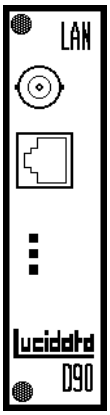
D90-TCP MODULE

For connection to an Ethernet network



Contents

Introduction

Like all Lucidata Diplomat data communications products, the D90-TCP module has been designed to be easy to use in most normal applications whilst retaining a large degree of flexibility.

It is Lucidata's policy to try and find out as much about the intended application of its products before shipment so that the unit can be pre-configured at the factory for easy installation. In this case you may skip many sections of this manual. If the application changes with time then you will need to refer to those sections to reconfigure the unit.

In the event of difficulty, please contact Lucidata's technical support staff who will be able to guide you through the process.

Description

The D90-TCP module is implemented on a LAN hardware module which is fitted with an RJ45 connector. The LAN module thus allows connection to an Ethernet network with 10Base-T interfaces via UTP cable or optionally, 10Base-2.

Within the module there is an EPROM which contains the program (Firmware) and various communication parameter defaults which are set when the module is powered up. These defaults can only be changed by installing new firmware or by means of a Diplomat Network Monitor (DNM) if one is present in the rack. The defaults for these parameters will be found in the *Customer System Summary* section of this manual which is specific to each customer.

The functional characteristics are controlled by certain parameters which can be changed by means of the *Configuration Bytes* described later in this section.

The D90-TCP module will normally be providing a Local Server function for other modules on the rack although on the network it can appear as a Client or a Server.

Interfaces

The TCP module is fitted with an RJ45 jack for connection to an Ethernet Hub using Unshielded Twisted Pair (UTP) cabling.

The pinouts for the RJ45 jack are as follows:

Pin 1	Tx+
Pin 2	Tx-
Pin 3	Rx+
Pin 6	Rx-

LED Indicators

There are three LEDs on the front panel of the D90-TCP module. The indicators are Yellow(Y) and Green(G) and have the following meanings when illuminated.

O (G) There is a good 10Base-T connection
O (Y) Flashes each time a packet is received
O (G) Flashes whenever module is communicating with another D90 module

Configuration

The TCP module contains two Configuration Bytes of “Silicon Switches”. These are used to control the low level behaviour of the module. The switches are listed in ascending order below together with a description of their function.

By convention the switches or bits of a configuration byte are numbered as follows

7	6	5	4	3	2	1	0

The meaning of the bits in Configuration Byte A are given in the following table.

Bit	Name	Description
0	BPEER	[0] – Do not service remote Client requests when configured as a Client. [1] – Allow remote Clients to initiate a session even when configured as a Client.
1	BALDPO	[0] – Insist that new TCP session attempts or RST commands must come from the same remote host and port as the current session that is about to be terminated. [1] – Allow a different port on the same current remote host to initiate a new session or RST the existing session.
2	BALDIP	[0] – Insist that new TCP session attempts or RST commands must come from the same remote host as the current session that is about to be terminated. [1] – Allow a different remote host from the current remote host to initiate a new session or RST the existing session.
3	Reserved	[0] – Must be zero
4	BSERVR	[0] – Module behaves as TCP Client [1] – Module behaves as a TCP Server
5	BSTART	[0] – Establish session on data present [1] – Establish session on module becoming ready
6	BMONIT	[0] – Normal operation [1] – Monitor the Network and send the first 60 bytes of all packets seen to a DNM if installed for display or printing. The module does not perform its normal function when so configured.
7	BEXREP	[0] – Do not send Network related error messages to Local Client modules in human readable form. [1] – Send Network related error messages to Local Client modules in human readable form.

The meaning of the bits in Configuration Byte B are given in the following table.

Bit	Name	Description
0	BETCP	[0] – Reject all TCP packets [1] – Enable the processing of TCP packets
1	BEUDP	[0] – Reject all UDP packets [1] – Enable Custom Processing of UDP packets
2	Reserved	[0] – Must be zero
3	Reserved	[0] – Must be zero
4	BMSPO	[0] – Must be zero if either BMDPO=1 or BMAUTO=1 [1] – Allocate a Local Client slot based on the Source TCP Port of SYN packet minus Base Remote TCP Socket value
5	BMDPO	[0] – Must be zero if either BMSPO=1 or BMAUTO=1 [1] – Allocate a Local Client slot based on the Destination TCP Port of SYN packet minus Base Local TCP Socket value
6	BMAUTO	[0] – Must be zero if either BMSPO=1 or BMDPO=1 [1] – Allocate next available Local Client slot to next TCP session to be started. This is the default setting if both BMSPO=0 and BMDPO=0.
7	BEXREP	[0] – Do not send informative messages to DNM module if installed. [1] – Send informative messages to a DNM module if installed.

