MAINFRAME-BASED TCP/IP stacks are ubiquitous, and are used to allow access to mainframe data from other platforms and to transfer files around the network.

Accessing mainframe data can be achieved using a telnet service called TN3270E (in reality a standard 3270 screen protocol inside special headers and trailers giving a telnet packet). There are TCP/IP stacks on the mainframe platform and on the other platforms (e.g. PCs and UNIX boxes) in the network and they handle the necessary handshaking and verification to ensure the data is transferred successfully. A socket interface is then used at the mainframe end to connect with CICS, for example.

Data transfer uses FTP. The mainframe and the other platforms use the TCP/IP protocol to ensure that all the packets that are sent arrive successfully.

When there is a problem with the network, it can be very hard to identify. In fact, it can be hard deciding where to look first. There are many ways of finding out where problems may have occurred and this article describes some of them.

COMMANDS

Let’s start with the simplest method, which is to use NETSTAT commands. The NETSTAT command is used to display the TCP/IP network protocol statistics and information. It can be used from Windows and UNIX platforms as well as from the mainframe. With USS, the command is ONETSTAT. The following commands can be used from TSO:

- **NETSTAT option [TCP procname]** displays the network status of the local host.
- ? can be used for a list of all the options.
- **NETSTAT ALLCON|CONN** displays port connections for the TCP/IP stack.
- **NETSTAT ARP ALL|ipaddress** displays ARP cache for the TCP/IP stack.
- **NETSTAT DEV** displays the status of the device(s) and link(s) for the TCP/IP stack.
- **NETSTAT GATE|ROUTE** displays routing information for the TCP/IP stack.
- **NETSTAT HOME** displays IP address(es) for the stack.

NETSTAT commands are probably most useful when you already have some idea of what the problem is and you are looking for more information. Although NETSTAT commands indicate active ports, they provide little information about non-TCP activity on the network (such as UDP, ICMP, and OSPF) because these protocols are connectionless in nature. There is no way of showing connections being rejected because a server is down or simply unable to accept new connections. Another difficulty with NETSTAT is that users must remember the commands (or create a REXX EXEC containing them). There is no simple GUI (Graphical User Interface). NETSTAT commands do not form the basis for regular monitoring because of the potentially large amount of data that can be returned and the difficulty of doing anything with this data—like having to screen scrape it into another application.

Another useful command is PING. This is used to test the connection between the local server and a remote server. The name and password of the administrator client issuing this command must be defined on the remote server and the person issuing the command must have system privilege.

To test the connection to a server called server1 the command would be:

```
ping server1
```

From TSO:

```
PING hostname
```
This would send an echo request to a host name or address to determine whether the computer is accessible.

PING can also be used as a simple measure of response time. PING sends out a block of data using an ICMP echo request. The response time is how long it takes for that block of data to be returned. It is not a reliable measure of response time because most users are not using ICMP (which may travel over a different route through the network compared to TCP or UDP traffic), ICMP packets can be given different priorities by intermediate routers, and users send and receive different sized packets. Another problem is that sending a host of PINGs is a way to attack a stack, so firewalls are often configured to block PINGs.

The traceroute command is used to identify the hops from one part of the network to the next. With z/OS, the command is TRACERTE, with DOS it’s tracert, and with UNIX it’s traceroute. From TSO you would enter:

```
TRACERTE hostname
```

Some MVS or VTAM commands can be used to help diagnose TCP/IP performance problems. z/OS console commands for TCP/IP include:

- V TCPIP,[procname],Telnet,xxx to perform a specified function for TELNET.
- VTAM commands related to OSA cards that might be useful include:
  - D NET,ID=name displays the network named in the ID field. Additional parameters that may be added include:
    - SCOPES=ONLY|ACT|ALL|INACT,E which gives extended information about the node.
  - D NET,MAJNODES|APPLS shows the status of all the active major nodes or applications.
  - D NET,PENDING lists all the nodes in pending states.
  - V NET,ACT,ID=name activates the VTAM resource identified by the name.
  - V NET,INACT,ID=name inactivates the VTAM resource identified by the name.

