42

TCP/SNA update

June 2001

In this issue

TCP/SNA Update

Published by

Xephon
27-35 London Road
Newbury
Berkshire RG14 1JL
England
Telephone: 01635 38342
From USA: 01144 1635 38342
E-mail: fionah@xephon.com

North American office

Xephon
Post Office Box 350100
Westminster, CO 80035-0100
USA
Telephone: (303) 410-9344

Subscriptions and back-issues

A year’s subscription to *TCP/SNA Update*, comprising four quarterly issues, costs £130.00 in the UK; $190.00 in the USA and Canada; £136.00 in Europe; £142.00 in Australasia and Japan; and £140.50 elsewhere. In all cases the price includes postage. Individual issues, starting with the March 1991 issue, are available separately to subscribers for £33.00 ($48.00) each including postage.

Editor

Fiona Hewitt

Disclaimer

Readers are cautioned that, although the information in this journal is presented in good faith, neither Xephon nor the organizations or individuals that supplied information in this journal give any warranty or make any representations as to the accuracy of the material it contains. Neither Xephon nor the contributing organizations or individuals accept any liability of any kind howsoever arising out of the use of such material. Readers should satisfy themselves as to the correctness and relevance to their circumstances of all advice, information, code, JCL, EXECs, and other contents of this journal before using it.

Contributions

When Xephon is given copyright, articles published in *TCP/SNA Update* are paid for at £170 ($260) per 1000 words for original material. To find out more about contributing an article, please download a copy of our *Notes for Contributors* from www.xephon.com/contnote.html.

TCP/SNA Update on-line

Code from *TCP/SNA Update*, and complete issues in Acrobat PDF format, can be downloaded from http://www.xephon.com/tcpsnaupdate.html; you will need to supply a word from the printed issue.

Printed in England.

A look at VTAM and TCP/IP support – CS/390 V2R5 onwards

Even after 32 years with IBM, including 15 years in the networking business, I'm still amazed by the disparity in administrator networking skills. However, you need only monitor an IBM list server to see the camaraderie within this profession. (You too can receive this help. Send an e-mail with SUBSCRIBE TCP-L 'your name' in the subject matter to LISTSERV@VM.MARIST.EDU). Often, a networking novice will ask what level of TCP/IP or VTAM his company should be at. This article offers my assessment of what release provided what significant function.

In 1997, IBM packaged over 70 separate products into a single System/390 operating system. TCP/IP, VTAM, and multiprotocol support (AnyNet) together became an OS/390 element called Communications Server for OS/390 (CS/390).

CS/390 V2R5

CS/390 V2R5 (1998) is the base on which all subsequent releases have been built. V2R5 was a very ambitious release that included a new TCP/IP stack and many new features. It was also beset with code quality problems, which mostly disappeared in subsequent releases:

- The TCP/IP stack was completely rewritten to leverage the MVS architecture to improve performance and increase scalability. The previous TCP/IP protocol stack, ported from VM, ran as a single MVS task and had disappointing performance.
- An earlier tn3270 server application did not address several functions required to make it a viable migration solution. It was rewritten to conform to the tn327E specification (RFC 2355), which added emulation of 328*x* printers – both LU 1 and LU 3.
- High Performance Data Transfer (HPDT), which reduced the number of internal data moves, was introduced. All applications receive somewhat higher throughputs and use less CPU, but applications must be rewritten to the HPDT interface to gain significant performance improvement.

- Multi-Node Persistent Sessions (MNPS) became available to SNA applications using Advanced Program to Program Communication (APPC) for MVS. Although both network-node and end-node failures are recoverable without session loss, few customers have implemented MNPS within their networks.
- Virtual IP Addressing (VIPA) was extended to work for inbound and outbound data flows. VIPA allows the real IP address assigned as network endpoints in the System/390 to be associated with a pseudo, or virtual, address. If a network connection fails, traffic can be automatically and transparently routed over an alternative connection with the same virtual address.

CS/390 V2R6

CS/390 V2R6 is the earliest Version 2 release that still has service support, but this will be discontinued on 31 March 2002. If you have the tape available, you might want to install it. Besides improving V2R5 quality, additional TCP/IP performance gains made it the first release that provided competitive throughput:

- Enterprise Extender (EE) became available as an alternative to the popular Data Link Switching (DLSw). Like DLSw, EE enables enterprises to converge onto a single IP network. SNA clients and applications communicate over the TCP/IP network while maintaining ‘SNA-like’ benefits – congestion control, class of service, and transmission prioritization.
- tn3270 server application was enabled for Secure Sockets Layer (SSL) communications to an SSL-enabled client. SSL security ensures data privacy through encryption, message integrity, and ensuring that clients communicate only with authorized servers – critical requirements for commerce over public networks.

CS/390 V2R7

The V2R7 release is no longer available. With this release:

- A Cache Accelerator function that stored static Web pages in a cache within the TCP/IP stack to speed up WebSphere server processing was created.

- Support for a Gigabit Ethernet adapter called OSA-Express (available on Generation 5 and higher System/390 Servers) became available. Full duplex transmission capability known as Queued Direct Input/Output provides high-throughput direct access to System/390 memory.

- A Service Policy Agent enables a network administrator to set the Differentiated Services bits to control IP data packet flow. Use by queuing and discard algorithms ensures proper IP prioritization. Set-up uses a flat configuration file or an LDAP directory.

- Virtual Private Network (VPN) authentication and encryption capability (a V2R5 capability) was updated to the then current standards for Internet Protocol Security (IPSec). Set-up is done using a flat configuration file.

CS/390 V2R8

CS/390 V2R8 introduced some significant functions that are mostly unknown and unused. Until IBM addresses the need for a policy director to simplify the definition and management of VPN and policy functions, they will not get widely deployed. They include:

- Internet Key Exchange (IKE), an IETF-endorsed key and security association management protocol for IPSec, works in conjunction with Security Server (a priced OS/390 feature) to automatically create and distribute encryption keys.

- Unauthorized access to System/390 SNA applications from TCP/IP clients was made more difficult by the addition of SSL client authentication and use of RACF’s certificate registration capability.

- Service Policy Enhancements improved the management of network performance. Service policies can be dynamically updated to meet Service Level Agreements, without impacting network availability. A Resource Reservation Protocol (RSVP) agent can invoke reservation services, reserve bandwidth, and classify reservations.

- VIPA became dynamic, allowing automatic movement to an available alternative image in the sysplex.

- tn3270 Fast Reconnect enables client reconnect to the tn3270 server following a connection outage, without the client having to wait for the server time out.

