle TCP Sockets

Idle TCP Sockets are related to item #5 (Cost to Communicate). The obvious solution to reducing cost is to slow down data polls. However on current cellular networks, TCP sockets idle longer than 5 minutes become "unreliable." The TCP sockets are unreliable not 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. This activity is in vain, as the only solution will be to abort (not close) and then reopen the socket. In addition, all of these retry packets may be incurring charges.

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.

3.2.4 UDP Reliability

UDP reliability may seem like a moot point, in that, as defined 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.

3.2.5 Cost to Communicate

Few applications are written to optimize network traffic; after all, it is usually the end devices themselves and not the "Ethernet" which is 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 data. Here are some example monthly data usages, based on 200-byte transactions.

- 200 bytes per second = 518Meg/month
- 200 bytes per 5 seconds = 86Meg/month
- 200 bytes per minute = 9Meg/month
- 200 bytes per hour = 0.14Meg/month (treated as 0.78Meg due to round-up)

Remember the issue above requiring TCP sockets to move data every 4-5 minutes. Ultimately, to minimize cost, applications may need to be rewritten to implement Report-By-Exception or Change-of-State – preferably by UDP/IP.

3.3 IP Address Considerations

In general, there are three types of "service plans" for IP address assignment that you can contract. However, not all carriers provide all three options.

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 response.

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 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, it is not the data that reaches your cellular device for which you are charged. Instead, you are charged for the data that is * **SENT** * to your cellular device regardless of whether your device is even powered up. Therefore an application that tries too hard to stay connected or reconnect is not suitable for use with a cellular network.

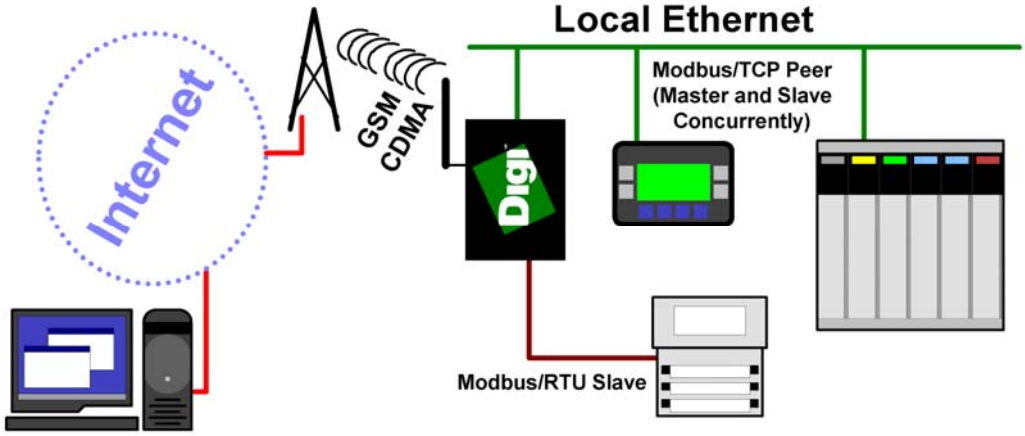
Your task is simpler if you plan on occasional access only. You can view the costs much like long-distance telephone costs. Real-world PLC tests show that connecting with programming tools causes from 5k to 25k of data to move per minute. For your average cell plans – assuming you have already used up your "included kilobytes" – 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

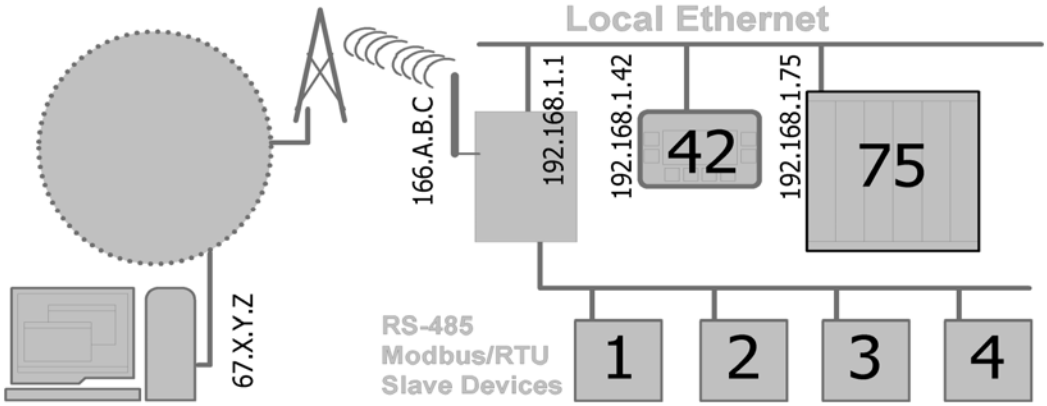
The Digi Connect WAN IA acts much like your home DSL/Cable router. It is assigned an IP address by your service provider (your "cellular ISP"). Outgoing TCP/IP connections are handled with Network Address Translation (NAT), just as your home DSL/Cable router does. This allows any number of local Ethernet devices to connect out into the Public Internet. However, to the Public Internet, the Digi Connect WAN IA appears as just a single IP address.