**SMF RECORDS**

SMF records can be used to provide information on mainframe performance, not just TCP/IP. With OS/390, type 118 SMF records provide data on TCP initiation, TCP termination, ftp servers, ftp clients, and statistics. z/OS has type 119 SMF records, which provides data on TCP initiation, TCP termination, ftp servers, ftp clients, statistics, servers, and interface. It also includes round-trip times and retransmissions for each TCP connection in the TCP termination record. Software can then pick up the information from the SMF records. The good news is that SMF exit-based monitoring forces the collection of SMF records from TCP/IP. The downside is that if the information is not needed there is an unnecessary system overhead with the associated performance implication.

**SNMP**

Most network hardware and software vendors implement SNMP (Simple Network Management Protocol) Management Information Bases (MIBs) or diagnostic databases in their products. Various products provide generic access to SNMP MIBs. Usually a vendor provides a product to access its own MIBs. Monitors accessing the data held in the MIBs use the UDP-based SNMP protocol.

TCP/IP stack errors can often be found by using SNMP. The z/OS MIB (1.2, 1.3, and 1.4) has hundreds of variables. The major sections in the z/OS MIB are:

- SNMP public MIBs
- z/OS extensions
- OSA/ATM
- OMPROUTE
- SLA subagent

It makes use of the OSNMP and SNMPQE address spaces. To get information from other platforms, it makes use of agents. Both these things have a negative impact on performance.

SNMP would seem like the ideal way to monitor IP-related data. However, the information is retrieved by polling. This means there is always a delay between the information being stored in the MIB and being available to the monitoring software. The second problem is that every data item in the MIB has its own unique address or textual name. This has to be supplied by the monitor in each GET instruction. As each GET request requires one UDP datagram to be sent to the SNMP port (161) on z/OS, and the response from the subagent is another UDP datagram, this adds to the load on the network.

**TRACES**

Packet traces can be very useful in identifying where a problem is. It works by allowing a monitor product to be aware of all the IP packets flowing through the TCP/IP stack (and this includes UDP, ICMP, and OSPF packets). This interface is event driven. The monitor builds a complete picture of all IP-related activity, in real-time, by analyzing the packet headers, together with the maintenance of statistics based on packet counts and lengths.

There are two big drawbacks with this way of monitoring. First, the amount of CPU time, paging rates, and virtual storage required can be large—and increases whenever the network traffic increases. Second, there can be a problem with selecting what data will be monitored.

**TCP/IP MONITORING SOFTWARE**

What’s needed is software that gives users real-time information about problems as they are developing, and historical information for identifying trends. An easy to understand picture with drill-down ability would also be nice. More over, the product should not duplicate information already on the system or consume too much valuable system resource acquiring and presenting the information. And that’s what most of the third-party software does.
There are products such as Dave’s freeware TCP/IP monitor, available from http://www.planetmvs.com/freeware/tcpipm.html, to more expensive and ambitious products such as Impalex from William Data Systems (http://www.willdata.com/v2/products/implex.htm), VIP from SDS (http://www.sdsusa.com), MainView for IP from BMC (http://www.bmc.com/products/proddocview/0,2832,19052_19429_22915_1874,00.html), and, of course, IBM Tivoli OMEGAMON XE for Mainframe Networks (http://www-306.ibm.com/software/tivoli/products/omegamon-xe-mainframe) (and many others not listed here).

These are all able to monitor the network effectively, but, apart from products like Dave’s, they have a financial cost.

CONCLUSION

There are a variety of ways to find out what is happening on a network. Each has plus points and each has drawbacks. For larger networks, a monitor is going to have to be installed and this can help identify where problems are occurring before the end users start to notice.

Trevor Eddolls is CEO of iTech-Ed, a UK-based company that specializes in consultancy and writing about mainframes and networking.

E-mail: trevor@itech-ed.com. Web site: www.itech-ed.com