

# Common Architectures

One of the strengths of Digi's line of IA-focused products is their flexibility. Since no two solutions are the same, Digi products have been designed to correctly facilitate the connection in any situation.

The following is a list of potential IA architectures designed with Digi products. Each example shows how the Digi device server may be installed in various network scenarios in order to meet specific network requirements. The Digi One™ IAP is used in the following examples.

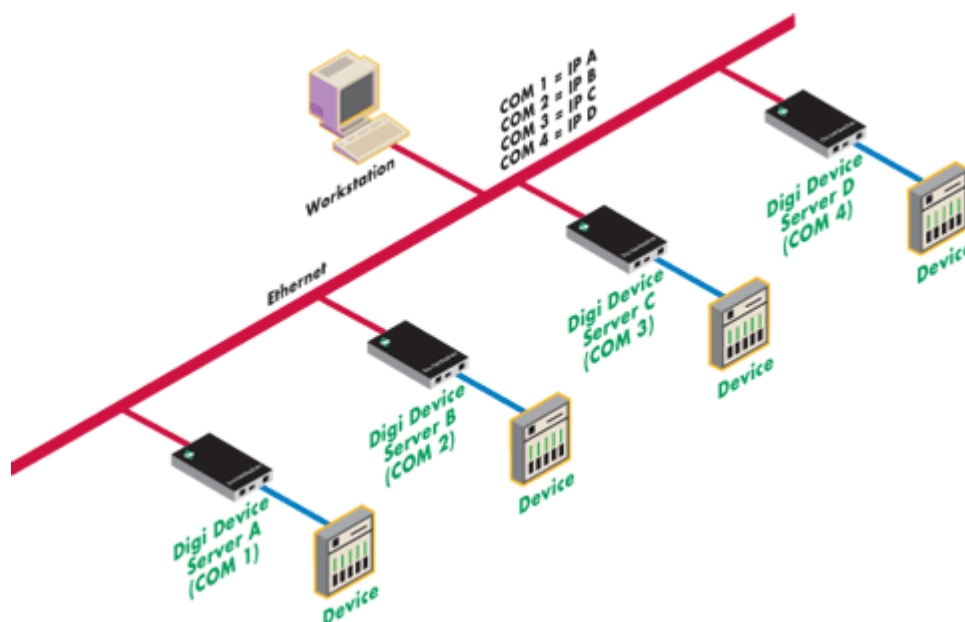
**RealPort®**  
**Modbus/TCP**  
**Allen-Bradley DF1**  
**TCP Socket**  
**UDP Socket**

**UDP Multicast**  
**TCP Tunnel**  
**UDP Tunnel**  
**PPP Model**

## RealPort®

When your solution includes software running on a PC, and that software uses COM ports to communicate with devices, use only the best COM port redirection solution –

Digi's patented RealPort software. RealPort is the only implementation that allows multiple connections to multiple ports over a single TCP/IP connection. This means RealPort supports the maximum number of remote devices.