SUMMARY OF V2R5-V2R8

V2R8, available in September 1999, was the last CS release to come out on a six-month release cycle – there was no new CS function in OS/390 V2R9. Long test cycles made it difficult to create significant new functions, and CS/390 adopted a yearly release schedule. I address what's new in V2R10 and 'R12' below. Before we move on, however, there are just two final points to make:

- First, migrating to later releases will significantly increase the functions available to you. These functions provide higher performance, and increase the availability of System/390. Later releases also have better quality.
- IBM has signalled that it intends to reduce the number of products and releases it will support. You will be forced to keep up with the new releases in order to receive service support.

V2R10 NIRVANA? (THE LAST CS/390 RELEASE)

The versions of CS/390 reviewed above (V2R5–V2R8) were the basis for the last Communications Server for OS/390 release,V2R10. Combining and building on availability features, V2R10 is the last release before the reversion of System/390, which created a name change to z/OS Communications Server (z/CS). Because of the hardware platform reversion to e-server, IBM has stated that V2R10 will be available for purchase through March 2002.

I mentioned above that CS development was on a yearly release cycle, and this is not about to change. This means that z/OS V1R1 has no new Communications Server function other than that available in V2R10. The next release, V1R2, has been previewed for October 2001.

IBM signalled, via a Statement of Direction (201-044), that future z/CS releases would support IPv6. Read this as no significant new IPv4 function for the next several releases – IBM spent over six years developing CS/390, CS/390 has many millions of lines of code, and the restructure to accommodate the IPv6 architecture will not be a quick (or easy) task. Therefore, unless you are about to migrate your existing applications to IPv6, what you see is what you get – for the next several years.

(Editor's note: For an in-depth review of the functions referred to below, and of VIPA in particular, see 'Increasing the availability of IP-centric mainframes', pp 10-21)

SYSPLEX DISTRIBUTOR AND NON-DISRUPTIVE WORKLOAD MOVEMENT

Workload can be distributed throughout a parallel sysplex without requiring an outboard distributor (eg Cisco Systems MultiNode Load Balancing (MNLB)). V2R10 combines XCF and VIPA functions to provide workload balancing.

Note that, unlike outboard distributors, all traffic must be forwarded by the server instance providing the Sysplex Distributor function. Collaboration between Cisco and IBM provides a V1R2 solution to this limitation. Sysplex Distributor replaces the Service Manager in the Cisco MNLB solution. (This enables Cisco to concentrate on packet forwarding and dump the software it obtained from Computer Associates/Sterling/Interlink.) Sysplex Distributor selects the server for new connection requests and provides this information to the Cisco MNLB agent for subsequent direct packet forwarding.

V2R10 supports planned workload movement between servers without disruption to established connections.

SERVICE POLICY AGENT

IBM continues to invest in the Service Policy Agent that was introduced in V2R8, undaunted by the lack of a policy director that makes administering policy practical. V2R10 enhancements include: traffic shaping to reduce low priority connections to 'best effort' when high-priority connections are not meeting QoS objectives set by Service Level Agreements (SLAs), Secure Socket Layer (SSL) encryption to protect from unauthorized entry through the LDAP interface, and the posting of a server QoS for use by Sysplex Distributor. V1R2 provides additional control options, including server SLA performance information for use by Sysplex Distributor, and VLAN priority tagging of outbound frames to provide user-level priority.

TELNET SERVER

V2R10 removes a limitation that favoured the outboard tn3270 server solution. Auto-logon support by the tn3270 server provides automatic session set-up when resource becomes available – in the past, users had to request the session if it was not available at initial log-on. (Expect this fix to be made available in V2R8.)

V2R10's use of system SSL for encryption key management improves tn3270e performance, scales linearly for multiple CPUs, leverages the common key ring support with RACF, and supports query of the vault registry certificate revocation list (to determine whether the client is still authorized). Telnet protocols are used (draft RFC) to negotiate whether the connection will be SSL protected – improving tn3270e security flexibility. Note that there was a noticeable lack of TLS support for tn3270e in the V1R2 preview information.

V1R2 provides Express Logon support for host-based tn3270 SSL clients. PKI-based identification and authentication removes the need for user IDs and passwords. Used in conjunction with IBM Host-On-Demand Version 5, the user is spared the inputting of ID and password for multiple applications. Currently V2R10 supports Express Logon for outboard tn3270 servers only, but expect the V2R10 host-based tn3270 server to also receive this support.

SECURITY

On-demand security associations in V2R10 enable VPN tunnels to be established dynamically – significantly reducing the administrative burden of establishing and allocating resources to the IPSec encrypted tunnels. This support requires a component of the OS/390 Security Server – a priced feature. In the V1R2 reversion there is the opportunity to eliminate this additional cost hindrance to providing VPN support.

A V2R10 traffic management daemon works in conjunction with the Service Policy agent to control the number of inbound connections, either by application port or client (eg IP subnet). V1R2 extends this concept by introducing host-based Intrusion Detection Services (IDS). Host-based IDS provides the opportunity to analyse encrypted traffic, and look at traffic real time and with lower overhead. Unfortunately, like the Policy Agent, the management and reporting tools to make implementation viable are not yet available for this function.

EXPECTED ENHANCEMENTS

Both releases provide the expected accommodation of requests for FTP enhancements (including TLS support in V1R2), improve usability, increase performance, and support the new server hardware. Additionally both releases continue to improve their support for Enterprise Extender – more about that in a future article.

SUMMARY

V2R10, the last Communications Server for OS/390 release, ties many of the previously separate features together to provide a solid release. Expect IBM to add functions from follow-on releases into V2R10 and further extend the purchase window beyond March 2002.

Richard Tobacco
Independent Consultant (USA)

Increasing the availability of IP-centric mainframes

By 2003, it will no longer make sense to boast about the legendary 99.99% availability of mainframe-hosted SNA networks. Although SNA mission-critical applications will still be running on mainframes, most mainframes, and the networks connected to them, will by then be increasingly IP-centric. So, the challenge facing us now is to ensure that these IP systems can at least approach, if not match, the 'non-stop' availability to which mainframe users are accustomed.

The good news is that, since 1996, IBM has been working on a raft of IP-related high-availability features which, when used in conjunction with the 'single system image' clustering capability of parallel sysplex, ensure that you can already have an IP-based mainframe system with less than ten minutes outage per year – even with 24 x 365 operation. Compare this with the 20 hours of downtime a year which is still the norm for a typical 'non-mainframe' Unix system.

