Resolving Modbus Device ID Conflicts

Many of the most common challenges faced when configuring Modbus installations involve conflicts arising from device ID assignments. Device ID reassignment is often required when deploying, updating or reconfiguring installations and that is not always easy or possible.

In an ideal situation, device ID reassignment can be accomplished rather easily with a simple configuration setting or setting a toggle switch. But all too often device IDs can be difficult or, in some cases, impossible to reassign. Devices may be inaccessible or a system is already in operation and changes to the existing installation are not allowed. Furthermore, reassigning device IDs almost always requires changes to the controller logic and there could be problems or barriers to changing that as well. Even when controller logic can be modified, doing so is often time consuming and expensive. Compounding this are everyday problems such as limited project funding, engineering resource availability, aggressive deployment schedules, system testing, and approval processes.

With the goal to solve these problems, Comtrol has developed the Alias Device ID and Device ID Offset functionality. This functionality has been designed to resolve even the most challenging device ID conflicts. Modbus device IDs can be effectively reassigned without modifications to controller logic or slave devices. All it takes is a few entries through logical and easy-to-use web configuration pages. With this advanced functionality, system integrators and plant engineers can now solve device ID conflicts in minutes instead of days, weeks, or months.

MADE IN USA

COMTROL®

Call or email for more information: 1.800.926.6876 | 763.957.6000 | IADSales@comtrol.com

1. Common Causes of Modbus Device ID Conflicts

A. Modbus Specification Limitations

Many device ID conflicts arise simply due to the limitations of the Modbus Specification. The Modbus specification has the following limitations:

- Requires all devices attached to gateway to be addressed by a device ID.
- Allows only 256 device IDs with a range of 0 to 255.
- Not all device IDs can be used for addressing devices.
    - o Device ID 0 is reserved for broadcast messages
    - o 1-247 are for device addressing
    - o 248 to 255 are reserved for such things as gateway functions. Depending on your environment, these device IDs may or may not be available for assignment to devices.

B. Common Implementation Constraints

The following implementation constraints can also cause device ID conflicts:

- It is not always possible or practical to change the device ID of serial Modbus slave devices.
- It is not always possible or practical to modify the device IDs on existing Modbus master programs. For instance, this is often true when adding a SCADA system to an existing PLC controlled system.
- A gateway must route Modbus messages based on the device ID. Therefore, it cannot route to multiple Modbus devices with the same device ID.
- Modbus masters with only one available connection may need to access multiple devices with the same device ID. Furthermore, these devices may be located locally or remotely.

The Alias Modbus Device ID and Device ID Offset functionality have been developed to resolve these device ID conflicts.

2. Alias Modbus Device ID Functionality

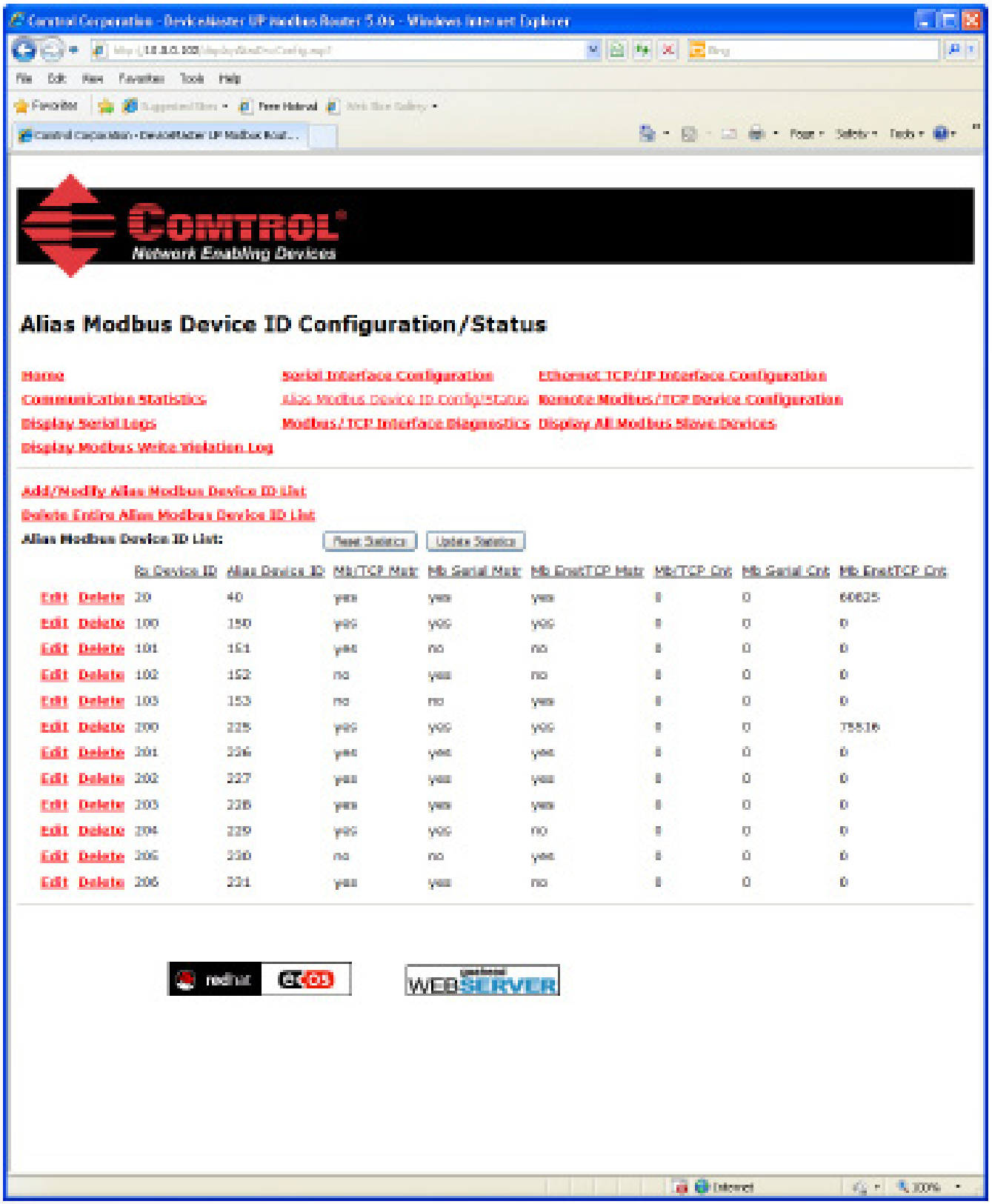
Available on both the Modbus Router and Modbus/TCP firmware applications, the Alias Device ID functionality has been developed primarily to help resolve device ID conflicts involving Modbus masters. These conflicts arise from situations such as:

- A controller requiring connectivity to multiple devices with the same device ID located locally and remotely
- Both controllers and slave devices that cannot be modified, but yet require connectivity
- Multiple controllers requiring access to the same device, but must use different device IDs to access the device
- Single connection controllers, such as serial or Ethernet TCP/IP, that require full usage of the Modbus device ID range

Device ID Aliasing involves:

- Modifying the received device ID to an “aliased” device ID immediately after the message is received from a Modbus master
- The Modbus message is then routed throughout the Modbus gateway or network with the aliased device ID.
- The response message is returned to the Modbus master with the original device ID.
- Modbus masters can communicate to slave devices through either the actual device ID or through an Alias Device ID conversion.

The following web page displays Alias Modbus Device ID Configurations:



3. Device ID Offset Functionality

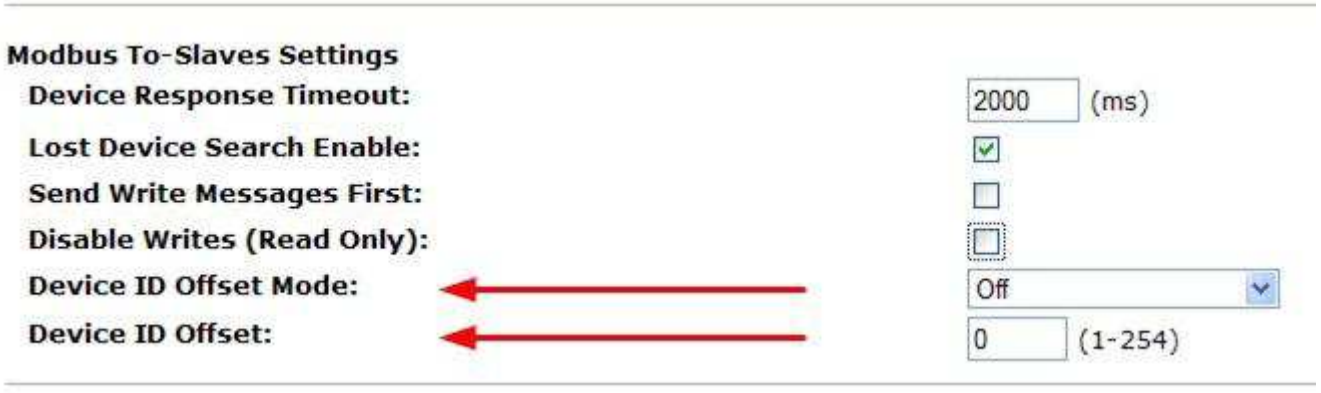
Available on the Modbus Router firmware application, the Device ID Offset functionality has been designed primarily to resolve device ID conflicts involving Modbus slave devices. These conflicts typically arise when slave device IDs cannot be changed and it is desired that two or more slave devices with the same device ID be attached to the same gateway.

The Device ID Offset functionality involves:

- Adding or subtracting an offset to the message device ID immediately before the messages are transmitted out the serial port.
- The device ID, as seen by the Modbus gateway and/or network, is effectively reassigned without any configuration changes to the slave devices.
- The responses, when received from the serial port interface, are immediately converted back to the message device ID and routed back to the Modbus master on the Modbus network.
- The device ID offset is applied to all devices connected to the same serial port.