Therefore, the Modbus Bridge functionality in the Digi Connect WAN IA works much like the Digi One IAP and other Ethernet-to-Serial Modbus products. It manages the TCP and UDP port 502. Incoming Modbus requests are received, time-stamped, and queued within the Digi Connect WAN IA. The Modbus Bridge uses the Modbus/TCP unit Id (or serial slave address) to select which Modbus slave should process the request.



4.2 Modbus Example

Here is an example of a remote Modbus site with two Ethernet Masters (which are also slaves) and four multi-dropped serial slaves. The bold numbers (1-4, 42, 75) in the drawing are the assigned Modbus unit ID or slave address.



4.2.1 IP Addresses

In this Modbus example, all devices on the local 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. The default gateway of 192.168.1.1 is the Digi Connect WAN IA's IP address. The external public IP addresses shown such as 67.X.Y.Z and 166.A.B.C are defined by your corporate IT department and/or your communication provider. Many large users contract an "IP Cloud" from a service provider which includes a mix of technologies all assigned private IP addresses.

Since the Digi Connect WAN IA is also the local IP Router, any of the local Ethernet devices can connect out to remote IP addresses and the Digi Connect WAN IA uses normal Network-Address-Translation (NAT). Remote IP devices can connect, in a more limited way, into local Ethernet devices by configuring the Digi Connect WAN IA to forward TCP and UDP ports other than 502, which is assigned to the Modbus Bridge.

Document 90000774 "*Remote Cellular TCP/IP Access to Industrial Ethernet and Serial Devices*" goes into more details on how IP addressing is handled.

4.2.2 Local Ethernet Masters

The two local Modbus/TCP masters see the Digi Connect WAN IA as a Digi One IAP or Modicon CEV with four serial slaves attached. The Digi Connect WAN IA bridges the Modbus/TCP to either Modbus/RTU or Modbus/ASCII protocols. The two local Modbus/TCP masters can freely communicate between each other.

4.2.3 Remote IP Masters

Remote Modbus/TCP masters see the Digi Connect WAN IA as a Digi One IAP or Modicon CEV with six serial slaves attached. The Digi Connect WAN IA bridges the Modbus/TCP (or Modbus/UDP) to the appropriate local protocol based on Unit Id. Slaves 1 to 4 are accessed by serial Modbus, while slaves 42 and 75 are accessed by Modbus/TCP over the local Ethernet.

4.3 Use of UDP/IP

Although traditionally Modbus/TCP is based on TCP/IP, the Digi Connect WAN also supports Modbus/TCP transported with UDP/IP.

Real-world tests of Modbus/TCP in UDP (or as Modbus/UDP) over cellular networks show that monthly communications costs can be reduced by 60 to 95%. Since you pay for all TCP, UDP, and IP header bytes, UDP/IP gives you an immediate small saving for every packet sent.

UDP/IP over cellular is also surprisingly reliable with measured packet loss at less than 0.1%. However, there is loss, so UDP/IP is only appropriate for applications that include loss detection and recovery.

The largest savings by using UDP/IP come from eliminating the connection overhead and maintenance for TCP/IP. With TCP/IP, you must pay for TCP socket opens and closes, keepalives, retries, header options, and acknowledgments. For example, many OPC servers, when configured to poll a remote Modbus/TCP slave slower than once per minute, will open and close the socket between all polls. This can easily result in the majority of your monthly data bill being related to the TCP/IP protocol, and only a small minority being related to actual Modbus data movement.

