How to Architect Your Systems to Get the Most Out of Your Modbus Devices

Edward Lin
Product Manager

MOXA®

Introduction

Modbus is the lifeblood of a large number of automation applications. This de facto industrial communications protocol, following a master-slave architecture, has two major formats: Modbus RTU and Modbus TCP, a modified version of the former. In essence, both formats fulfil the same functions. The main difference between them pertains to the physical layers they run on: Modbus RTU is used in serial communications (RS-232 or RS-485) while Modbus TCP facilitates communications over TCP/IP networks. Furthermore, Modbus TCP data packets use an additional header for TCP connections.

Many field devices support Modbus RTU protocol because it is easy to implement and allows troubleshooting at a very low cost—which explains why Modbus RTU has become so popular. As more and more industrial applications continue to migrate to Ethernet to fulfil more complex networking needs, many systems now use Modbus TCP in SCADA (supervisory control and data acquisition) software. An unfortunate outcome is communication issues between Modbus RTU and Modbus TCP protocols.

As expected, with any transition come new opportunities. And, so do challenges—four major ones to speak of when linking up Modbus-supported devices to SCADA systems. The first challenge involves dealing with the oversupply of proprietary SCADA software on the market. As each software program brings different supporting abilities for Modbus drivers to the market, things become complicated for system operators to match the right product with their requirements. Another headache for system operators is coping with mounting pressure to manage a large number of serial devices over a TCP-connected architecture. Complicating things further is when multiple SCADA hosts need to access the same Modbus RTU-supported device. Then, last but not least, ensuring a faster response time for mission-critical applications has proven to be quite the hurdle to scale in order to achieve optimum operations. In this article, we will outline these aforementioned challenges and offer solutions to bring your operations up to speed in an ever-changing technological world.

Challenge 1—A Gateway To Performance

SCADA software providers abound in the marketplace; each of them offers different capabilities to support Modbus drivers. Pinpointing your system's exact requirements to a tee is crucial to select the most suitable software. You don't want to fumble around afterwards with interoperability issues once you have connected your Modbus RTU-supported serial devices to an Ethernet network:

Released on February 26, 2016

Moxa is a leading manufacturer of industrial networking, computing, and automation solutions. With over 25 years of industry experience, Moxa has connected more than 30 million devices worldwide and has a distribution and service network that reaches customers in more than 70 countries. Moxa delivers lasting business value by empowering industry with reliable networks and sincere service for automation systems. Information about Moxa's solutions is available at www.moxa.com.

**How to contact Moxa**
Tel: 1-714-528-6777
Fax: 1-714-528-6778

MOXA®
Reliable Networks ▲ Sincere Service

(1)　A SCADA host with a Modbus TCP driver

A protocol conversion gateway can be used here. With a gateway's help, you can use Modbus TCP protocol to communicate with Modbus RTU-supported devices. Many gateways that enable Modbus TCP connections for Modbus RTU slave devices are available in the automation market. When the gateway receives the Modbus TCP request, it converts the packet to a Modbus RTU packet and sends it to the Modbus RTU-supported devices immediately.
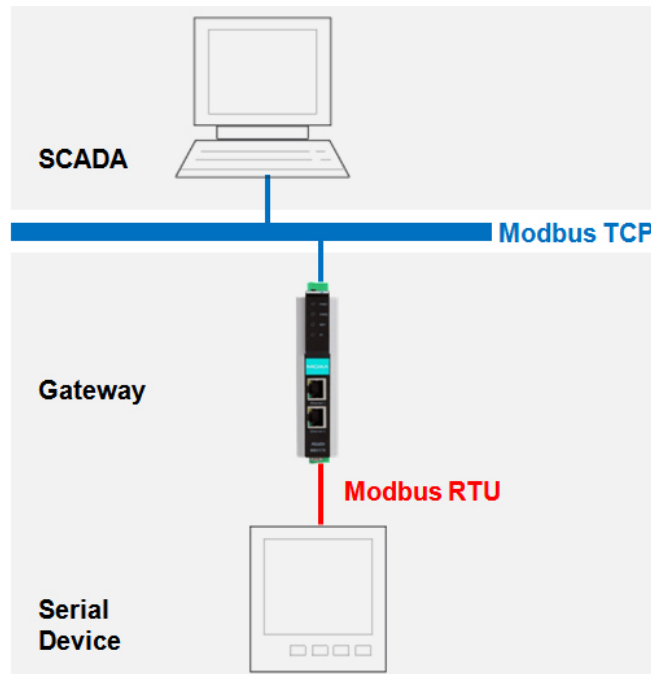