The Device ID Offset and Alias Device ID functionality can be used together to solve device ID conflicts.

Device ID Offset web page configuration options:

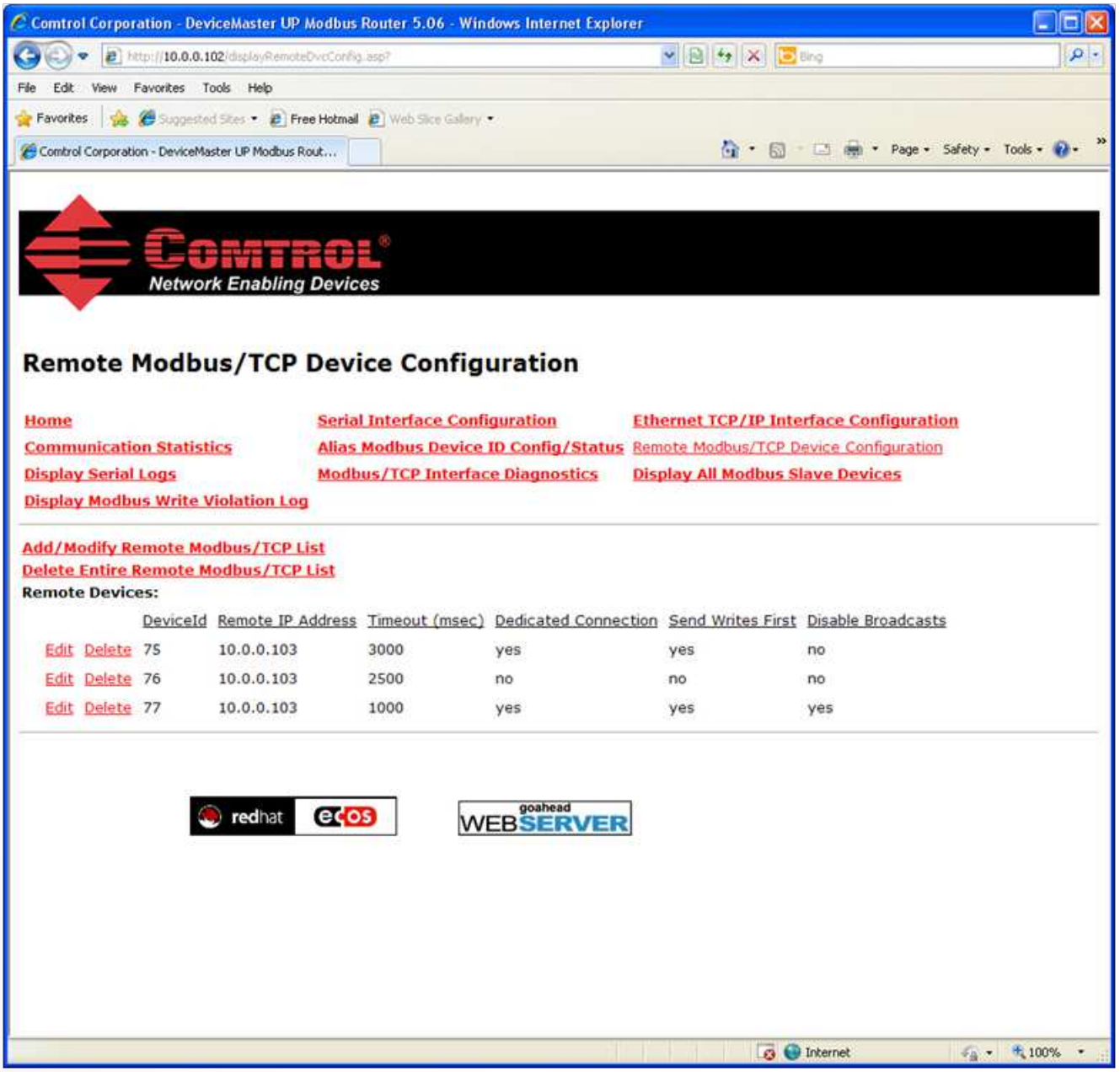


Device ID Offset Mode options include: ***Off, Add-To-Msg-ID, Subtract-From-Msg-ID.***

4. Remote Modbus/TCP Device Connectivity

Available on the Modbus Router firmware application, the Remote Modbus/TCP Device ID functionality provides connectivity to either Modbus/TCP slaves or serial Modbus slave devices connected to other Modbus gateways. The Remote Device Routing functionality can be used in conjunction with the Alias Device ID and Device ID Offset functionality to provide network-wide Modbus connectivity solutions.