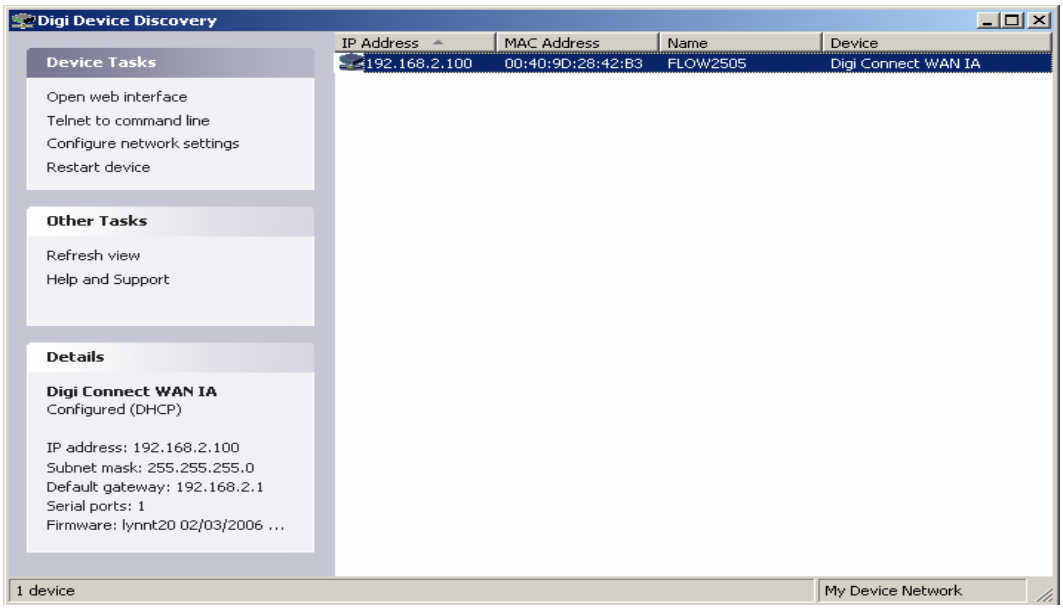
5 Configuring the Digi Connect WAN IA

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

5.1 Device Discovery and IP settings

Install the Digi Device Discovery tool that is included on the CD with your Digi Connect WAN IA 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, for example, Personal Firewall" products, and some combinations of Ethernet hardware under Windows and "cross-cables," which will block or not allow proper device discovery. If you cannot see your Digi Connect WAN IA in 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.



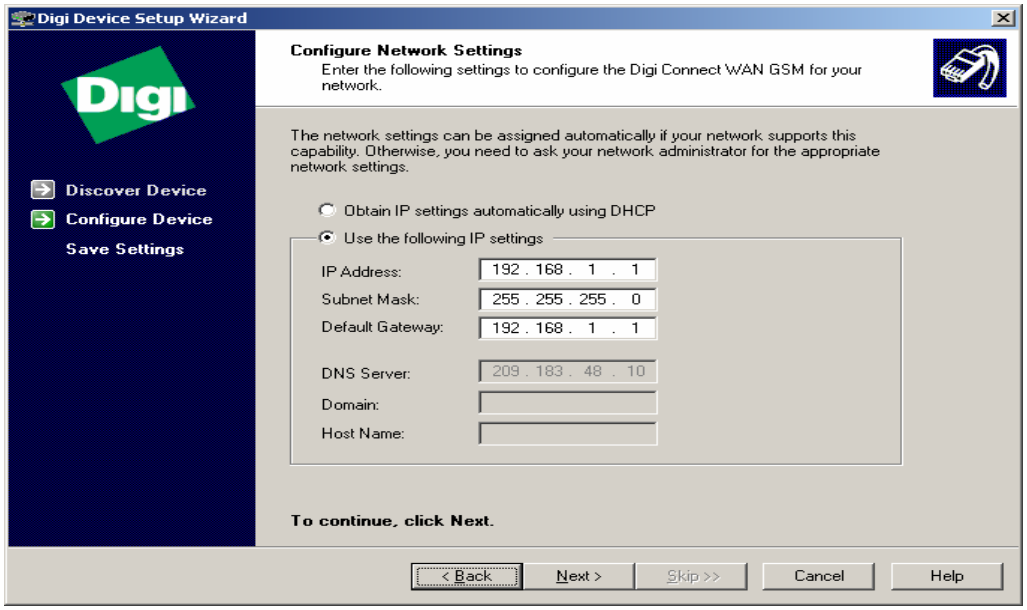
If you see a "<misconfigured>" warning in the device-discovery results, this is caused by the Digi Connect device having an IP address assigned on a different subnet. For example, your PC may have the IP address 192.168.1.201 and the Digi device an IP address of 192.168.20.1. This does not prevent using the **Configure Network Settings** task to correct the IP address information.

The **Name** field shown is the standard hostname, and can be set through the Web user interface under Network Configuration – Advanced Network Settings.

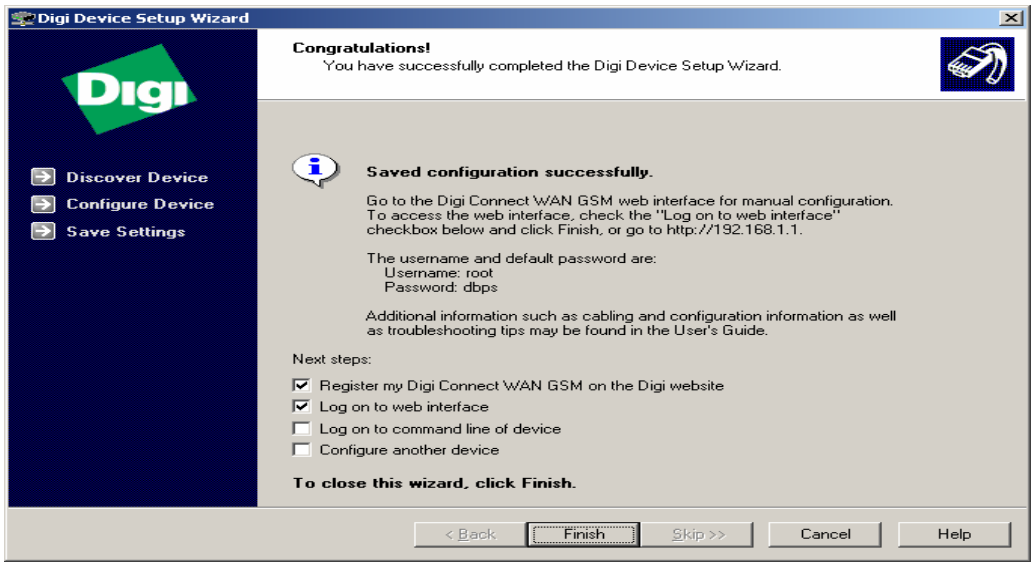
Remember that at this point, we are just assigning the IP address used by the *Ethernet port* of the Digi Connect WAN IA. The IP address used by the *cellular port* will be assigned by your cellular service provider. Even if you have arranged for a fixed IP address to be used, the ISP will "dynamically" 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 the IP address 192.168.1.1 as both the IP address and default gateway.