The primary IP-related high-availability features that have been systematically introduced as OS/390 (and now z/OS) has evolved over the last six years are as follows:

1 *Virtual IP Addresses (VIPA)* – introduced in 1996 for TCP/IP for MVS and included in OS/390 Version 2 Release 5.

2 *Workload Manager (WLM) and Domain Name Server (DNS) integration* – OS/390 Version 2 Release 5.

3 *Sysplex Awareness* – OS/390 Version 2 Release 7.

4 *VIPA Takeover and Dynamic VIPA working in conjunction with Sysplex Distributor* – OS/390 Version 2 Release 8.

5 *Application-initiated Dynamic VIPA* – OS/390 Version 2 Release 8.

6 *VIPA Non-Disruptive Takeover* – OS/390 Version 2 Release 10.

7 System-Managed *Coupling Facility (CF) Structure Duplexing* – z/OS Version 1 Release 2.

8 Integrated *Intrusion Detection* for Securing Networks – z/OS Version 1 Release 2.

VIPA – THE BASIC BUILDING BLOCK

Even a quick glance at the above list shows that VIPA is a recurrent theme in IBM's OS/390 high-availability strategy for IP. VIPA is the fundamental building block used by IBM to ensure that mainframe IP systems, in time, will match the 'non-stop' operational capabilities of SNA systems.

VIPA, in essence, permits external entities to address an OS/390 IP resource using a virtual IP address rather than that IP resource's actual address. VIPAs can therefore be assigned to IP applications (eg tn3270(E) server, WebSphere's HTTP Web server, FTP application) or to IP stacks. OS/390, transparent to the user, will do a virtual-to-real address conversion – on-the-fly. VIPA thus ensures that clients are insulated from IP address changes at the mainframe caused by network connection or IP stack failures.

VIPA is thus pivotal in providing fault tolerance for IP applications – particularly in the event of network attachment failures (eg OSA-Express failure). Although some highly technical IBMers cringe at this analogy given that there are some small exceptions, think of VIPA as being the IP equivalent of SNA 'Generic Resources' which permit you to refer to an application by a 'family name' (eg CICS) rather than by its actual name (eg CICS1 or CICS2).

VIPA is thus an IP address which, though it is known to the network, is not actually associated with any particular IP adapter (eg OSA-Express). To external routers and other IP stacks, a VIPA appears to be just an address (or a subnet) reachable via an 'internal' physical attachment to the IP stack hosting that VIPA. Another way to think of this, especially if you're familiar with the nuances of IP routing, is that an IP stack that owns a VIPA appears to the rest of the routing network as just another router – and the VIPA is an address reachable via that router.

As with other IP addresses, the availability of VIPAs are routinely advertised around the network using either static or dynamic routing protocols. When an OS/390 IP stack receives an IP packet destined to a VIPA that it supports, it immediately recognizes the VIPA as a local address. It will then route that packet to a higher layer (eg TCP, UDP, ‘raw’ socket, etc) – just as with IP packets specifying real, physical addresses.

The main advantage of assigning a VIPA to an application running on an OS/390 mainframe with multiple network attachments, rather than using its actual physical address, is that a failure of any one network attachment will not disrupt connectivity to the application. As long as there’s an active network path between the client and the IP stack hosting the VIPA, OS/390 will ensure non-stop continuous operation between the client and the application – even if there are outages to one or more of the network attachments. When there’s a network attachment failure pertaining to a VIPA, the routers in the network simply re-route around that failure.

As is the case with any IP address, only one IP stack may own any particular VIPA at any given time – at least from the perspective of the routing network. So, if there’s an outage to the IP stack owning the VIPA or the system on which that stack is running, the VIPA becomes temporarily unavailable. You can, however, easily get around such a VIPA outage by moving the VIPA to another active IP stack. This can be very easily achieved, since a VIPA, by definition, is never associated with a physical network interface.

Thus a VIPA can be moved either manually by an operator, or automatically using an automation procedure to another IP stack hosting another instance of the application associated with the VIPA. Clients that use the subject VIPA will now get routed to this other instance of that application – as opposed to the application instance that is currently unavailable. Dynamic VIPA addresses and VIPA Takeover make this process even simpler, more transparent, and robust.

VIPA TAKEOVER (DYNAMIC VIPA) – AUTOMATING IP RECOVERY

VIPA Takeover (also known as Dynamic VIPA) significantly bolsters IP application availability in System/390 parallel sysplex environments. If a stack or system outage impacts an application instance with a VIPA, that VIPA can be quickly, dynamically, and automatically moved to an alternative available application image within the sysplex cluster. To expedite connection reestablishment, the new application instance will send ‘connection resets’ to all the affected clients so that they don’t have to wait for a connection time-out to occur to realize that they need to reconnect to the application. This ‘connection reset’-based rapid reactivation process can reduce application reconnect times by as much as 60%. Furthermore, the restarting of the failed IP stack or application instance can be fully automated through the use of the OS/390 Automatic Restart Manager (ARM).

VIPA Takeover permits IP software to deal with the automatic movement of VIPAs within a parallel sysplex with just a minimum level of configuration information. As of OS/390 Version 2 Release 8, IP applications can readily activate dynamic VIPAs, either on a continuous basis or on demand, using simplified configuration definitions. Other IP stacks within the parallel sysplex cluster can also be configured to act as automatic back-ups for continuously-active dynamic VIPAs. In such a hot stand-by back-up configuration, the dynamic VIPA will be automatically and nearly instantaneously activated on a back-up stack if there is any type of outage to the original application instance associated with that Dynamic VIPA. This automatic back-up scenario is referred to as VIPA Takeover.

Starting with OS/390 Version 2 Release 7 (1999), IP stacks in OS/390 began to gainfully exploit parallel sysplex functionality to gain awareness of other IP stacks within the cluster – and exchange information with these other stacks. This stack-to-stack, real-time communication is realized using the Cross System Coupling Facility (XCF) messaging. XCF is a fundamental and long-standing sysplex enabling and empowering technology that appeared with MVS/ESA SP4 in the early 1990s. XCF lets OS/390 applications, in this case IP stacks and IP applications, communicate peer-to-peer across a parallel sysplex cluster using ESCON channels, 800Mbps HiPerLinks, or internal links. XCF is a basic OS/390 facility that works independently of conventional network protocols.