(2) A SCADA host with a Modbus RTU driver, with built-in serial port

For some reason, an existing SCADA host with built-in serial ports and Modbus RTU-supported devices can't be changed, but both ends need to be bridged by an Ethernet network. In this instance, you can use a pair of gateways to enable connectivity. As shown in the topology illustration below, the gateway can convert the Modbus RTU packet into a Modbus TCP packet and back to a Modbus RTU packet.
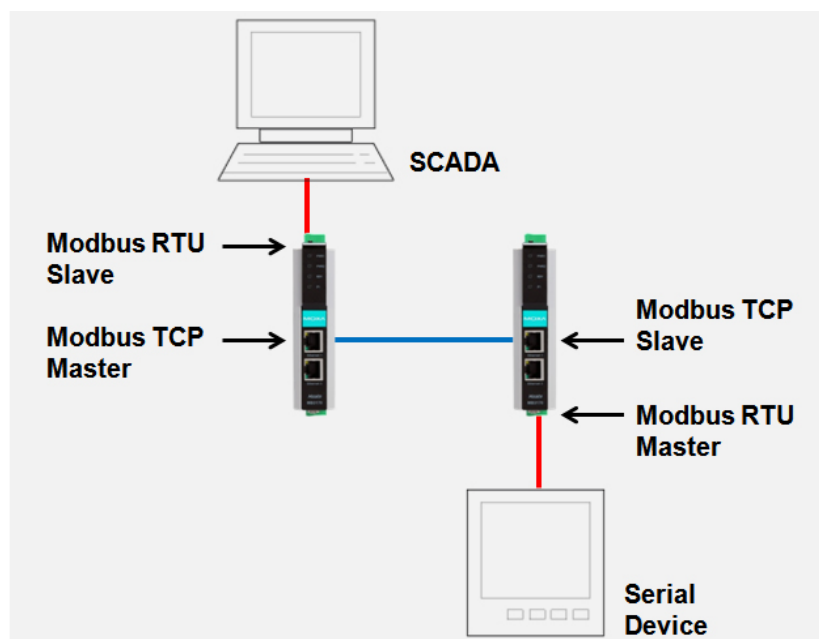
(3) A SCADA host with a Modbus RTU driver, without a built-in serial port

If you want to use your existing SCADA program and devices, but your original SCADA host does not have a built-in serial port, a serial device server can be used to build a virtual COM port for the serial port on the remote serial device server connected to your serial devices. This configuration allows you to access remote serial devices through the serial device server as if it had a native COM port. The serial device server will install the virtual COM port driver on your SCADA host to create a virtual COM port. To enable the virtual COM port, the serial device server must be configured to virtual COM mode. The data sent to this virtual COM port will be transferred to the remote serial port of the serial device server. Actions for the modem signal will also be handled in the same way. Since you can use the virtual COM port in the same way as a native local COM port, you can send the Modbus RTU request to the COM port directly, just as you would if it were a physical COM port.
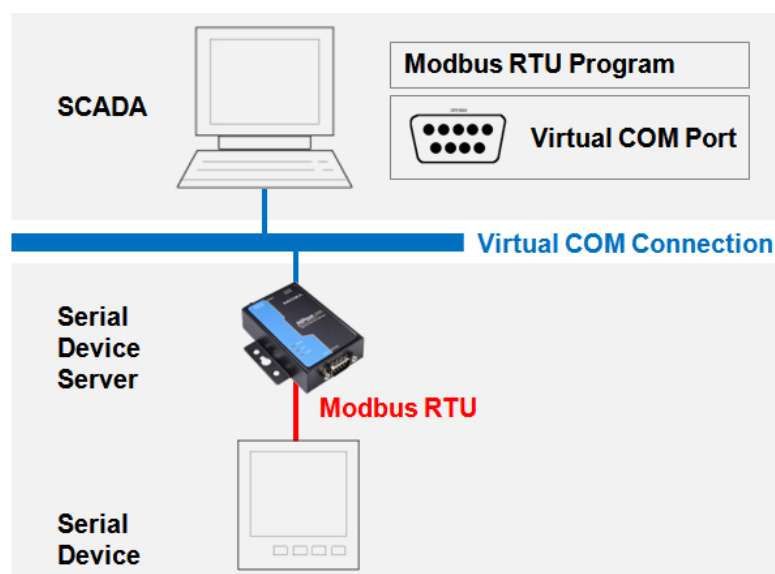
## (4) A SCADA host with an Ethernet Encapsulation driver