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

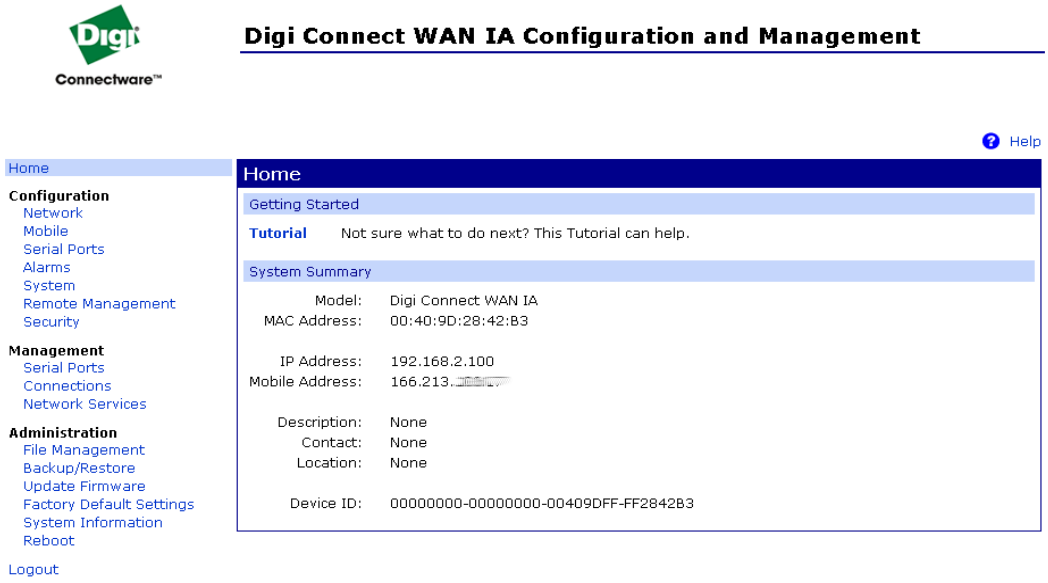


After a minute or two, you should see the **Congratulations** screen below.