## Industrial Protocols with Multi-Master Support - Modbus/TCP and Allen-Bradley DF1

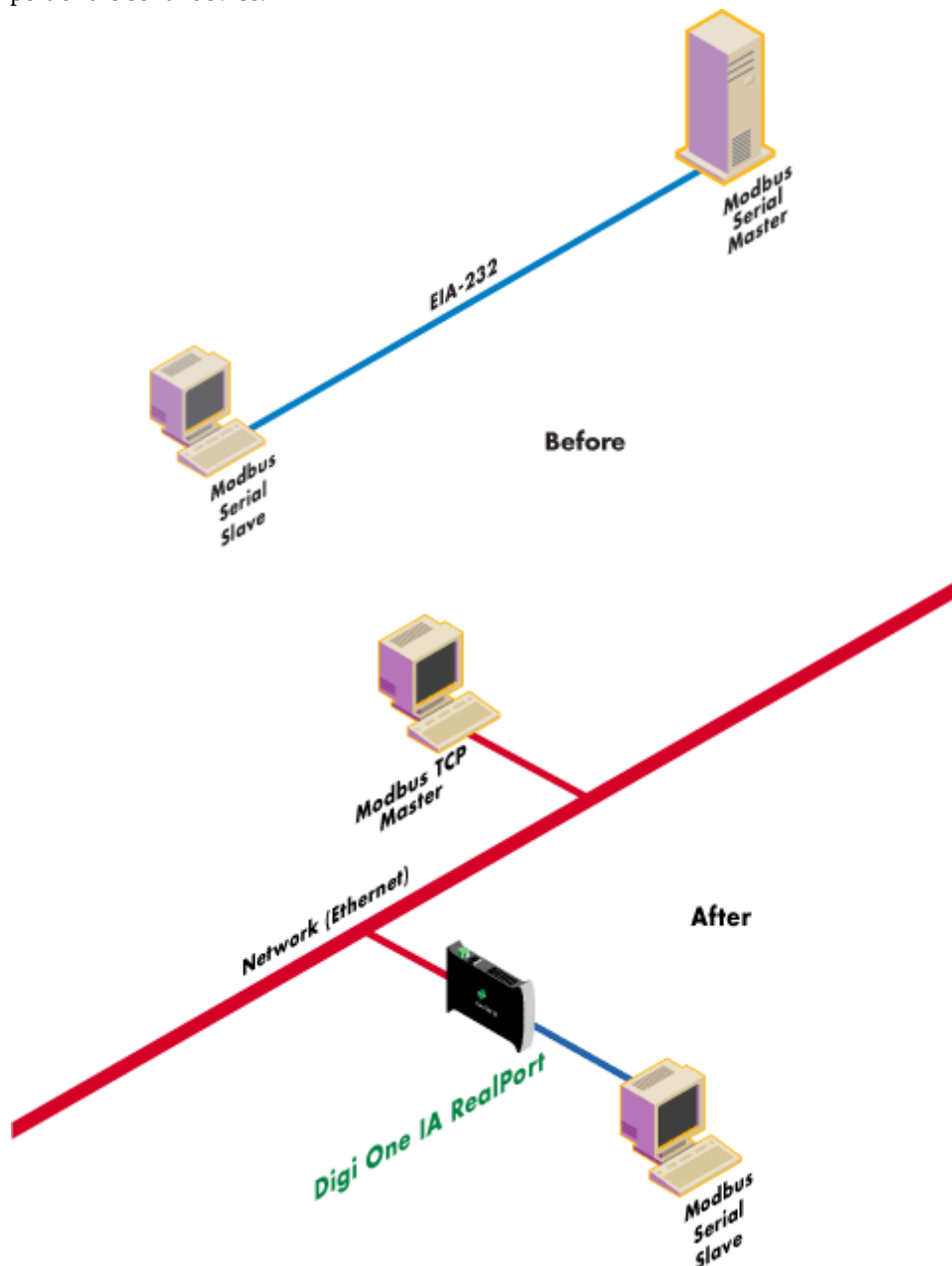
### Modbus/TCP

In recent years, Modbus has been adapted to the Ethernet by an encapsulation of the serial protocol into a TCP/IP packet. This new protocol is called ModbusTCP. ModbusTCP allows many devices to communicate over a standard Ethernet network.

The Digi One IAP device server acts as a Modbus to ModbusTCP gateway. Simply plug any Modbus serial device into the serial connection on the Digi One IAP, then plug a network cable into the RJ-45 Ethernet port. Digi One IAP converts the Modbus information into ModbusTCP packets into and out of the device. This means that any host can now transparently connect to the Modbus serial device using ModbusTCP. No changes need to be made to the host or slave device. The DigiOne IAP can act as both a master and a slave.