If you don't have a built-in serial port on your SCADA host, and you don't want to install a virtual COM port driver, you can consider an Ethernet encapsulation driver instead. However, your SCADA software must be able to support the Ethernet encapsulation driver, which isn't always the case. Generally speaking, Ethernet encapsulation drivers are the best option if you have an in-depth knowledge of serial and TCP/IP networks. In this case, you will need a serial device server with raw socket mode or tunneling mode selected, which means the connection between the host and serial device server uses transparent TCP/IP or UDP communication without any protocol involved when SCADA sends Modbus RTU packets to field devices. The serial device server needs to be configured properly because Modbus RTU uses interval timeouts to determine the end of a packet. If the host is not configured properly, the Modbus RTU packet may be divided into two or more TCP/IP or UDP packets, and the host will drop the incomplete packets.
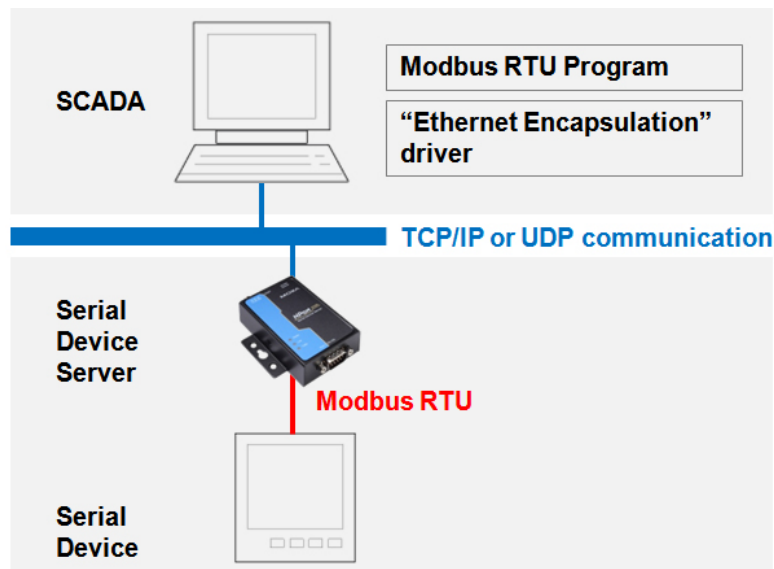
Although serial device servers can also connect Modbus RTU devices to an Ethernet network, a gateway solution can meet almost every system requirement. Your host must be able to support Modbus TCP connections. This shouldn't be a problem since Modbus TCP is very popular and already widely supported. Here are a few situations where you will need to use a designated gateway solution:

(1) Multimaster or redundancy

In addition to enabling remote access, Ethernet connections also provide multiple connection access capabilities. Most gateways can support up to 32 connections, which means up to 32 SCADA hosts can query the Modbus RTU-supported devices simultaneously. Although it will be difficult for a serial device server to provide network redundancy in this situation, as most serial device servers do not support multiple masters, a gateway, on the other hand, will have no problem.
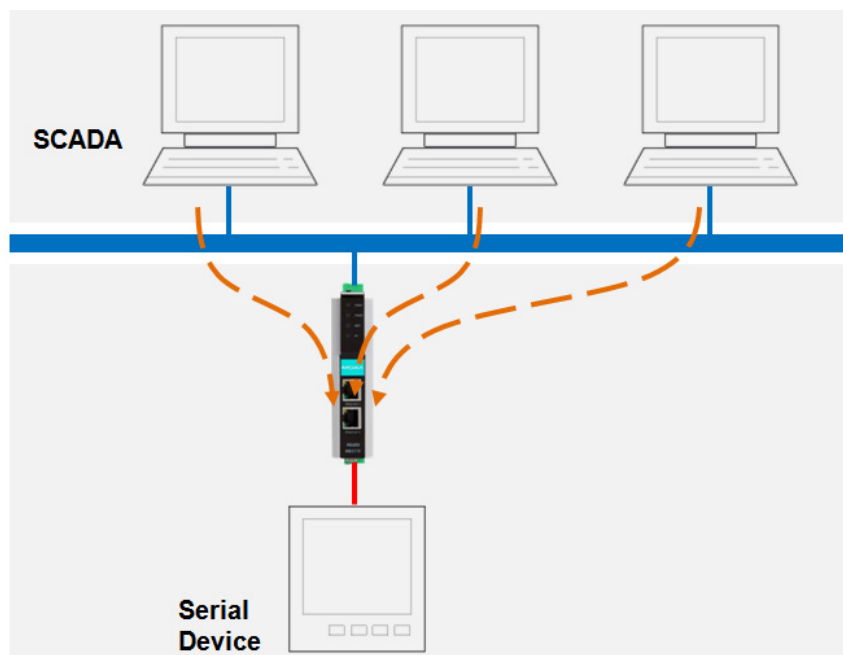
(2) A single connection for multiple Modbus RTU-supported devices

Sometimes, you may want to use a single connection on a SCADA host to query multiple Modbus RTU-supported devices connected to different serial ports. In this case, a gateway is the only solution to process the routing mechanism. Multiple serial port gateways can be used and configured to send the Modbus request to the correct serial port based on unique slave IDs. A serial device server can't handle such a complicated request.
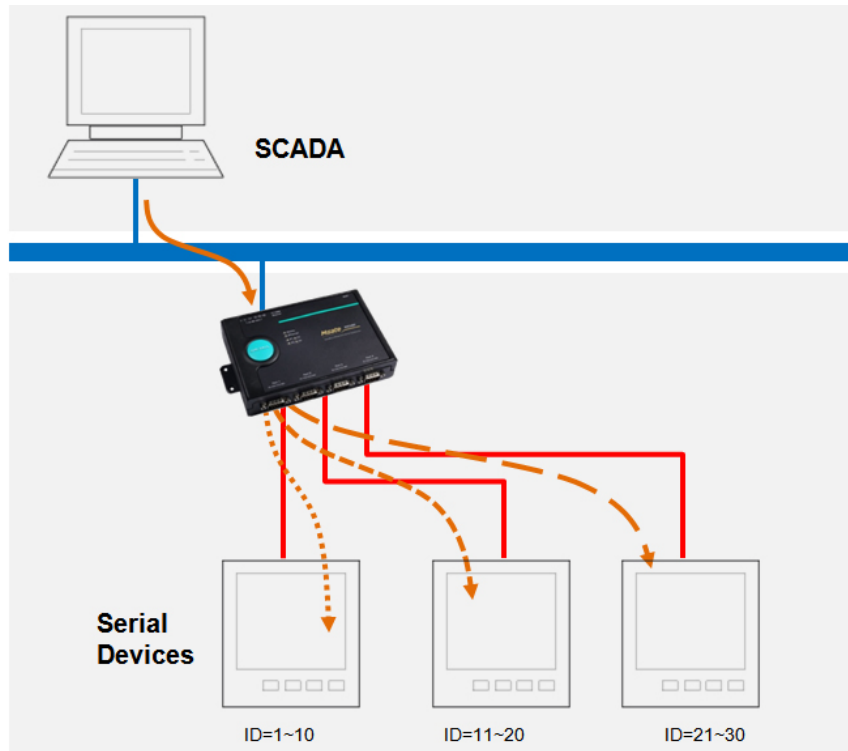
(3) Simultaneous device access for old Modbus RTU HMI and new Modbus TCP SCADA