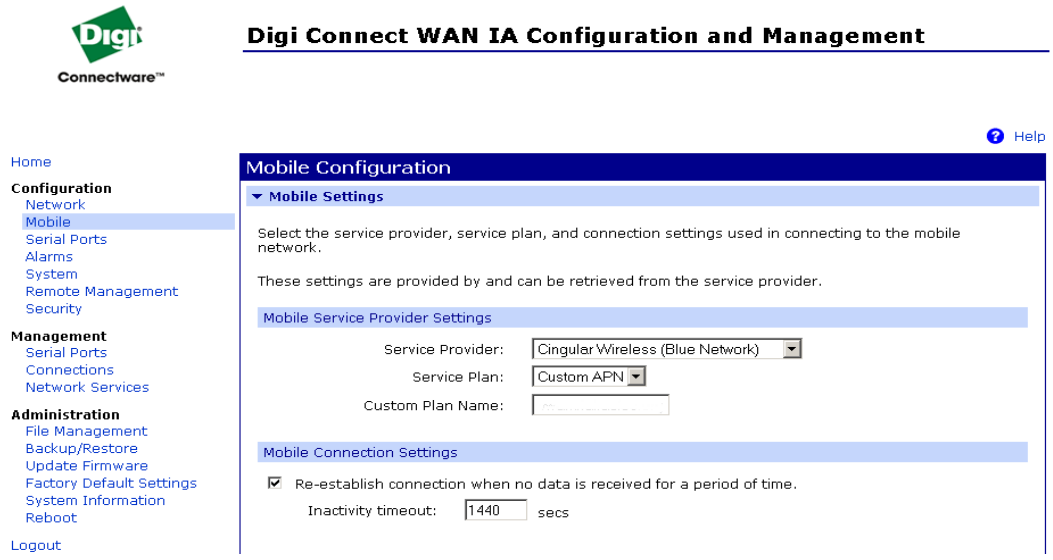


5.2 Web Interface and Service Plan Settings

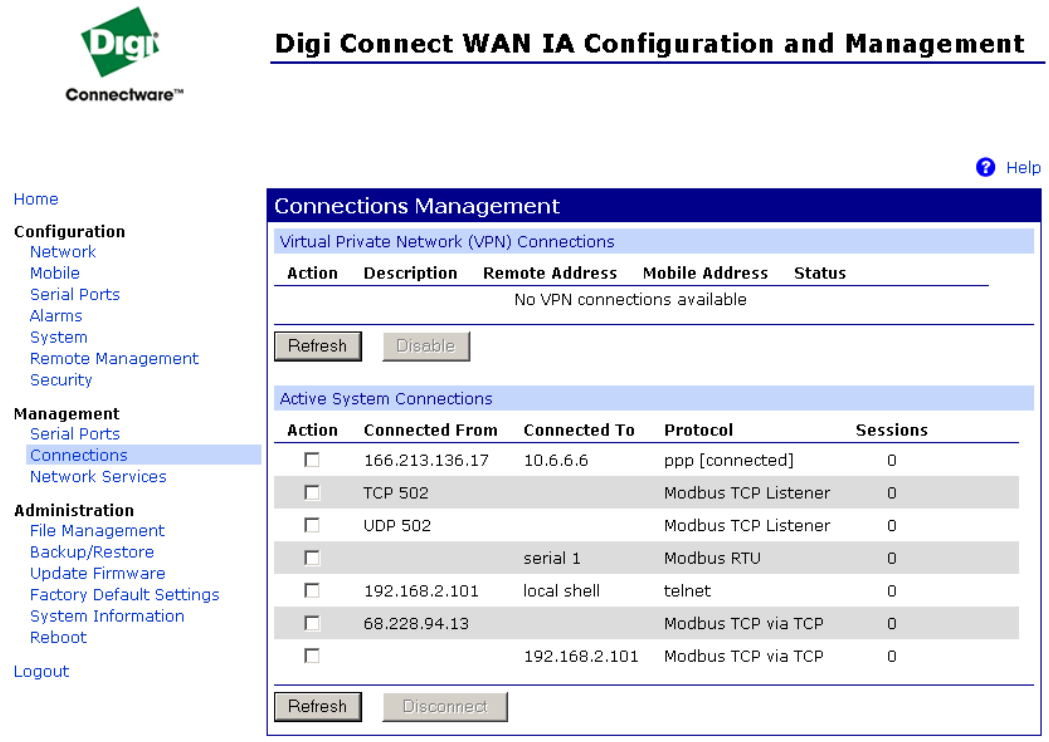
Next, open the Web user interface for you newly installed Digi Connect WAN IA. You can either open the Web user interface from the last screen of the Digi Device Setup Wizard, as shown above, or launch your desired web browser, specifying the address of the Digi Connect WAN IA. The home page for the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



At present, your Digi Connect WAN IA will not be 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. **[connected]** 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, the roaming partner to which you are connected to may not support the data services required for IP traffic.