The following web page capture displays Remote Modbus/TCP Device Configurations:

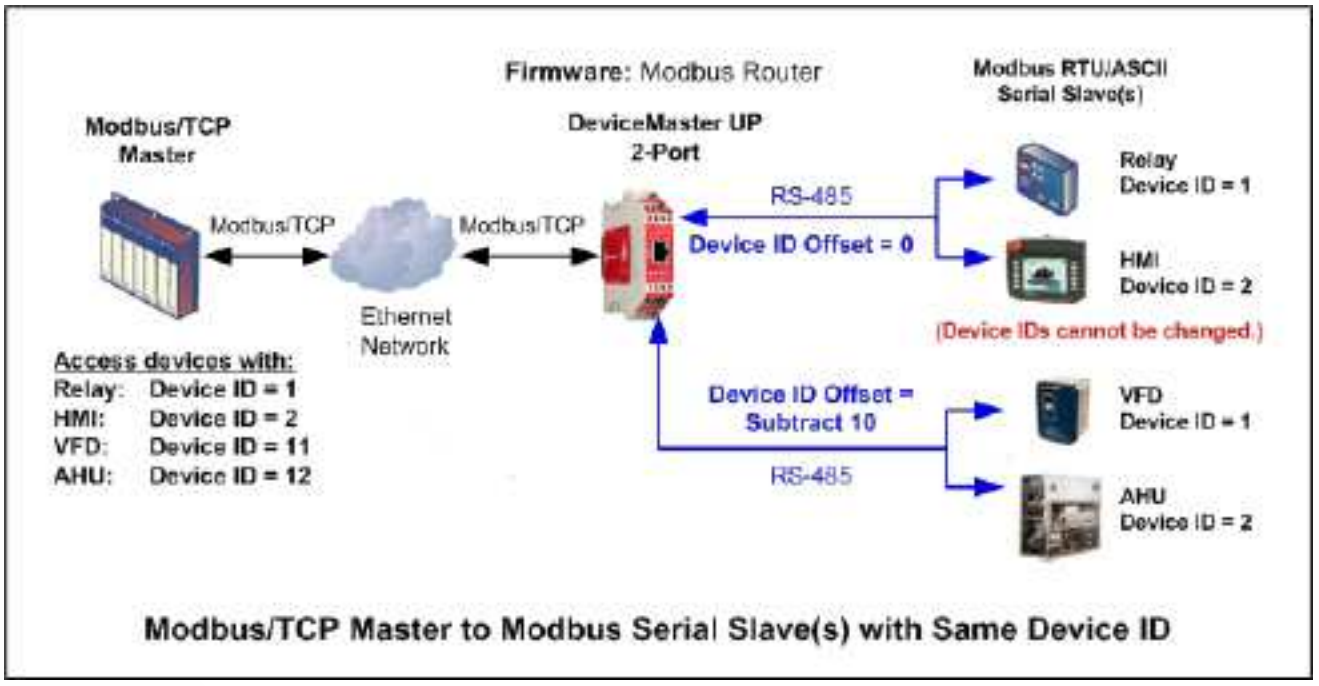


5. Solutions to Common Device ID Conflicts

A. Modbus/TCP Master Communicating to Local Device(s) with Same Device ID

Problem: A Modbus/TCP master needs to communicate to different devices on the same gateway with the same device IDs. It is desired to connect all devices to the same gateway.

Solution: Use Device ID Offset to provide unique device IDs for each device.