IP stacks exchange the IP addresses they are currently supporting with other stacks within the sysplex, using XCF messaging on an on-going basis. Consequently, all stacks are continually aware of all the active IP addresses within the sysplex – including dynamic VIPAs. An IP stack can therefore also tell when an IP entity it is dealing with (eg a client) is also interacting with IP addresses or VIPAs supported by another stack within the sysplex. If new IP stacks are added to the sysplex, this information is also dynamically conveyed to the other stacks. This inter-stack address information is automatically updated whenever an IP address is deleted from or added to a stack. There is also rapid notification if a stack is halted or suffers an outage so that the other stacks can expunge the now inactive addresses. This XCF-based Sysplex Awareness plays a key role in facilitating VIPA Takeover. With z/OS, IBM intends to further improve this capability by 4Q01 through a new feature known as 'System Managed CF Structure Duplexing'.

SYSTEM MANAGED COUPLING FACILITY STRUCTURE DUPLEXING

The logical and physical connections between the mainframes and processors exploited by XCF are realized via what IBM refers to as the 'Coupling Facility' (CF). The CF is the very heart and soul of the parallel sysplex technology and the basis for the read/write data-sharing that is imperative in order to achieve the single system image and MNPS characteristics. It is the CF that provides parallel sysplex systems, whether running IP or SNA, with three key and prerequisite functions:

1 Very fine granularity data locking to permit data sharing and data updating between multiple copies of an application, without contention and loss of integrity.

2 Data caching, as in the case of I/O caching, on both a local and distributed basis, including high-performance data caching for the shared data access.

3 Queuing mechanisms to support workload distribution, message interchange, and state information sharing.

The CF *per se* consists of hardware and specialized IBM-licensed internal code ('microcode') known as 'CF Control Code' (CFCC). IBM currently provides two distinct means of implementing a CF: an integrated System/390 solution and one that requires a separate, stand-alone box. The integrated option is known as the Internal

Coupling facility (ICF), while the stand-alone CF is implemented using either an IBM 9674 box or an IBM 9672 Model R06 machine.

With ICF, you use one (or more) of the processors in a multiprocessor System/390 to act as the CF and run the requisite CFCC software. The processor(s) used to run CFCC can be dedicated just for the CF function. IBM also provides the option whereby the CFCC code can run in shared-mode with processors being used to support a true application partition (ie an LPAR). The 9674 and 9672-R06 have their own processors to run the CFCC software. Both these boxes can have up to 10 processors. The 9672-R06, though constrained in that it can only act as a CF box, is in reality just another System/390 G5 mainframe, with one to ten standard System/390 processors.

In many implementations, modified data (ie cached data) remains in the coupling facility for a short period of time. This could compromise total recovery with no data loss whatsoever in the event of a CF failure. Other systems require manual procedures and sysplex-wide log merge processes in order to recover their CF structure data when trying to recover from a failure. System-Managed CF Structure Duplexing, which will be available in 4Q01, will provide an elegant solution to these potential setbacks by enabling the CF data structures to be automatically duplexed. The robust failure recovery capability of duplexing will be achieved by creating a duplicate copy of the CF structure in advance of any failure, and then maintaining the two structure instances in a synchronized duplexed state during normal operation.

The System-Managed CF Structure Duplexing capability is a combination of z/OS support and Coupling Facility Control Code (CFCC) Level 10 LIC support on zSeries servers. Rollback of the processor CFCC Level 10 LIC functions to G5/G6 mainframes and R06 coupling facilities will also be provided in the z/OS Version 1Release 2 timeframe. This function will further enhance mainframe availability by essentially eliminating a small but potential risk to parallel sysplex integrity.

APPLICATION INITIATED DYNAMIC VIPA

VIPA Takeover can be put into play whenever there are multiple instances of the same application, each of which can appropriately respond to any client request. However, there can be situations where some applications need to have a particular IP address (or VIPA) associated with a particular instance of that application, so that client requests to that IP address always go to the same application instance. OS/390 supports this type of scenario, where there is a need for client-to-application correlation, with Application-Initiated Dynamic VIPAs. Application-initiated Dynamic VIPA ensures that clients always reconnect to the same application instance, even if the application is now running on a different OS/390 image.

Thus, there are two distinct scenarios in which VIPA Takeover can be profitably used to ensure the continuous operation of IP applications. These two scenarios are:

1 Situations in which the IP clients do not need to be serviced by a particular instance of an application. Web serving is an example of this type of scenario.

2 Situations, as discussed above, in which there is a defined client-to-application instance relationship that dictates that IP clients must connect to a specific instance of an application. CICS, for one, can be configured for this type of operation, in which clients connect only to a specific instance of CICS, rather than to any CICS instance. These are the situations handled by Application-Initiated Dynamic VIPAs.

Take the situation where any application instance can serve any client. In this scenario, you can have multiple, identical instances of an application running on separate OS/390 images within the parallel sysplex cluster. Any of these application instances can readily service any request from that application’s client base. Consequently, clients can connect to any of the application instances. It really doesn’t matter which application instance serves which client. Web serving, such as with OS/390-resident WebSphere, is a good example of this type of ‘application instance independent’ application. Because all instances of a Web server can have access to the same set of static and dynamic Web pages, it doesn’t really matter to which Web server instance a user connects. All the Web server instances can provide the user base with the same set of information. In this type of scenario, each application instance accepts client requests addressed to any of the local IP addresses. (The technical term for this is binding to INADDR_ANY.)

In this type of scenario, a Dynamic VIPA is assigned on behalf of the application on each IP stack hosting an instance of that application. The stack serving the application instance is designated as the primary owner of this Dynamic VIPA. (This is achieved using the VIPADEFINE configuration statement.) Other stacks serving different instances of the application are configured as back-up VIPAs for a given primary VIPA. (This is achieved using the VIPABACKUP configuration statement.) Thus, a given IP stack will be the primary VIPA owner for an application instance it is hosting, while being a back-up for other instances of the application that are being hosted by other stacks. There are no hard and fast rules about how the back-ups need to be configured. For a start, there's no requirement that each stack has to act as a back-up for all VIPAs. Systems programmers have total latitude in determining how they want to configure back-ups and which stacks back up which VIPAs. XCF messaging is used by the stack to automatically exchange information about the Dynamic VIPAs it is currently supporting.

In the event of an outage affecting a stack hosting an application instance, one of the back-up stacks will automatically take over the impacted dynamic VIPA from the failed stack. The application instance outage will disrupt the client connections that were being supported by that instance. The clients, however, will attempt to recover their connections by attempting to connect to the same IP address – which in this case will be a dynamic VIPA. Since the required dynamic VIPA is now being hosted by a different stack, the re-connection request (as well as any new connection requests) will be sent to this new stack. Dynamic routing protocols (such as RIP or OSPF) within the routed network will recognize that the dynamic VIPA has moved and will take care of the necessary rerouting to the new stack.