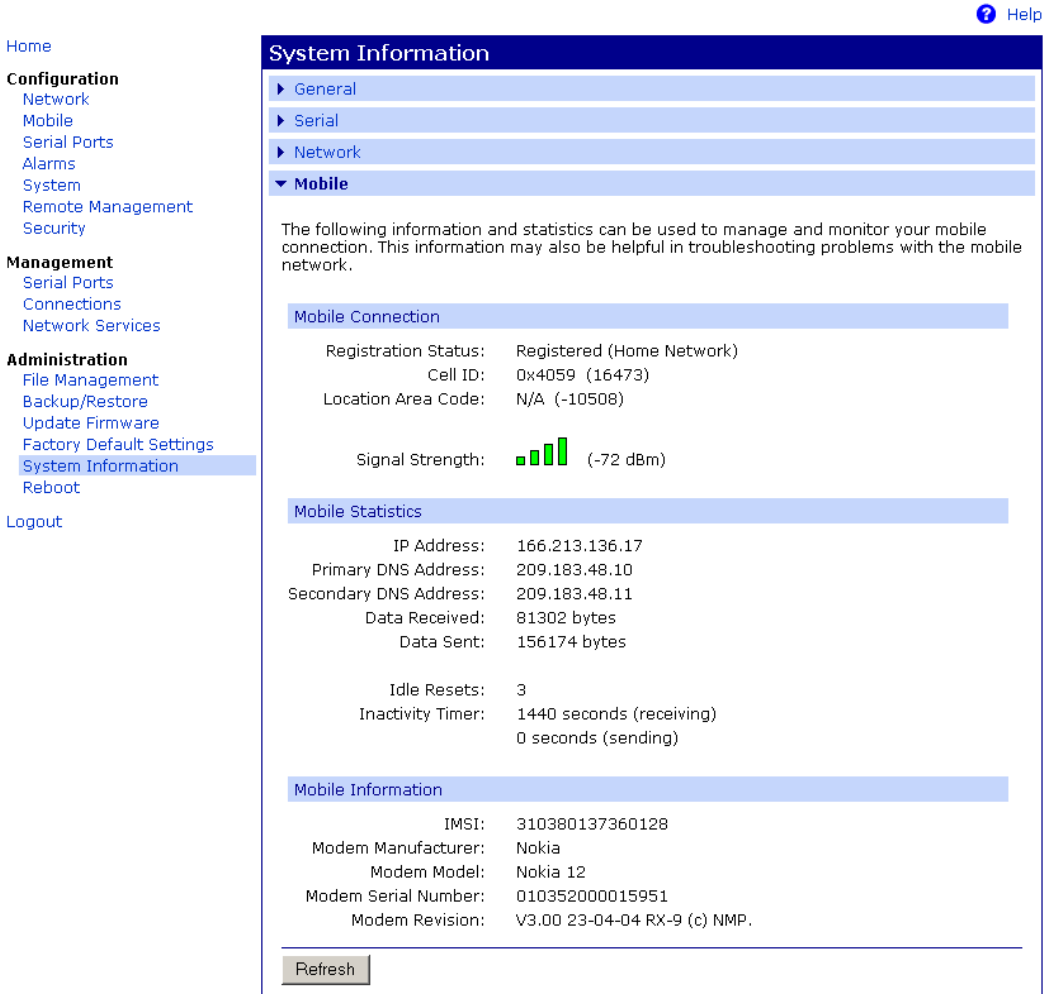


The next three lines in the list (the two Modbus TCP Listeners and Modbus RTU) define the standard tasks running in the default configuration. The Digi Connect WAN IA accepts incoming Modbus/TCP via TCP port 502 or UDP port 502 (that is, the two "listeners"), and also has a task managing the serial slaves on port 1.

The "telnet" connection in the connection list shows someone on the PC at address 192.168.2.101 has a Telnet session open to the Digi Connect WAN IA. Normally, this connection would not exist. In this case, the "someone" (that is, the author of this document) has the Telnet session open to monitor the Digi Connect WAN IA directly by the command line.

The final two connections in the connection list show an actual remote Modbus/TCP Master connected through the Digi Connect WAN IA to a local Modbus/TCP slave. The "Connected From" IP is the firewall connection of a remote corporate network. The "Connected to" IP happens to be a Modbus slave simulator running on the same PC as the Telnet.

Another useful status display is under **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 Connect WAN IA and the DNS address or addresses for your field devices to use.

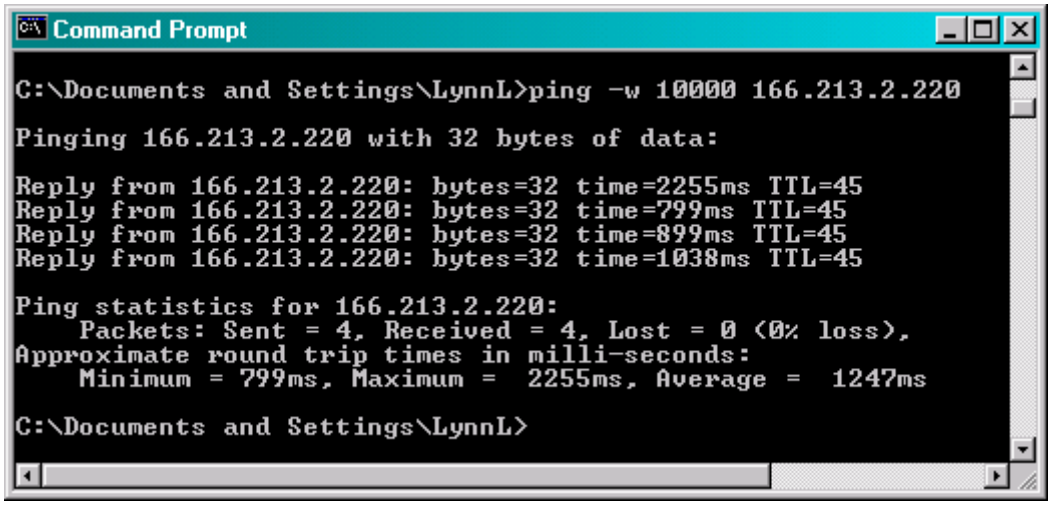


5.3 Out-Going is Active

As configured, your local field devices can initiate outgoing connections to the Internet or central servers you maintain. They use the Digi Connect WAN IA (IP address 192.168.1.1) as the router to forward the connection; the Connect WAN IA uses Network-Address-Translation (NAT) to access the remote resource. For example, if a local PLC connected through the Digi Connect WAN IA to a server at IP address 67.43.210.56, the server would see a connection from the Digi Connect WAN IA device's IP address (for example 166.213.X.Y) and NOT from the IP address of your field device.

5.4 Using the "ping" Command

Just like all IP devices, you can use the "ping" command to test access. However, many "ping" utilities assume a short 1- 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.