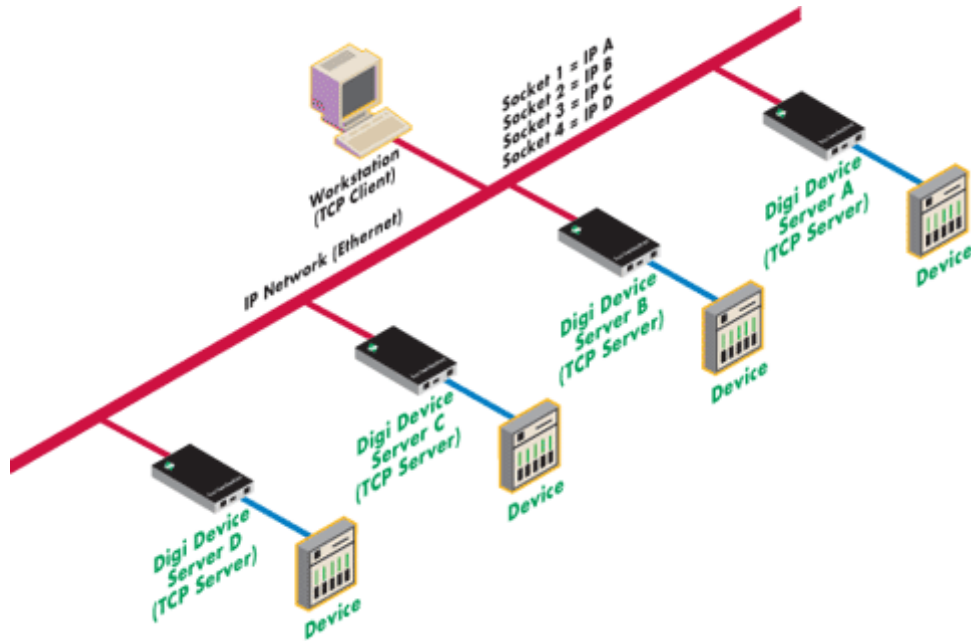
### Allen-Bradley DF1

The same concept as above for Modbus applies to the Allen-Bradley (AB) conversion. You can connect any Allen-Bradley DF1 serial device to a DigiOne IAP and be able to communicate to that device using Allen-Bradley Ethernet or EthernetIP. This means whether you have a software application or another AB networked device, either can communicate across Ethernet to the DigiOne IAP, which is connected to the DF1 serial port of the serial device.