When a dynamic VIPA is taken over, 'Connection Resets' will be sent to clients attempting to send data to a moved VIPA to expedite reconnections. In the case of the tn3270(E) Server within Communications Server for OS/390, using this 'Connection Reset' mechanism can expedite connection reestablishment by as much as

60%. This has been repeatedly confirmed by various performance studies. Other applications should enjoy similar, if not better, results.

The failure of an IP stack hosting dynamic VIPAs is automatically detected by the other stacks via XCF. The back-up stack that will take over the workload of the failed application instance is determined by the surviving stacks by weighing current workloads against pre-defined configuration data. The stack that best meets the WLM criteria will be designated as the new primary for that dynamic VIPA. If a failed stack was supporting multiple different application instances, each with its own dynamic VIPA, the take-over of these dynamic VIPAs may be distributed across multiple stacks to ensure that the workload is balanced across multiple OS/390 images during the recovery phase.

Now let’s look at the other scenario, in which clients have to interact with a specific instance of an application, as opposed to being able to connect to any instance of that application. This means that the IP address of an application instance must remain constant and also be different from the IP addresses being used by other application instances. Consequently, the IP address has to move with the application instance. Such a move can be realized only using dynamic VIPA.

Let’s look at an example to reinforce the issues involved here. You could elect to assign a unique IP address to a CICS instance so that this CICS instance services only requests that explicitly specify that address. In this way, you could have multiple CICS instances, even within the same LPAR, each serving a pre-specified set of clients, without there ever being a danger of client requests ending up at the wrong CICS instance. Should a CICS client, in this type of configuration, experience a connection failure during a transaction, that client would need to reconnect to the same CICS instance. It would not do to have the CICS name resolved to the IP address of a different CICS instance.

To make matters more complicated, most enterprises invoke the Automatic Restart Manager (ARM) capability when they deploy CICS. Consequently, if a CICS image fails, or the OS/390 image hosting it suffers an outage, ARM will automatically restart that CICS image on a surviving OS/390 image. The OS/390 image that will be chosen by ARM will depend on current workload criteria as determined by WLM, as well as pre-defined restart preferences. The bottom line here is that it is therefore nearly impossible to predict, in advance, the OS/390 image on which the subject CICS instance will be restarted. Thus, trying to assign IP addresses ahead of time is not feasible. The only practical way to get around this is to enable the application instance to tell its hosting IP stack which Dynamic VIPA it would like to activate at any given time. So in this case, when the CICS instance is restarted on another stack, the CICS instance will ask the new stack to activate the required VIPA. This will happen, as long as that VIPA is not already active somewhere else within the parallel sysplex cluster.

A somewhat ironic issue that comes up with any VIPA Takeover scenario is what should happen when the IP stack that was originally handling the application instance is restored. Non-disruptive VIPA Movement, which was introduced with OS/390 Version 2 Release 10, provides a way to overcome this challenge.

When a previously failed application instance is restored, it is obviously highly desirable to move the impacted (but now recovered) workload back to this 'home' application instance in order to balance workloads and ensure maximum processing efficiencies. However, you could not normally do this without disrupting the on-going connections to the back-up instance. Neither can you afford to wait until there are no longer any active connections to the back-up instance since this might never happen given that there may also be also new connections being established. This is where Non-disruptive VIPA Movement steps in.

With Non-disruptive VIPA Movement, the Dynamic VIPA that was moved to a back-up stack is automatically given back to the original 'home' stack as soon as that stack and the subject application instance are restored. The back-up stack immediately notifies the restored stack of all the active connections that it is currently supporting using XCF messaging.

The restored stack will not interfere with these currently active connections. These will continue to be handled by what was the back-up stack. However, all new connections to the Dynamic VIPA in question will now go to and be serviced by the restored stack. The restored stack, furthermore, will forward all requests related to the previously active connections to what was the back-up stack. The end result of this is that all new connections go to the restored stack, while what was the back-up stack continues to handle previously active connections until the connections are terminated.

Non-disruptive VIPA Movement works with all VIPA Takeover scenarios, including Application-Initiated Dynamic VIPA situations. Non-disruptive VIPA Movement is also a very powerful tool if you want to move an application instance to a different OS/390 image within the parallel cluster (eg a new processor).

INTRUSION DETECTION

A major source of instability and downtime for IP systems, especially of late, has been hacker attacks on them, such as the widely publicized 'denial of service' attacks. Firewalls can provide a level of protection against hackers and these attacks. However, they cannot provide protection when the attack may come from within the enterprise or when the hackers employ the availability of end-to-end encryption into the mainframe to get inside the system.

The new mainframe-based Intrusion Detection Services (IDS) provided in z/OS Version 1 Release 2 will augment network-based IDS sensors and scanners. This IBM capability can discard attacking packets before they cause damage, discard packets exceeding established thresholds, and limit the number of connections from greedy users (a symptom of denial of service attacks). IDS will also provide event recording and reporting, including stand-alone reporting of IDS events (attacks) to console and Syslog, a new specialized IDS packet trace for off-line analysis, and statistics-gathering baseline and exception reporting.

THE BOTTOM LINE

Thanks to all these new OS/390 and z/OS Version 1 Release 2 features, it's now possible for IP-centric mainframe systems to offer unprecedented continuous availability. Features such as Dynamic VIPA and non-disruptive VIPA movement facilitate smooth recovery in the event of a failure, and also ensure that IP applications can be freely moved within a parallel sysplex without disrupting any existing connections or halting new connections.

Anura Gurugé
Strategic Consultant (USA)

Contributing to *TCP/SNA Update*

In addition to *TCP/SNA Update*, the Xephon family of *Update* publications now includes *CICS Update*, *MVS Update*, *VSAM Update*, *DB2 Update*, *RACF Update*, *AIX Update*, *Domino Update*, *MQ Update*, *NT Update*, *Oracle Update*, and *TSO/ISPF Update*. Although the articles published are of a very high standard, the vast majority are not written by professional writers, and we rely heavily on our readers themselves taking the time and trouble to share their experiences with others. Many have discovered that writing an article is not the daunting task that it might appear to be at first glance.

They have found that the effort needed to pass on valuable information to others is more than offset by our generous terms and conditions and the recognition they gain from their fellow professionals. Often, just a few hundred words are sufficient to describe a problem and the steps taken to solve it.