6 Digi Connect WAN IA Default Configuration

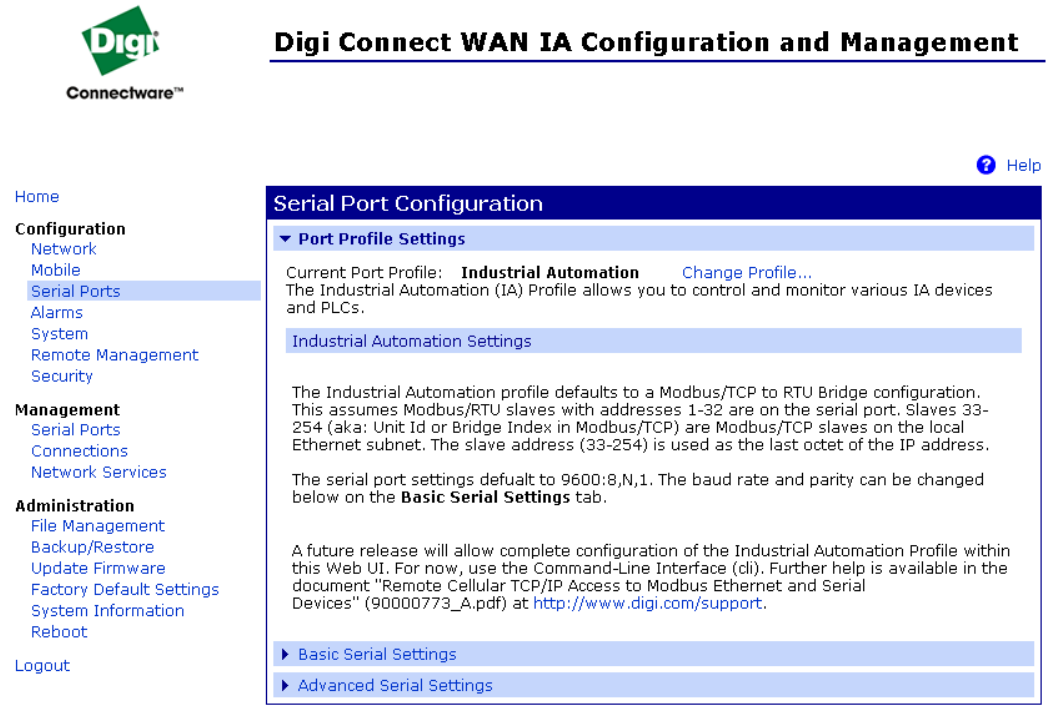
Following is a discussion of the factory default settings for the Digi Connect WAN IA, and how to modify settings as needed.

6.1 Default Port Profile

The Digi Connect WAN IA defaults to a port profile of Industrial Automation. In the Web user interface, Port profiles are set and changed under **Serial Ports** > **Port Profile Settings**.

To disable the Industrial Automation function (that is, the Modbus Bridge), click **Change Profile**, select any port profile other than IA and reboot.

To re-enable the Industrial Automation function (that is, the Modbus Bridge), reselect the IA profile and reboot; or click **Administration** > **Factory Default Settings**.



6.2 Default Serial Port Settings

The Digi Connect WAN IA defaults to assume Modbus/RTU slaves with addresses 1 to 32 are attached to the serial port. Default port characteristics are 9600:8,N,1. Unit Id zero (0) is auto-mapped to Modbus/RTU slave address 1. The electrical interface is set as EIA-232, EIA-422, or EIA-485 by the four DIP switches on the bottom of the unit.

In the Web user interface, use the **Basic Serial Settings** to change the baud rate, parity, and so on.

Use the **Advanced Serial Settings** to enable or adjust the RTS/CTS behavior for half-duplex or radio modem support. The "Pre-Delay" setting defines how long the Digi Connect WAN IA pauses after RTS is asserted before sending data. The "Post-Delay" setting defines how long the Digi Connect WAN IA pauses after the last byte of data is sent before dropping RTS.

IA configuration is also performed by the "set ia" command from the Telnet command line. The serial port configuration defaults to the following "set ia" command settings. Future Digi Connect WAN IA firmware releases will expose these configuration settings in the Web user interface.

```
set ia serial=1 active=on type=slave protocol=modbusrtu table=1
set ia serial=1 messagetimeout=2500 slavetimeout=1000 chartimeout=20
set ia serial=1 priority=medium idletimeout=0
set ia serial=1 errorresponse=off broadcast=replace fixedaddress=0
set ia table=1 route=1 active=on type=serial protaddr=0-32 port=1
```

To enable a Modbus/RTU serial Master instead, change the configuration settings by entering the following "set ia" commands via telnet. These "set ia" commands change the serial device type from slave to master. In addition, we need a 31-second or longer timeout to handle cellular latency. Finally, we can just turn off the first route, which forwards requests to the serial port.

```
set ia serial=1 type=master messagetimeout=31000
set ia table=1 route=1 active=off
```