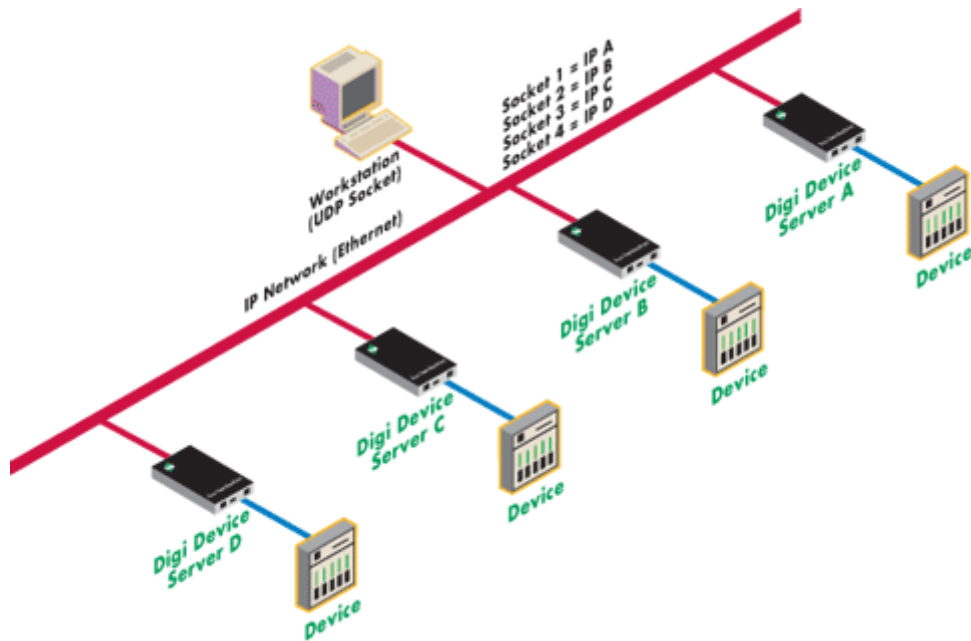
## TCP Socket

This is the most common model used when writing a new application to control devices on the network. The TCP Socket programming model is straightforward and reliable. It also works with Telnet applications. Many applications that support serial (or COM) ports also support TCP Socket or Telnet.



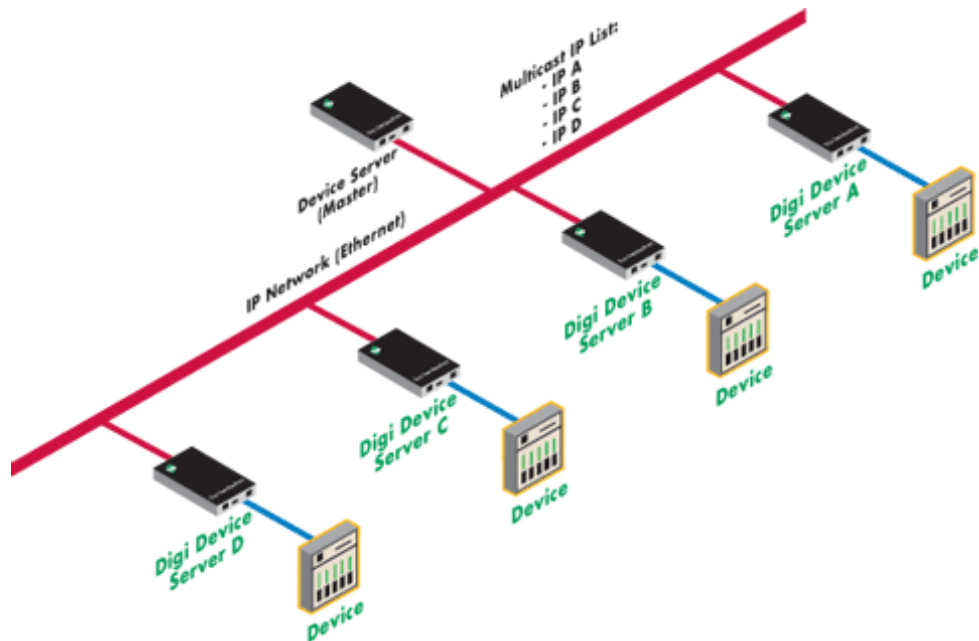
## UDP Socket

Like TCP Socket, this programming model is straightforward and reliable. Some applications are written to use UDP because it is even simpler than TCP. UDP does not guarantee delivery of messages, so the application is responsible for validating whether or not data was received.