If you have ever experienced any difficulties, or made an interesting discovery, you could receive a cash payment, a free subscription to any of our *Updates*, or a credit against any of Xephon's wide range of products and services, simply by telling us all about it. For a copy of our *Notes for Contributors*, which explains the terms and conditions under which we publish articles, please write to the editor, Fiona Hewitt, at any of the addresses shown on page 2, or e-mail her at fionah@xephon.com

SNA over TCP/IP

This article takes a look at the AnyNet product family, which enables application programs to use different network transport protocols across interconnected networks for communication purposes. The main purpose of this is to reduce the number of transport networks and reduce operational complexity. These benefits can be obtained without changing or purchasing additional hardware or application programs.

There are various functions that in essence ‘are’ the AnyNet suite of products:

- *AnyNet APPC over TCP/IP* is available on MVS, AIX, Windows, and OS/2 platforms, and is used to enable APPC or CPI-C application programs to communicate with other APPC or CPI-C programs using TCP/IP networks. It also supports independent logical units using LU 6.2 protocol.
- *AnyNet SNA over TCP/IP* is available on MVS, OS/2, and OS/400 systems, and enables an SNA program to use a TCP/IP network. As well as supporting LU 6.2, it provides support for dependent logical unit communication with traditional printer and terminal emulation products. The latter support requires host definition as a dependent logical unit server (DLUS). At the workstation end, VTAM-provided OS/2 dependent logical unit requester (DLUR) support is enabled.
- *AnyNet SNA over TCP/IP Gateway* is available in the MVS and OS/2 environments, and connects an IP and an SNA network, thus enabling communication between these network types. Using this feature, SNA applications that run non-natively on an IP network can communicate with SNA applications on an SNA network without any changes being required to the application. If the communication involves independent logical unit communications, two or more such gateways can be used to connect multiple IP and SNA networks. This then allows applications to span networks.
- *AnyNet Sockets over SNA* is available in MVS, AIX, OS/2, and OS/400, and its purpose is to provide application programs that

use C socket interfaces to communicate over SNA networks. The product uses LU 6.2 conversations to achieve the communication.

- *AnyNet Sockets over SNA Gateway* is available only in the OS/2 operating system, and connects IP and SNA networks to enable communication between socket applications. Socket applications that exist on TCP/IP networks can communicate with socket applications on SNA networks without change. The SNA networks must have AnyNet Sockets for MVS, AnyNet Sockets over SNA for OS/2, AnyNet Sockets over SNA for OS/400, or AnyNet Sockets over SNA for AIX installed on them.

HOW SNA OVER TCP/IP WORKS

Applications that use SNA protocols can benefit from SNA over TCP/IP to send and receive information across an IP network using the MPTN architecture. CICS and IMS applications can access an IP network without any modification. This is achieved by using protocols that bypass the lower transport layers of the SNA architecture and instead encapsulate an SNA path information unit in a TCP frame. The basic information unit is built by SNA over TCP/IP for TCP when sending and receiving data.

This process of building the unique transmission frame is totally transparent to the SNA application program. A supported API is used, so that application programs pass data to the presentation services of VTAM or the OS/2 Communications Manager/2. The data is passed through the various SNA architectural layers and then presented to SNA over TCP/IP instead of the normal presentation to SNA path control. When a session is started for the application program, SNA over TCP/IP will translate the SNA routing detail into IP routing detail, and use IP to create a TCP connection to the relevant system. The configuration information defined by the systems programmer will enable SNA over TCP/IP to determine a number of factors:

- Whether to route the data using SNA transport or TCP/IP transport.

- The IP address associated with the relevant fully qualified logical unit name.

A TCP or UDP port will be designated for the exclusive use of SNA over TCP/IP, thus enabling TCP/IP to determine whether to route the transmission frame it receives to a TCP/IP application or an SNA over TCP/IP feature.

To facilitate SNA over TCP/IP, certain minimal levels of software are required by the different IBM operating systems. These are:

- IBM MVS/ESA SP Version 3.1.3 and SMPE 1.5
- IBM TCP/IP Version 2.2.1 for MVS
- LE for MVS and VM
- VTAM 4.3 Base
- VTAM 4.3 AnyNet Host Feature.

Note that, even if your SNA node does not use SNA over TCP/IP, if it needs to communicate via an intermediate node that does have SNA over TCP/IP then you must have at least VTAM Version 3.2.

SNA over TCP/IP uses the following support services provided by TCP/IP for MVS:

- The IUCV socket interface
- Socket API for C language programs
- TCP/IP protocol support
- HOSTS file or Domain Name System resolve function.

The product supports a number of application programs and subsystems that use the VTAM APPCCMD API, the MVS/APPC including CPI-C APIs using LU 6.2 for networking support, and the VTAM record application program interface (RAPI).

MANAGING AN SNA OVER TCP/IP NETWORK

A number of tools exist to facilitate the management of network communication for MVS hosts. The SNA over TCP/IP support allows IP devices to be addressed using LU names. Using the HOSTS file or the domain name server (DNS) database, a fully qualified LU name is added to a user-specified IP sub-domain name to form the full IP domain name. The IP name is then mapped to an IP address used to route network data. Because the IP domain name contains an LU name, TCP/IP commands such as ping and tracerte will work with LU names defined in the DNS or HOSTS file. This allows network connections to be tested using standard IP tools.

You can define a host as a central host for receiving SNA over TCP/IP alerts on problems in the network. You can also use NetView on the MVS host, NetView/6000, and the general NetView family of products to manage SNA over TCP/IP. NetView/6000 acts as a service point to the host NetView. NetView/6000 provides the following TCP/IP management functionality:

- Generic support for Management Information Base 2 (MIB2)
- Support for fault applications
- Support for performance applications
- Support for Trouble Ticket/6000 applications
- Support for TCP/IP routers and SNMP devices.

If the TCP/IP network connects to an SNA network, NetView and NetView/6000 will work together to do the following:

- Manage SNA using NetView.
- Manage TCP/IP using NetView/6000.
- Filter SNMP traps.
- Convert SNMP traps into NetView alerts.
- Customize alerts at NetView/6000.
- Use the RUN commands from NetView to NetView/6000 for any SNMP and TCP/IP commands.
- Access SNMP commands via an MVS/ESA TCP/IP server if IBM MVS TCP/IP is installed.
- Provide access for MVS/ESA TCP/IP commands through TSO.

PLANNING FOR ANYNET SYSTEMS

As with any system, planning is key to a successful implementation.