Although Ethernet connections offer easy-to-deploy remote access, there may be times you want to keep your existing local HMI connections. The problem is the serial port on the device is already connected to a gateway, so there is no available serial port for the HMI connection. In this situation, some gateways provide a serial port redirector to overcome this obstacle. The serial port redirector is very similar to a router in that the gateway can transfer the request between different serial ports based on the slave ID.
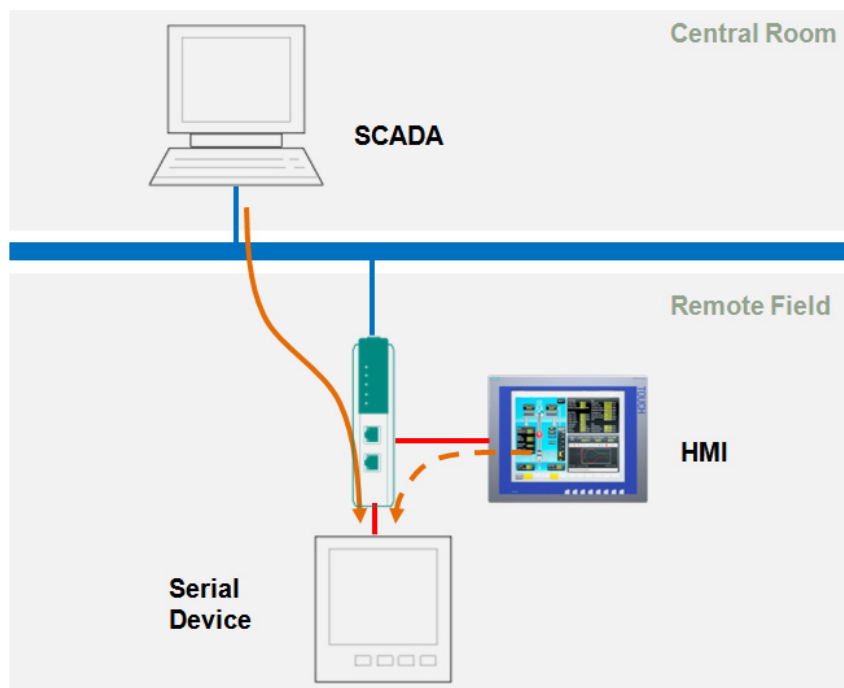
The Best Option

Many different solutions for serial-to-Ethernet communications are available. Although these solutions can be as simple as a transparent data transfer between serial and Ethernet ports, a designated gateway is generally a far better option if you need to handle industrial protocols such as Modbus. A designated gateway may involve a larger initial investment, but it provides more stable communications in the long run, and it can recognize Modbus packets to properly handle entire packets.

Challenge 2—Too Much To Handle

It is a case of the more things you have, the more you have to cope with. And, it is exactly the case when a large number of serial devices need to be connected to an Ethernet network. Just ask any network engineer when faced with the planning of the architecture of a TCP connection. The good news is that most gateways provide flexible solutions to plan efficient TCP connection architectures for multiple Modbus RTU-supported devices connected to the different serial ports of a gateway. Usually, three different routing mechanisms apply:

(1) Link a serial port to a unique TCP port

This is the most popular setup for a gateway topology. In this gateway configuration, each serial port will be identified by one unique TCP port. For example, 4001 for serial port 1, 4002 for serial port 2, etc. If you want to communicate with Modbus RTU-supported devices on serial port 1, you need to build a Modbus TCP connection to 4001. The gateway will transfer the Modbus packet between TCP connection on TCP port 4001 and serial port 1. In this topology, the SCADA driver needs to build multiple Modbus TCP connections.
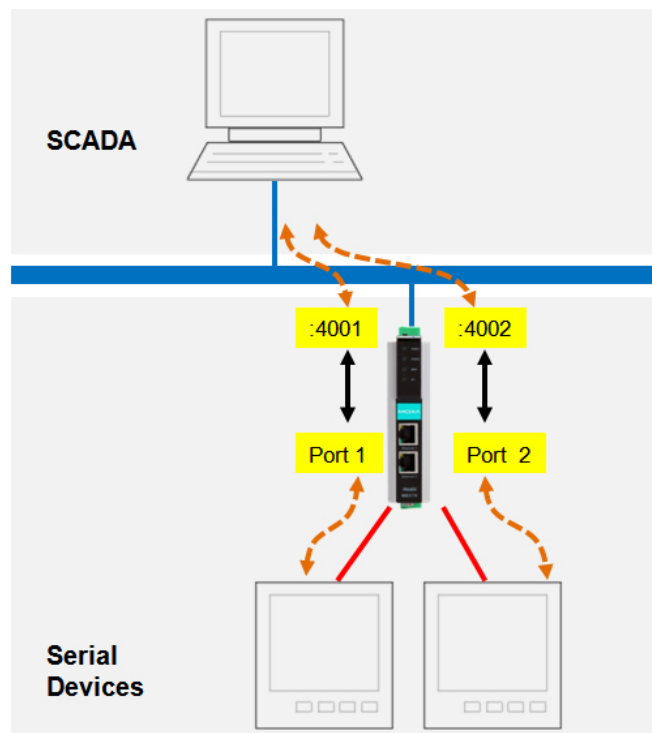
(2) Link a serial port to a unique IP address

This is very similar to the previous mechanism, except the gateway uses different IP addresses to identify different serial ports, for example, 192.168.2.1 for serial port 1; 192.168.2.2 for serial port 2, etc. If you want to communicate with Modbus RTU-supported devices on serial port 1, you need to build a Modbus TCP connection to 192.168.2.1 with TCP port 502. The gateway will transfer the Modbus packet by TCP connection between 192.168.2.1:502 and serial port 1. In this topology, the SCADA driver also needs to build multiple Modbus TCP connections.
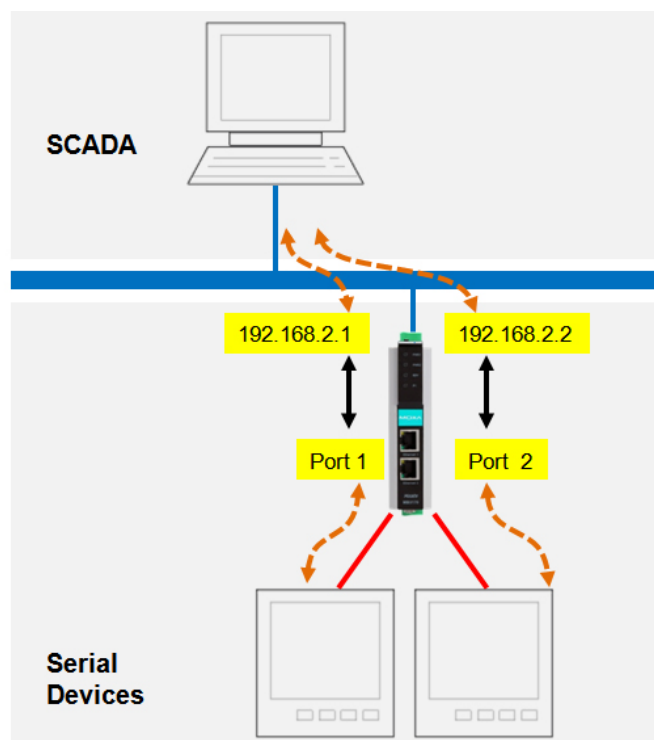
(3) Use a routing table

This topology uses a single connection to communicate with multiple devices. To make sure the request can be transferred to the correct serial port, both the gateway and the routing path need to be configured properly. For example, serial port 1 handles all Modbus packets that have slave IDs 1 to 10, serial port 2 handles slave IDs 11 to 20, etc. The drawback of this mechanism is that this topology only uses one connection; thus, it will be slower than the other two. But, if you have budget constraints, which subsequently will limit the number of SCADA connections, and the performance is still acceptable for your needs, then a single connection can be a good option.

Note:

If you connect more devices to a single serial port, or if you link multiple serial ports to one TCP connection, you will increase the Modbus polling time. But again, if you want to increase the polling speed, more TCP connections might be required, and you need to confirm whether this will increase your SCADA costs.
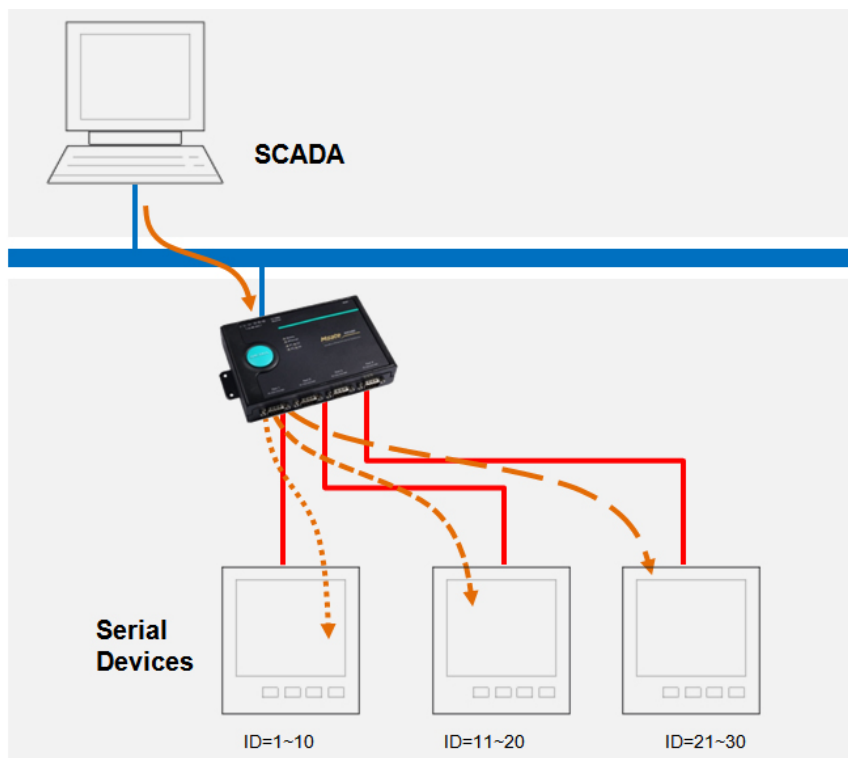

Challenge 3—Unlimited Access

In projects with a big scope, several hosts could monitor a territory, as well as act as a redundant host to monitor devices in other territories when required. Therefore, Modbus RTU-supported devices need to be connected to multiple masters to allow simultaneous access.

Although a gateway can handle this adequately, you need to remember that the serial port bandwidth stays the same. So, if multiple requests are received through the same serial port, you will likely experience a delay because the gateway needs to handle the earlier requests in the queue first. This means you need to calculate the proper polling time if you want to enable multiple masters to access the same Modbus RTU device simultaneously.

For example, if one request takes 100 ms, five multiple connections would require you to wait at least 100 ms $\times (5 - 1) = 400$ ms before sending the next request. This means each SCADA host will need to use 400 ms (plus some tolerance) for polling cycle time.
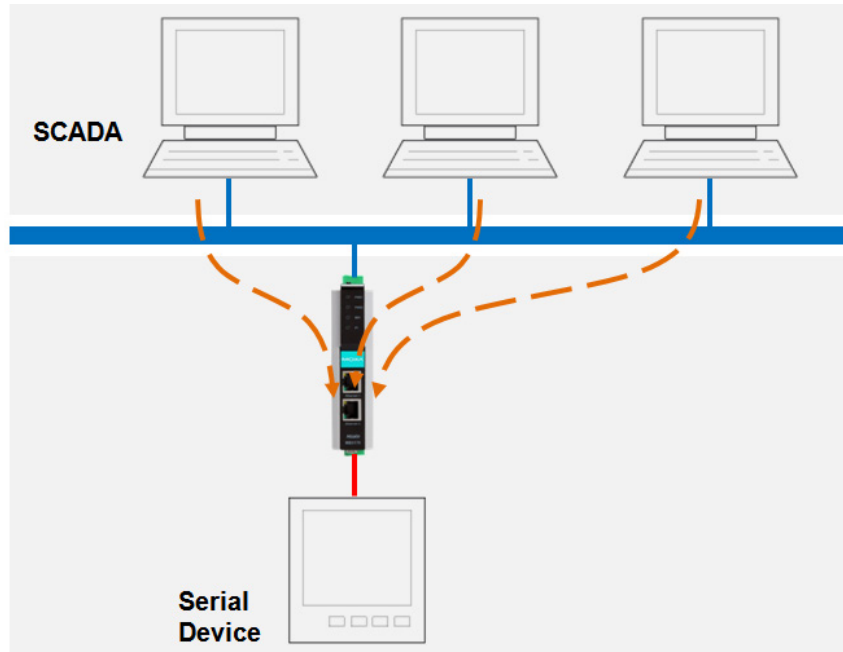