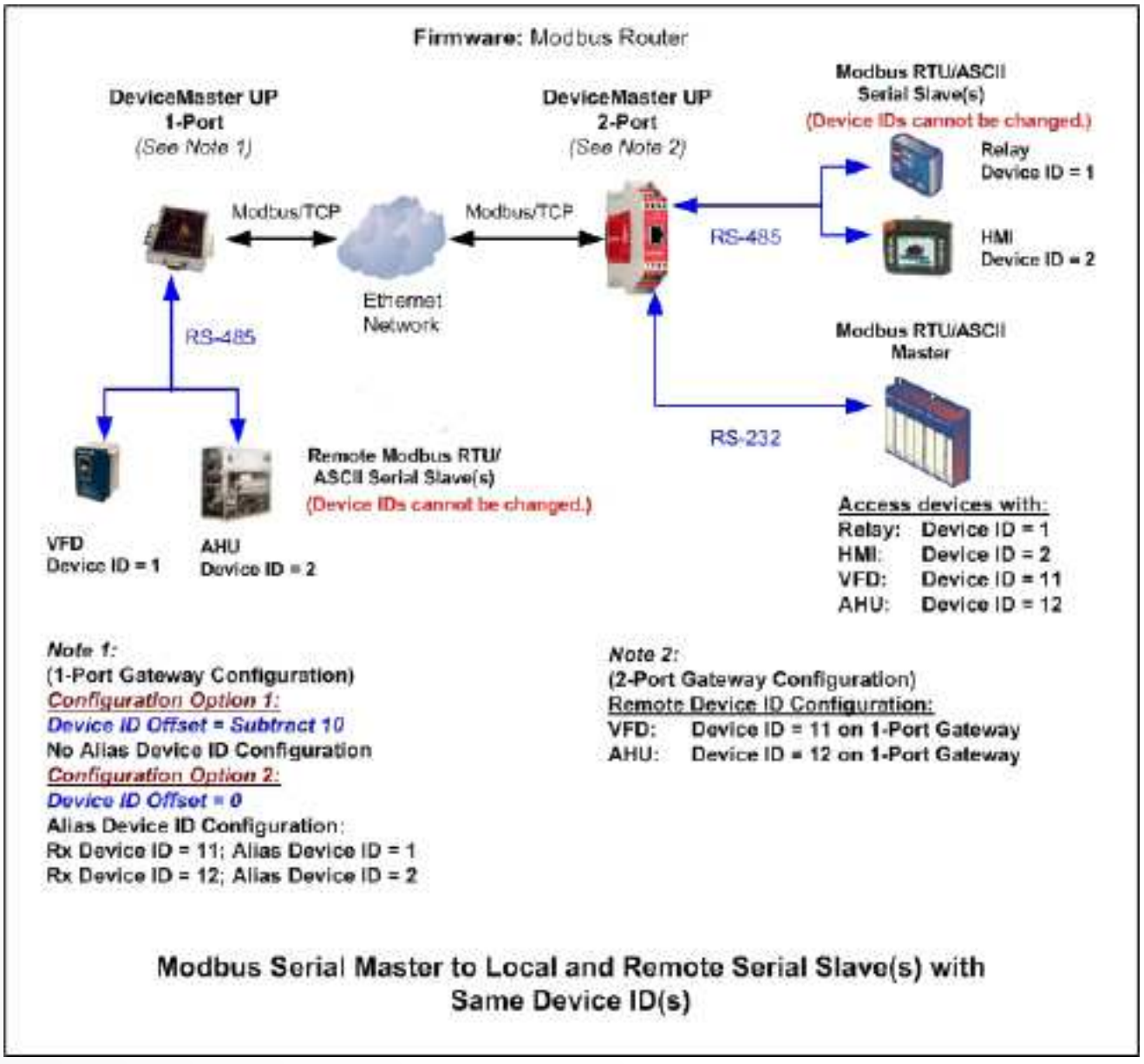


Modbus/TCP Master to Modbus Serial Slave(s) with Same Device ID

B. Modbus Serial Master Communicating to Local and Remote Devices with Same Device IDs

Problem: A Modbus Serial Master needs to communicate to different devices with the same device IDs. Two are connected locally and two are connected remotely via another gateway.

Solution: Use Remote Device Routing and Alias Device ID functionality to provide unique device IDs for both the remote devices.



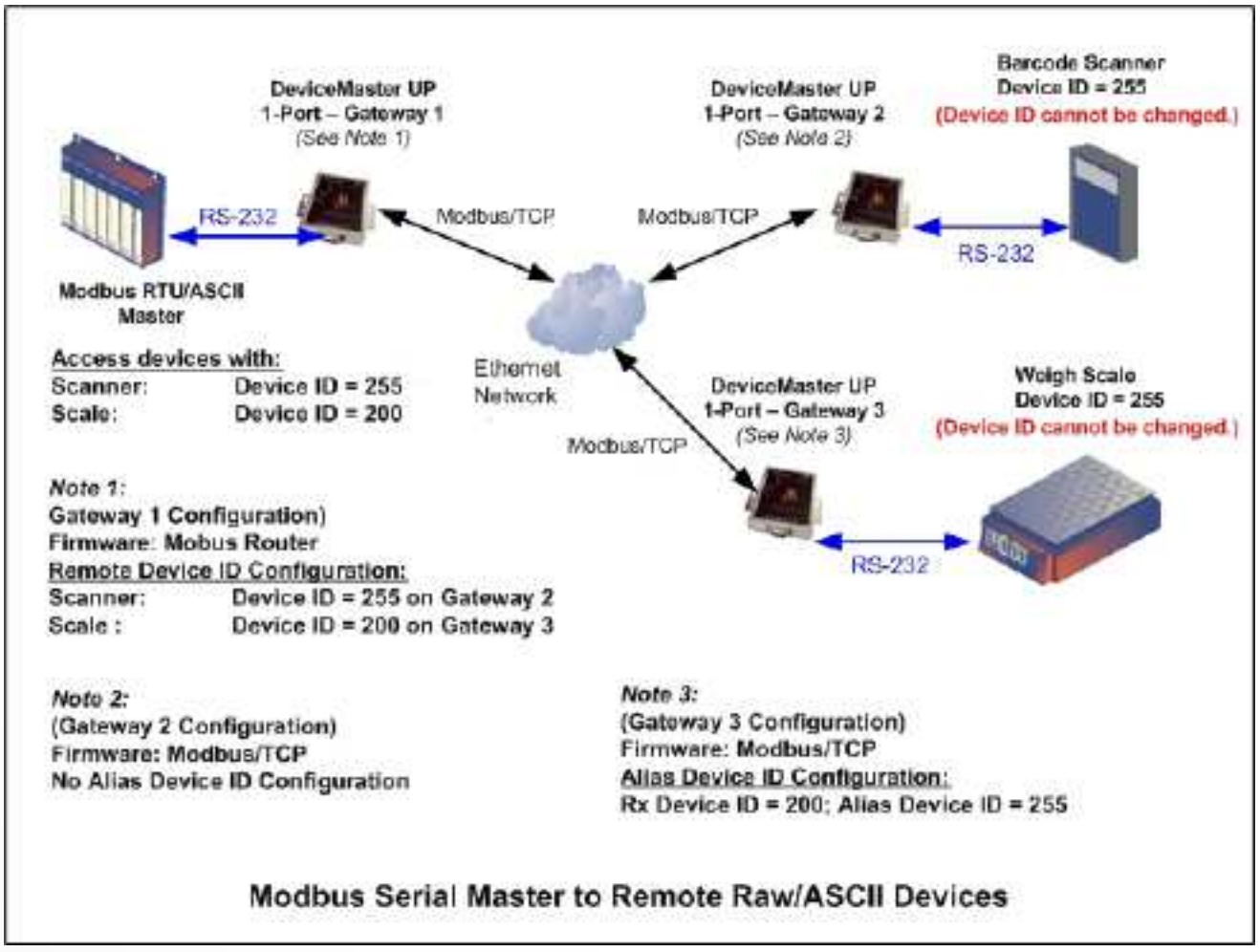
Modbus Serial Master to Local and Remote Serial Slave(s) with Same Device ID(s)