Note: So called 'Panic Debug' messages will be sent to an installed DNM module no matter what the setting of BEXREP is. Should a Panic Debug message ever appear it should be copied down and reported to Lucidata Technical Support – email support@lucidata.com

Initialisation

On power-up the TCP firmware delays for about ten seconds to allow time for any Diplomat Network Monitor (DNM) module to finish configuring the rack. If no DNM is present then the TCP module will use its own EPROM based IP Addresses, Sub-net mask and Service Socket.

During this delay the TCP module will have received details of all other modules on the rack and also sent its own particulars to anyone who asked.

If the TCP module has been configured as a Client, after this initial delay the TCP module will use the current values of IP Addresses and Sub-net mask to generate an ARP request to find the Ethernet address of the device it will be talking to. This could be the Remote Server if it is on the same network or a local Gateway. The firmware checks every 15 seconds to see if it has a valid ARP table entry and if not it sends another ARP. The TCP module can do nothing else until it receives a valid response to this ARP.

If the TCP module has been configured as a Server, it just sits and waits for a remote Client to start a TCP session.

Firewall

If a Firewall Table has been defined (see D90-DNM section) then the D90-TCP module will only accept TCP connection attempts from Client hosts whose IP Address is found in the Firewall Table. If a Port is also defined in the Firewall Table entry for a given host then the connection attempt must also be shown to come from that Source Port. The Firewall Table may also be used when the D90-TCP module is configured as Client or Peer and a Port Mapping is chosen which assigns consecutive Firewall Table entries to consecutive Slot numbers.

If UDP datagrams are being used the Firewall Table is used to validate the source of all UDP datagrams received.

Number of Sessions

The normal limit on number of active TCP sessions or UDP associations is 32. This value is not configurable by the user but can be factory defined prior to shipment.

A timeout value may be defined by the user to cause UDP associations to be broken after a certain time as being a connectionless protocol there is no natural way of "dropping" a UDP association.

Operation

There may be more than one D90-LAN module on a single rack and they could be running different firmware. However each LAN module will have one or more Local Client modules such as D90-SA2s associated with it. Normally the Local Client modules are fitted to the rack immediately to the right of a Local Server module but they do not have to be. To keep the discussion simple we will assume there is a D90-LAN module in slot 1 running D90TCP firmware and a few D90-SA2 modules running D90-ASR firmware in slots 2,3,4 etc.

If the TCP module is configured as a Client it just acts as a concentrator for each of the modules. As a module initiates a session the TCP module makes the request to the single remote host on the single remote TCP socket defined in the TCP modules configuration parameters. The source port used for the slot to identify its session from everyone elses using the same TCP module is calculated as follows. The Local TCP Socket value, let us call it N, defined in the configuration parameters is associated with the TCP module itself. The module in the next slot to the right of the TCP module gets N+1 assigned to its Port A and N+2 assigned to its Port B. The next module gets N+3 for its Port A and N+4 for its Port B and so on.
The TCP module sends status messages to Local Client modules whenever a TCP session is started, closed, lost or aborted.

If the TCP module is configured as a Server it has three different ways of assigning an incoming connection request to a Port resource.

Auto Assignment

If there is no difference between the devices connected to the ports of the modules then a new request can be assigned to any port. In this case all remote hosts attempts to start a session on the single Local TCP Socket value defined in the configuration parameters. The TCP module makes the assignment to the next available and ready port. Subsequent TCP packets are identified by their source IP and port values. If no ports are available an appropriate ICMP message is sent to the remote host.

Destination Port Steering

In this case the remote host in effect specifies which slot and port it wants to set up the session with. The destination port used to set up the TCP session is calculated as described in the Client Mode above i.e. starting at the base Local TCP Socket value + 1 for the A Port of the first module, +2 for the B Port etc. If the specific port is not available or not ready an appropriate ICMP message is returned to the remote host.

Source Port Steering

This method is used when two D90 systems are being used back-to-back as a multiplexer effectively. The source socket of a connection attempt has the value of the base Remote TCP Socket subtracted from it to produce an index N. This identifies the module and Port of the originating connection attempt using the same arithmetic we performed earlier. The TCP module uses this value to make the association with the same module on the local rack. Thus a one-to-one mapping is achieved between the slots and ports of the two racks.