For each LU-LU session that will be established over an IP network, a VTAM program will create one TCP connection. Each connection and the processing that occurs for data transfers relating to those sessions will use the following TCP/IP control blocks:

- *ACBs*. Activity control blocks specified in the TCP/IP ACBPOOLSIZE configuration statement. It's quite feasible for 2–3 such control blocks to be used for each LU-LU session if data transfer activity is high.
- *Data buffers*. These are regular data buffers specified in the DATABUFFERPOOLSIZE configuration statement. These can be 1–2 in number per LU-LU session.
- *Envelopes*. These are small envelopes as specified on the TCP/IP ENVELOPEPOOLSIZE configuration statement. From 1–4 can be used per LU-LU session, depending on the level of SNA-expedited data transfer activity.
- *SCBs*. These are socket control blocks as specified on the TCP/IP SCBPOOLSIZE configuration statement. One such control block is used per LU-LU session.
- *SKCBs*. These are socket interface control blocks specified on the TCP/IP SKCBPOOLSIZE configuration statement. Again, one such control block is used per LU-LU session.
- *TCBs*. These are transmission control blocks specified on the TCP/IP TCBPOOLSIZE configuration statement. One transmission control block is used by each LU session.

When setting up the TCP/IP address space for MVS you must define suitable values for the configuration statements that relate to the above control blocks to account for the number of LU-LU sessions that will be activated across the IP network. You should also take into account normal TCP connections in this calculation.

VTAM will see the IP network as a major node. When this major node is activated, VTAM will start several MVS tasks to provide service for the major node. As sessions become active on the network VTAM will start additional tasks as required to service the TCP connections. These additional tasks are controlled by the TCB operand of the TCP/IP major nodes VBUILD statement. If a default of 10 is taken, VTAM will start fourteen MVS tasks to service the major node – it starts four automatically at major node activation. Each task uses one TCP/IP client control block. When the TCP/IP region is set up, you should take into account the need to balance the TCP/IP client control blocks among SNA over TCP/IP and other TCP/IP applications. The number of available client control blocks is determined by the TCP/IP CCBPOOLSIZE configuration statement.

CONFIGURING THE IP NETWORK TO VTAM

The two major configuration tasks in setting up the AnyNet SNA over TCP/IP are the definition of the TCP/IP major node and the cross-domain resource definitions.

The TCP/IP major node is used to define the TCP/IP network to VTAM. There can be more than one TCP/IP major node, but only one physical unit and line can exist in each major node in an active state. Obviously, the number of nodes to define depends on the network that's in place. The statements and associated information required to define the TCP/IP major node are shown in Figure 1.

Name	Definition Statement	Operands	Default	Restrictions
Name	VBUILD|	TYPE=TCP		
		CONTIMER	30	
		DGTIMER	30	
		DNSUFX	SNA.IBM.COM	
		EXTIMER	3	
		IATIMER	120	
		PORT	397	
		TCB	10	
		TCPIPJOB	TCPIP	
Name	GROUP	ISTATUS	ACTIVE	
		SPAN		
Name	LINE	ISTATUS	ACTIVE	
		SPAN		
Name	PU	ISTATUS	ACTIVE	
		NETID		
		SPAN		

Figure 1: TCP/IP major node definition keywords

The TCP/IP major node must be defined as a non-switched major node with no switched lines or logical units. It is valid to define more than one Group, Line, and PU in a single TCP/IP major node, but only one can be active at any one time.

Now, let’s look in more detail at what each of the options on the various statements is used for:

- The CONTIMER operand is used to specify the length of time in seconds that VTAM should wait for the MPTN connection to start after the TCP connection is activated. If the connection fails to be established within this time period, the session initiation will fail.
- The DGTIMER operand specifies the length of time between retries of sending a datagram for SNA-expedited data. This again is coded in seconds. The datagrams are continuously sent at regular intervals dictated by this operand’s setting. This will occur until a response is received from the remote system, or until five datagrams have been sent without response. Retries are classed as exceeded when five have been sent without a response. The action that the system will take depends on the type of datagram:
  - If it’s an SNA-expedited datagram, the session that data is being transferred on will be stopped.
  - If it’s a session termination request, the TCP connection that was being used by the SNA session is closed.
  - If it’s an MPTN keepalive datagram, all sessions using the IP address pair that the MPTN keepalive represents are stopped.
- The DNSUFX is the domain name suffix used when VTAM creates an IP domain name from an SNA LU name and SNA network identifier. This ensures that any SNA LU name and network identifier specified as luname.netid is distinct from any existing IP domain name. Certain conditions apply to the domain name coded. The domain name suffix must be less than or equal to 237 characters in length. The maximum allowable length for a TCP/IP fully qualified name is 255 characters; an LU name can be eight characters, as can the netid, and a period separates each – this results in the 237 limit. Each label in the suffix list must begin with a character, end with a digit or a character, contain only

characters, digits, or a hyphen, and be less than 63 characters in length. The domain that is created by VTAM is in the form:

Luname.netid.cccc.cccc.ccccc......

Here, Luname relates to the SNA LU name; netid is the network identifier for the current SNA domain; and cccc.cccc.ccccc.... is the domain name suffix. An example for an SNA LU name SNAJB1 in network GBJB01 with a domain suffix of sna.jb.com would be SNAJB1.GBJB01.sna.jb.com. The SNA fully qualified name would be GBJB01.SNAJB1.

- The EXTIMER statement specifies in seconds the length of time between sending expedited data over the TCP connection associated with an SNA session and sending the SNA expedited data using a datagram. When a response is received from the remote system as a datagram or on a TCP connection, the process of sending is deemed complete.

- IATIMER is used to code the length of time VTAM allows two IP addresses to remain inactive before sending another MPTN KEEPALIVE packet to test the connectivity between the two addresses. This again is coded in seconds.

- ISTATUS determines whether this minor node is to be activated after the first activation of the TCP/IP major node. This is similar to the standard VTAM ISTATUS operand used for VTAM resources.

- NETID is the SNA network identifier. Port specifies the TCP and UDP protocol port that VTAM will use to support SNA sessions over an IP network. All nodes that establish SNA sessions over an IP network must use the same PORT number if they are to communicate with each other. This can be any integer value between 1 and 65,535 and the default is 397. IBM recommends using the default as it is a recognized standard registered with the Internet Activities Board.

- TCB is coded to specify the number of MVS subtasks that can be used by VTAM to access the TCP/IP components. Each sub-task can support 120 sessions. The value for TCB can range from 1 to 99. If you don’t specify a value, this defaults to ten.

- The TCPIPJOB statement is used to code the name of the TCP/IP started task that runs on your MVS system. When you use SNA over TCP/IP, you need to be aware that VTAM acts as a TCP/IP application and therefore must know the main TCP/IP started tasks name.
- TYPE=TCP basically tells VTAM this is a TCP/IP major node. The SPAN statement defines the span of control for the VTAM minor node resources. NetView uses this chiefly for operating capabilities.