C. Modbus Serial Master Communicating to Two Remote Serial Raw/ASCII Devices

Problem: A Modbus Serial Master requires connectivity to two remotely located raw/ASCII devices.

Solution:

- Use one gateway to provide Modbus/TCP connectivity from the serial master to the Modbus network.
- Use one gateway to provide communication to each remote raw/ASCII device.
- Use Remote Device Routing and Alias Device ID functionality to provide communication to the serial raw/ASCII devices.



Modbus Serial Master to Remote Raw/ASCII Devices

D. Merging Two Serial Modbus Networks

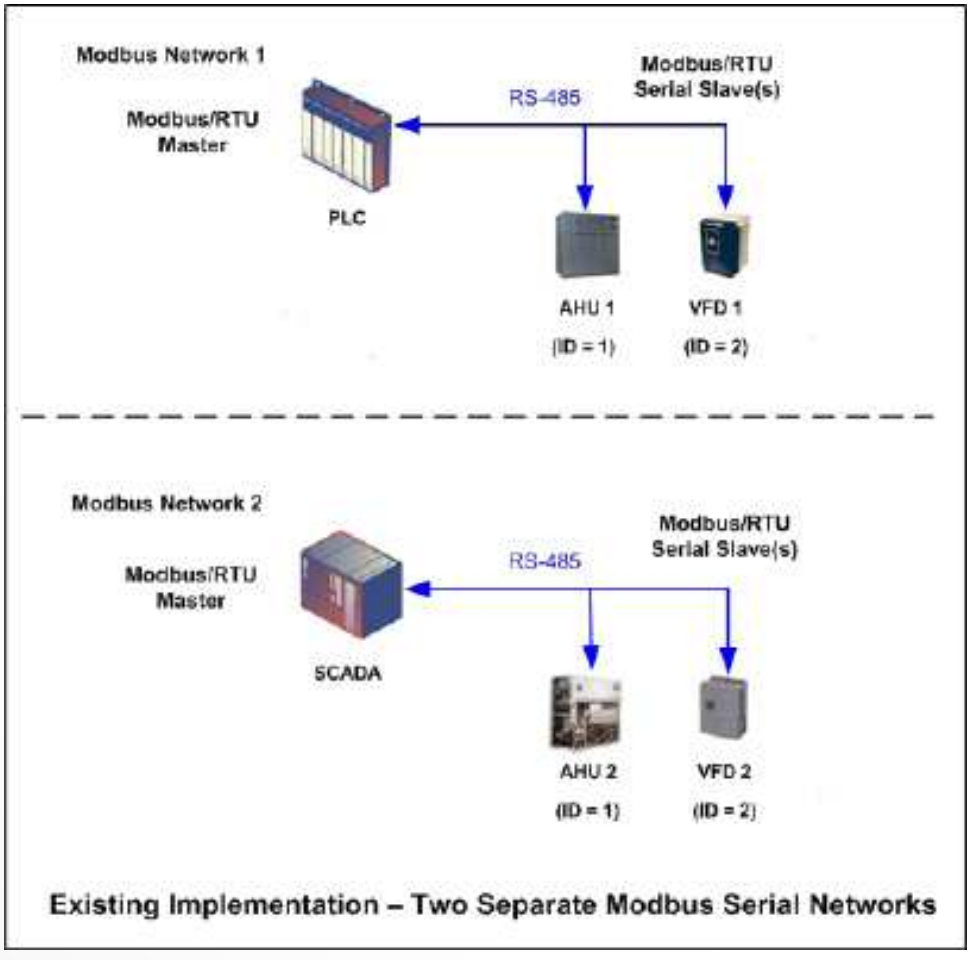
Problem:

- Two serial Modbus networks, each with a controller and slave devices, need to be merged into one.
- Each controller needs access to all of the slave devices.
- To limit the engineering effort, it is desired to limit the scope of the project by:
  - o Retaining the existing Modbus master program logic and slave device IDs.
  - o Limiting the project to only adding the additional connectivity and corresponding control logic.

Solution:

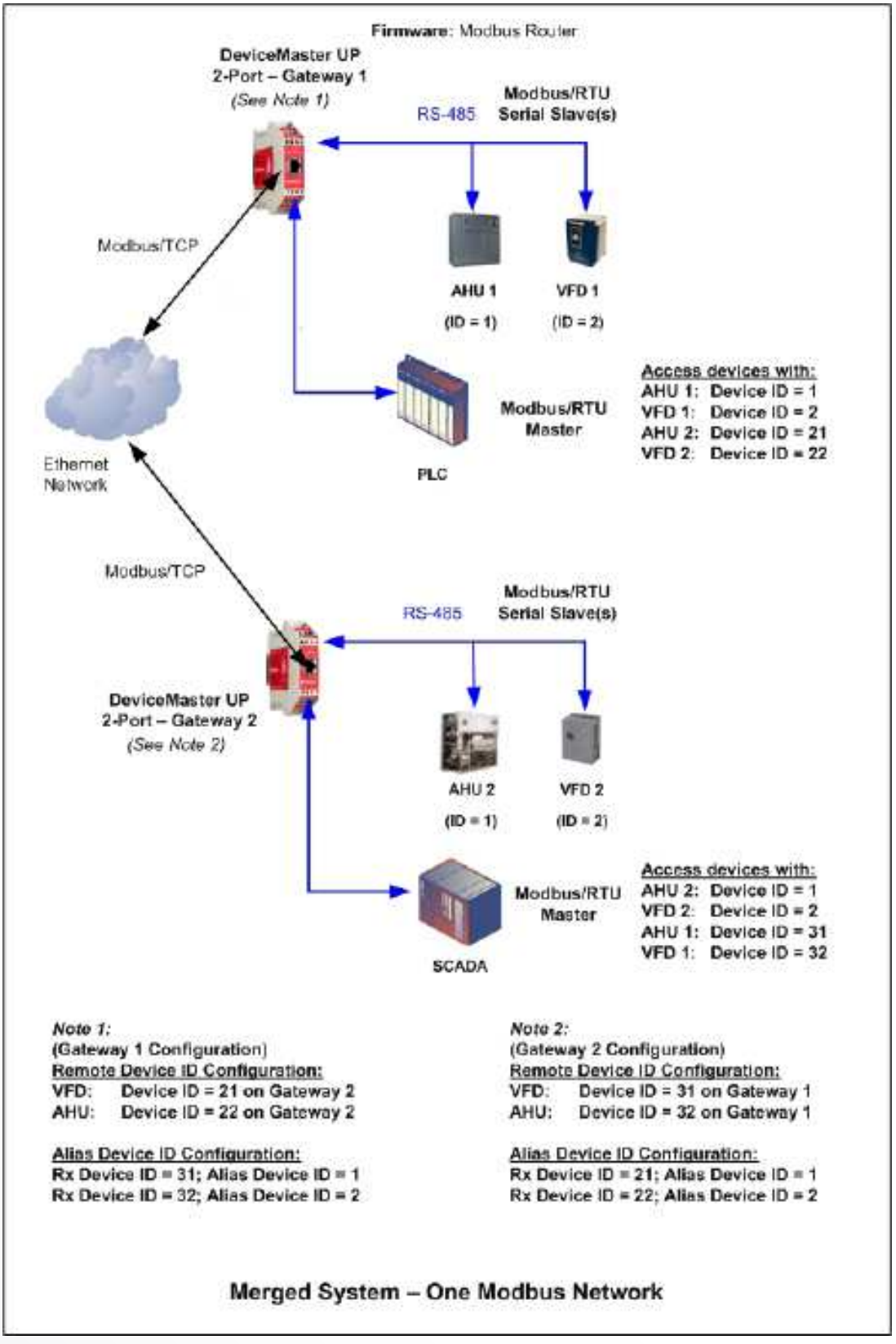
- Place a 2-Port DeviceMaster UP with Modbus Router between each controller and its corresponding slave devices.
- On each gateway, use Remote Device Routing and Alias Device ID functionality to provide access to the devices attached to the other gateway.
- Each slave device will then be accessed locally using its assigned device ID and from the remote controller using its aliased device ID.