6.3 Incoming Modbus network masters

The Digi Connect WAN IA defaults to enable incoming Modbus/TCP masters to connect by either the cellular or local Ethernet port. The incoming Modbus network masters can use either TCP/IP or UDP/IP. The long 30-second "**chartimeout"** is required to properly reassemble Modbus requests, which may arrive in more than one TCP or UDP packets.

```
set ia master=1 active=on type=tcp ipport=502 protocol=modbustcp table=1
set ia master=1 messagetimeout=2500 chartimeout=30000 idletimeout=4500
set ia master=1 priority=medium
set ia master=1 errorresponse=on broadcast=replace fixedaddress=0
set ia master=2 active=on type=udp ipport=502 protocol=modbustcp table=1
set ia master=2 messagetimeout=2500 chartimeout=30000 idletimeout=4500
set ia master=2 priority=medium
set ia master=2 errorresponse=on broadcast=replace fixedaddress=0
```

In addition, the Digi Connect WAN IA Modbus Bridge design includes "implied" masters linked to the serial port configuration. Since the serial port defaults to

Modbus/RTU, this means the Digi Connect WAN IA allows incoming Modbus/RTU masters on TCP and UDP port 2101. Changing the serial port to Modbus/ASCII automatically changes these "implied" masters to encapsulate Modbus/ASCII instead of RTU. To change the port number or enable mixing the serial protocols, you can explicitly *add* these masters to your configuration. For example, the following "set ia" commands enable incoming Modbus/ASCII masters on both TCP and UDP ports 7001. For serial protocols, you generally want "**errorresponse=off**," since few serial Masters expect to see Modbus exception responses 0x0A or 0x0B (bridge errors).

```
set ia master=3 active=on type=tcp ipport=7001 protocol=modbusascii table=1
set ia master=3 messagetimeout=2500 chartimeout=30000 idletimeout=4500
set ia master=3 priority=medium
set ia master=3 errorresponse=off broadcast=replace fixedaddress=0
set ia master=4 active=on type=udp ipport=7001 protocol=modbusascii table=1
set ia master=4 messagetimeout=2500 chartimeout=30000 idletimeout=4500
set ia master=4 priority=medium
set ia master=4 errorresponse=off broadcast=replace fixedaddress=0
```

6.4 Modbus Request Forwarding and Proxy

The Modbus Bridge function acts much like a Proxy slave. As each Modbus request arrived, it is time-stamped and queued. The "**messagetimeout**" option in the **serial** and **master** configuration defines how long a request can sit in the queue before it is discarded or returns an exception 0x0B error.

Once in queue, the Modbus slave address (also called the Unit Id or Bridge Index) is used to select the appropriate slave destination. The first default route sends all requests to slaves 0 to 32 to the serial port. The protocol behavior is defined by the "set ia serial" command previously discussed.

```
set ia table=1 route=1 active=on type=serial protaddr=0-32 port=1
```

The second default route proxies Modbus requests on the local Ethernet. The slave address is used as the last octet of the local IP address. For example, if the Digi Connect WAN IA has the address 192.168.20.1, then slave 75 will be the Modbus/TCP server at 192.168.20.75, and so on. The "**fixedaddress=1**" option means the Unit Id received by the slaves will always be 1. The "**idletimeout=120**" option means that TCP sockets to local slaves will automatically close if no new messages arrive within 2 minutes.

Besides overcoming the limitation that the Digi Connect WAN IA appears to the outside world as a single IP address, this "proxy" function is also useful, since many small Modbus/TCP slaves have wide-area-network-unfriendly TCP stacks. These TCP stacks may rapidly force incoming TCP sockets closed when idle or expect TCP acknowledgement packets faster than is reasonable over a wide-area-network. This "proxy" function also allows the use of Modbus/TCP in UDP packets over the cellular link, which can cut your cellular data costs by from 60% to 95%.

```
set ia table=1 route=2 active=on type=ip protaddr=33-255
set ia table=1 route=2 protocol=modbustcp transport=tcp connect=passive
set ia table=1 route=2 ipaddress=0.0.0.0 ipport=502 replaceip=on
```

```
set ia table=1 route=2 slavetimeout=1000 chartimeout=50 idletimeout=120
set ia table=1 route=2 errorresponse=on broadcast=replace fixedaddress=1
```

To enable local Masters to access remote IP-based slaves, add routes, as shown in the following example. The first "set ia" command changes the protocol addresses for the existing default route to *not* include the address 33. Then we create a new route for slave #33, which targets a remote IP address. The long 30-second slave and character timeout are required for the high-latency cellular link. While you may routinely receive responses in only 1-2 seconds, in this case, we want to back off and be patient when the network is slow – remember that you are paying for all data moved, which includes retries and duplicate requests due to impatience.

```
set ia table=1 route=2 protaddr=34-254
set ia table=1 addroute=3
set ia table=1 route=3 active=on type=ip protaddr=33
set ia table=1 route=3 protocol=modbustcp transport=tcp connect=passive
set ia table=1 route=3 ipaddress=67.0.0.1 ipport=502 replaceip=off
set ia table=1 route=3 slavetimeout=30000 chartimeout=30000 idletimeout=300
set ia table=1 route=3 errorresponse=on broadcast=replace fixedaddress=1
```