But, the above offering is not the only solution. Some gateways support an agent mode, which actively and continuously retrieves data from connected devices. The updated data is stored in internal memory, which is used to reply to host requests. Although this solution is faster and more efficient, the data retrieved might not be the most current in real time.
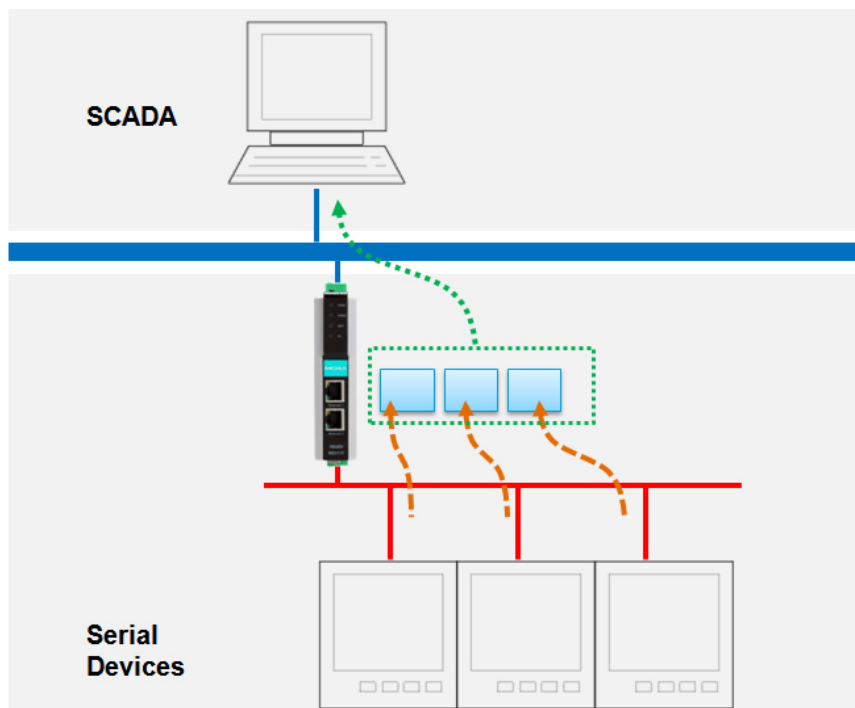
Challenge 4—The Need for Speed

For some critical applications, it is vital to have fast responses from devices. However, speed is lacking when multiple Modbus RTU-supported devices are connected due to Modbus behavior (one polling with one response) and the speed limitation of serial transmissions. In these cases, an agent gateway is needed. An agent gateway can send a Modbus command to each Modbus RTU-supported device one by one, so it can actively retrieve data from multiple Modbus RTU-supported devices and place them into a single register, storing the data in its internal memory. Thus, to get a faster response, the host just needs to retrieve data through an Ethernet connection. Besides, all the data can be arranged into a single data block in the gateway's internal memory to retrieve data from a host with just one Modbus read command.
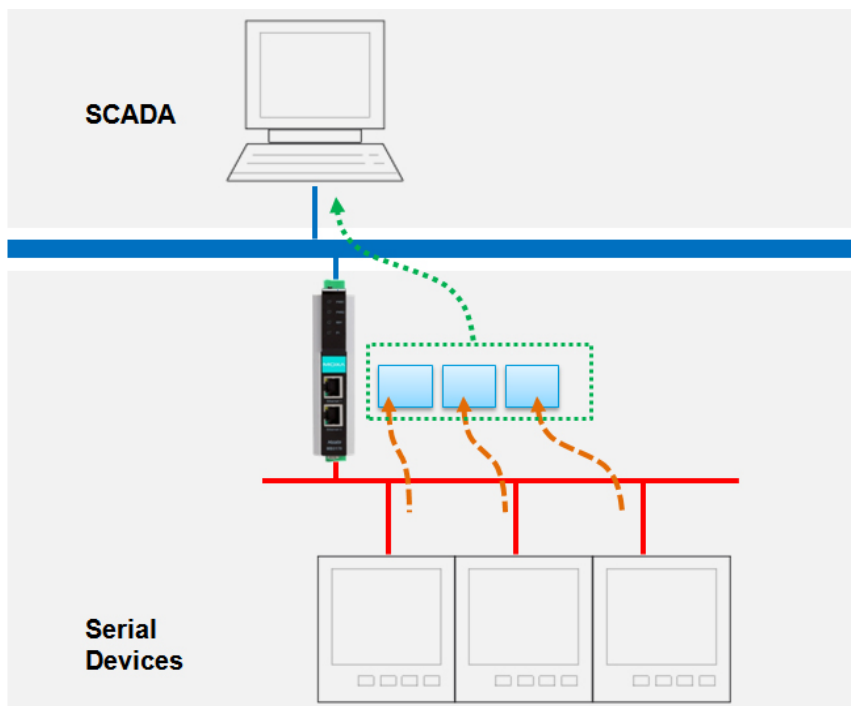
About Moxa's Solutions

Moxa is the industry leader in serial-to-Ethernet connectivity. NPort® serial device servers provide plenty of operation modes, such as Real COM Mode, TCP Client Mode, and UDP Mode, to connect types of serial devices to a network. What's more, with each operation mode, the NPort® is also equipped with various advanced functions to assist users in streamlining operations and maximizing the benefits of serial-to-Ethernet connectivity. Moxa's MGate™ gateways enable protocol conversions between SCADA/PLC and devices with different protocols. Standout features that set MGate™ gateways apart from others in the industry include easy configuration with a user-friendly web console, easy maintenance with built-in monitoring and diagnostics, and reliable performance. To learn more about our solutions, visit http://www.moxa.com/support/request_catalog_detail.aspx?id=1424

# Glossary

Serial Device Server

A serial device server is a standalone device that has at least one Ethernet port and one or more serial ports. Serial device servers are equipped with an embedded network operating system and allow computers to access serial devices over a network. A serial device server can transfer data transparently between a serial interface and an Ethernet interface, converting all of the serial data received for Ethernet protocol communications, as well as converting all Ethernet data received for serial protocol communications. Modern serial device servers can also provide a virtual COM port for host computers that lack an extra serial port, essentially converting an Ethernet port into a COM port. Besides these basic functions, more sophisticated serial device servers can even support PPP in serial or Telnet protocols over Ethernet networks. A serial device server can even be used for console management (which is why some vendors may call it a console server) or remote terminal applications in old banking systems (which is why it also sometimes called a terminal server).