Existing Systems:



Existing Implementation – Two Separate Modbus Serial Networks

Merged System:



Merged System – One Modbus Network

E. Providing Modbus Connectivity between Separate Ethernet Networks

Problem:

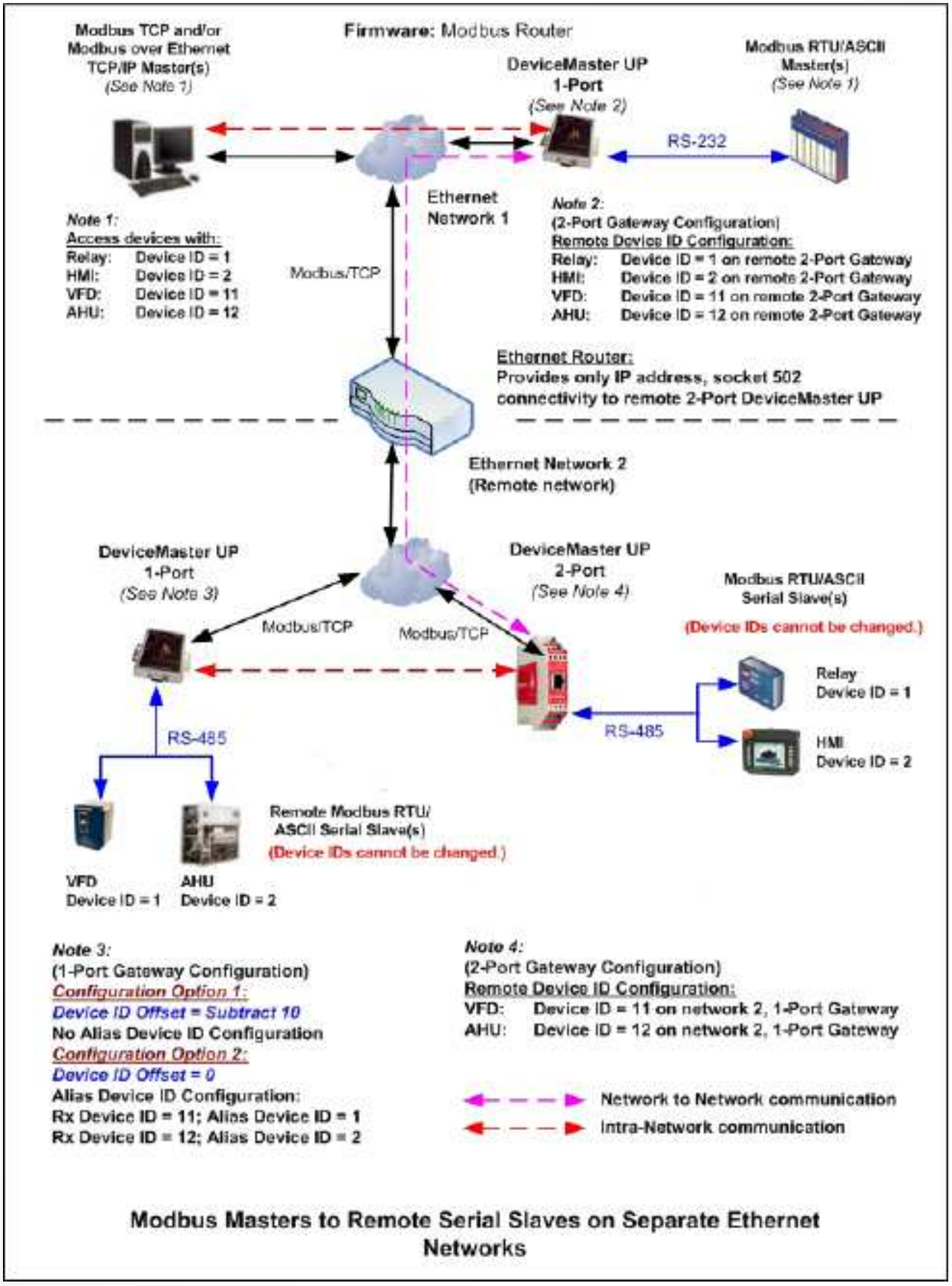
- Modbus Controllers on one Ethernet network need to connect to remote Modbus slave devices on a separate Ethernet network through an Ethernet Router.

Solution:

Using DeviceMaster UP gateways running Modbus Router firmware:

- Connect one or more DeviceMaster UP gateways to the Ethernet Network containing Modbus controllers. Connect any serial Modbus controllers to the DeviceMaster UP serial ports.
- Use one or more DeviceMaster UP gateways connected to the second Ethernet network to provide connectivity to the remote Modbus slaves.
- Use Remote Device Routing and, if needed, Alias Device ID functionality to provide connectivity from the Modbus masters to the remote Modbus slaves.

The solution is demonstrated in the following diagram:



Modbus Masters to Remote Serial Slaves on Separate Ethernet Networks