Modbus TCP/IP

TCP/IP

TCP is Transmission Control Protocol and IP is Internet Protocol. These protocols are used together and are the transport protocol for the internet. When modbus information is sent using these protocols, the data is passed to TCP where additional information is attached and given to IP. IP then places the data in a packet (or datagram) and transmits it.

TCP must establish a connection before transferring data, since it is a connection-based protocol. The Master (or Client in Modbus TCP) establishes a connection with the Slave (or Server). The Server waits for an incoming connection from the Client. Once a connection is established, the Server then responds to the queries from the Client until the client closes the connection.

Modbus RTU over TCP

Simply put, this is a Modbus RTU message transmitted with a TCP/IP wrapper and sent over a network instead of serial lines. The Server does not have a SlaveID since it uses an IP Address instead.

Modbus TCP

A Modbus Messaging Implementation Guide provided by Schneider Automation outlines a modified protocol specifically for use over TCP/IP. The official Modbus specification can be found at www.modbus-ida.org. The main differences between Modbus RTU and Modbus TCP are outlined here.

ADU & PDU

Aside from the main differences between serial and network connections stated above, there are a few differences in the message content.

Starting with the Modbus RTU message and removing the SlaveID from the beginning and the CRC from the end results in the PDU, Protocol Data Unit.

Here is an example of a Modbus RTU request for the content of analog output holding registers # 40108 to 40110 from the slave device with address 17.

```
11 03 006B 0003 7687
```

11: The SlaveID Address (17 = 11 hex)
03: The Function Code (read Analog Output Holding Registers)
006B: The Data Address of the first register requested. (40108-40001 = 107 = 6B hex)
0003: The total number of registers requested. (read 3 registers 40108 to 40110)
7687: The CRC (cyclic redundancy check) for error checking.

Removing the SlaveID and CRC gives the PDU:

```
03 006B 0003
```

MBAP Header

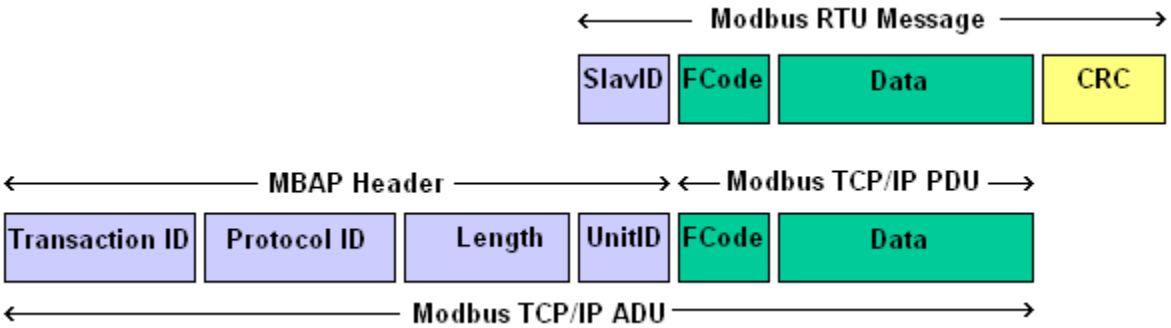
A new 7-byte header called the MBAP header (Modbus Application Header) is added to the start of the message. This header has the following data:

Transaction Identifier: 2 bytes set by the Client to uniquely identify each request. These bytes are echoed by the Server since its responses may not be received in the same order as the requests.

Protocol Identifier: 2 bytes set by the Client, always = `00 00`

Length: 2 bytes identifying the number of bytes in the message to follow.

Unit Identifier: 1 byte set by the Client and echoed by the Server for identification of a remote slave connected on a serial line or on other buses.



Summary

The equivalent request to this Modbus RTU example

```
11 03 006B 0003 7687
```

in Modbus TCP is:

```
0001 0000 0006 11 03 006B 0003
```

`0001`: Transaction Identifier
`0000`: Protocol Identifier
`0006`: Message Length (6 bytes to follow)
`11`: The Unit Identifier (17 = `11` hex)
`03`: The Function Code (read Analog Output Holding Registers)
`006B`: The Data Address of the first register requested. (40108-40001 = 107 = `6B` hex)
`0003`: The total number of registers requested. (read 3 registers 40108 to 40110)

TCP/IP Wrapper

CONSTRUCTION OF A TCP/IP-ETHERNET DATA PACKET