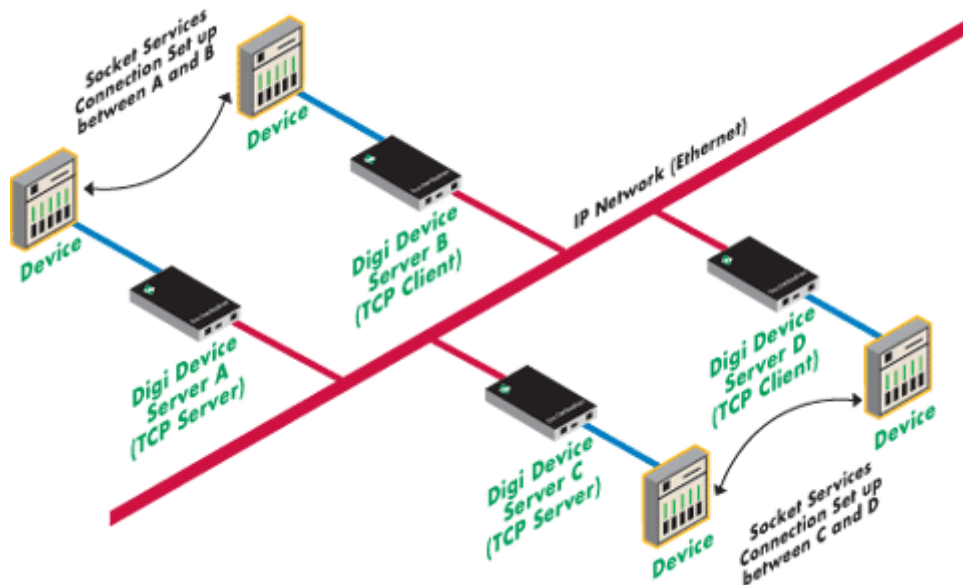
## UDP Multicast

UDP Multicast is the only solution that automatically sends messages to multiple recipients. It allows you to effectively broadcast messages to several recipients so that they update simultaneously.



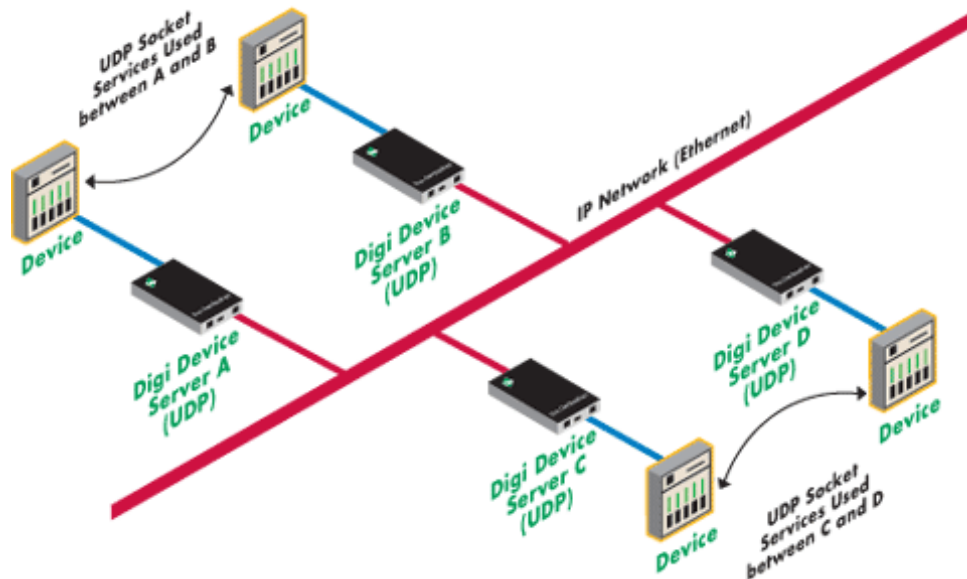
## TCP Tunnel

When both of your devices (sender and receiver) only have serial ports (no network ports) and you need to extend the connection using Ethernet, use TCP Tunnel. This model provides a clear channel connection across the network for passing raw serial data. It also guarantees delivery of data.



### UDP Tunnel

Like TCP Tunnel, use this model when both of your devices (sender and receiver) only have serial ports (no network port) and you need to extend the connection using Ethernet. Less widely used than the TCP Tunnel model, this model provides a clear channel connection across the network, but doesn't require the overhead of TCP sessions. UDP Tunnel is sometimes preferred when a very large number of devices live on the network. This model does not guarantee delivery of data so the devices themselves must manage and recover from lost data.



### PPP Model

Your serial device knows about networking, understands TCP/IP, but communicates using a serial port. In this case, the device server acts as a simple bridge onto the network to give your device access to the network and all the services it provides.