Virtual COM Port, Virtual COM Port Driver

A virtual COM port is not a real (i.e., physical) COM port on the host computer. Instead, a virtual COM port driver program under Windows is installed on the host computer to completely emulate the behavior of a local COM port. The driver is used for ports on a serial device server that connects to the host via a TCP/IP network. The serial port on the remote serial device server will exhibit the same behavior as a local COM port. This virtual COM port is configured to link to a serial device server. For example, suppose COM3 is serial port 1 on a remote serial device server that has an IP address of 192.168.2.1. When you write data to this port, all the data will be sent to "192.168.2.1@serial port 1" by the virtual COM port driver. All the requests to this virtual COM port will be re-directed to "192.168.2.1@serial port 1". Because newer host computers often lack built-in serial ports, virtual COM ports are an invaluable tool for connecting legacy equipment in industrial automation applications.
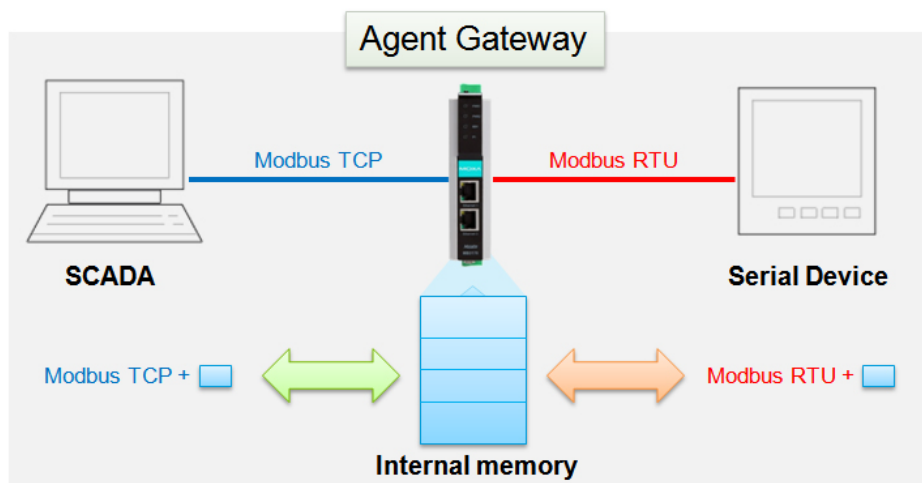
Transparent Gateway

A transparent gateway is the most basic way to use a Modbus gateway. Since Modbus RTU and Modbus TCP have the same PDU (Protocol Data Unit), and the only difference is the address field, it is easy for a gateway to transfer data between Modbus RTU and Modbus TCP. So when the gateway receives the Modbus TCP packet from the Ethernet network, it can simply replace the address field to fit Modbus RTU addressing and immediately send the packet to the serial port. When the gateway receives a reply from the Modbus RTU devices, it will reply to the Modbus TCP client.
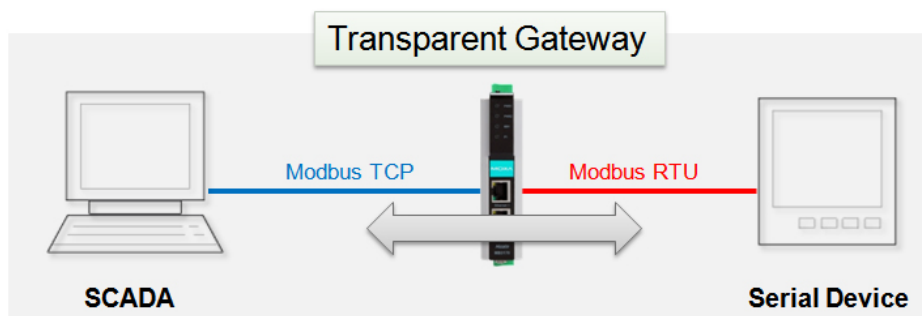
Agent Gateway

An agent gateway provides another way to use a Modbus gateway to transfer the data between Modbus TCP and Modbus RTU. The agent gateway has its own internal memory to store data temporarily, and it will query the data from the connected devices continuously. When the SCADA driver query is received, the gateway will use the data stored on its internal memory to reply to the query. In this way, the gateway works as an agent to query device data actively. This feature can be used for protocol conversion where: (1) the two protocols use different packet structure. For example, PROFIBUS and Modbus, EtherNet/IP, PROFINET, etc. (2) the two protocols use different cycle times. Some protocols—such as PROFIBUS, PROFINET, and EtherNet/IP—exchange data within very short time cycles, which transparent gateways cannot accommodate.