INDEPENDENT AND DEPENDENT LOGICAL UNIT DEFINITIONS WITH TCP/IP

SNA over TCP/IP supports both independent and dependent LUs. Several factors should be considered for the different types of LU.

If a remote independent LU in an IP network initiates a session with VTAM, then VTAM can define the LU dynamically as long as the DYNLU=YES operand is coded on the VTAM system start options in the ATCSTRxx member of SYS1.VTAMLST. In an IP network, remote independent LUs are defined as cross-domain resources. If VTAM initiates an LU-LU session to an LU over an IP network, you need to either dynamically define the cross-domain resource using the ALS selection function of the VTAM session management exit routine, or predefine the cross-domain resource to VTAM in a cross-domain resource major node. For cross-domain resources defined for independent LUs, you must code a name label that reflects the remote independent LU name. The physical unit defined in the TCP/IP major node is the adjacent link station for each session to the remote independent LU. Use the following to specify the physical unit:

- ALSLIST operand of the CDRSC definition statement
- ALS selection function of the session management exit routine.

DEFINING DEPENDENT LUs FOR SNA OVER TCP/IP

The dependent logical unit server (DLUS) function and the dependent logical unit requester (DLUR) enable dependent LU communication across networks. SNA over TCP/IP uses these two functions to support dependent LU communications across IP networks. Dependent LUs can communicate over the IP network as long as the host is defined as a dependent LU server and the workstation is enabled with VTAM provided OS/2 DLUR support. If you intend to use the DLUS-DLUR support you must take the following into account:

- VTAM has to be started as an interchange node (NODETYPE and HOSTSA are coded in the VTAM start options member ATCSTRxx in SYS1.VTAMLST) or as a Network Node (NODETYPE is coded).
- The DLUS must be adjacent to the TCP/IP network.
- The LU-LU session must flow through the DLUS node.

You must also adhere to the following guidelines when defining dependent LUs in an IP network to VTAM:

- Define the DLUR in the IP network as a VTAM cross-domain resource.
- A switched major node is used to define the dependent LUs.
- For outbound connections to the workstation, the DLURNAME operand on the PATH definition statement specifies the name of the cross-domain resource that represents the DLUR.

Finally, there are even more rules for the cross-domain resource definitions for the DLUR. These are:

- The cross-domain resource definition statements name label must be the control point name of the DLUR itself.
- The physical unit that's defined within the major node for TCP/IP will be the adjacent link station for each session established between the DLUR and any of its supported dependent LUs.

CONSIDERATIONS FOR USING SNA OVER TCP/IP WITH LUs IN DIFFERENT SNA NETWORKS

SNA over TCP/IP provides cross-network support to allow LUs with different SNA network IDs to communicate over the TCP/IP network.

To enable LUs from other SNA networks with different network IDs to communicate with SNA over TCP/IP, you have to code YES in the ATCSTRxx start-up options member for VTAM parameter XNETALS.

MAPPING CONSIDERATIONS

To allow SNA application programs to communicate over an IP network, the LU names are formatted into IP domain names that are then mapped to IP addresses. This reformatting is detailed in the following manuals supplied by IBM:

- ‘Defining an LU to VTAM’ in *VTAM Resource Definition Reference.*
- ‘Setting up and using TCP/IP domain name systems’ in the *TCP/IP Planning and Customizing Guide*.
- ‘Defining host names in HOSTS.LOCAL file of TCP/IP’ in the *TCP/IP Planning and Customizing Guide*.

It’s possible to use the HOSTS.LOCAL file, DNS, or both systems to allow the definition of domain names. You need to ensure that the domain names are consistent with the DNSUFX parameter described earlier in this article. Using the gethostbyname call of the socket API of TCP/IP, the SNA-over-TCP/IP system will access the DNS to obtain the IP address associated with a particular SNA LU. The actual DNS can be located anywhere in the IP network as long as the DNS resolver can locate the node on which the desired IP address resides. If the DNS is not accessible, the HOSTS.LOCAL file is searched for the address.

Every destination LU name is mapped to a corresponding Internet address in the DNS. Consequently, each LU name must have a corresponding resource record defined in a DNS resource set. You don’t need to define the domain name associated with the LU in any particular DNS. However, the LU name must be defined in a DNS that can be reached by the originating LU-LU session over the IP network. If a session attempt is made to an LU that is not in DNS, a 80040000 sense code failure will be returned.

In the HOSTS.LOCAL file, each destination LU is mapped to a corresponding Internet address. If both the LOCAL.HOSTS file and

DNS are used to map LU names to IP addresses, you have the additional overhead of maintaining both systems.

OPTIMIZING SNA OVER TCP/IP PERFORMANCE

The performance of SNA over TCP/IP is sensitive to tuning parameters that are specified by both VTAM and TCP/IP. In addition, the SNA over TCP/IP performance will be affected by how VTAM and TCP/IP tuning parameters impact each other. For any sessions that flow across the TCP/IP network, the maximum Request Unit size should be modified for sessions. The maximum RU sizes parameter (RUSIZES) on the MODENT macroinstruction in the mode table is used to adjust these sizes. When the maximum RU size is specified, you need to consider the values you have chosen for TCP/IP window size and the maximum transmission unit. SNA over TCP/IP performance can be enhanced by ensuring that RU maximum size plus 10 bytes of header data is equal to the maximum transmission unit size. IBM also recommends that the TCP/IP window size is a multiple of the maximum transmission unit size.

In the case of high-speed transfer of data, you can decrease CPU cycles and increase data throughput by increasing the maximum RU size.

You should also look at the four TCP/IP timers that can be used to tune SNA over TCP/IP. These are coded in the TCP/IP major node. These are the CONTIMER, DGTIMER, EXTIMER, and IATIMER:

- CONTIMER should be longer than the expected round-trip response time between the host and the remote partner. You need to allow additional time to permit temporary congestion in the IP network to subside.
- DGTIMER needs coding to be slightly larger than the expected round-trip response time between the system and the remote system. Additional time for retries between datagrams should be allowed for temporary congestion in the IP network to subside.
- EXTIMER should be slightly larger than the round-trip response time between this system and the remote systems to which it communicates. A larger value can be coded, thus reducing the number of datagrams sent with the same information (SNA

expedited data) that flows over the TCP connection. Take care, however, as a larger value increases the response time for SNA-expedited data if the TCP connection is congested.

- IATIMER needs to be considered. You need to balance between CPU usage and storage usage. If IATIMER is set too low, VTAM will expend CPU cycles by routinely sending MPTN KEEPALIVE datagrams to test IP address connectivity for idle, but active, connections. If IATIMER is set too high, storage can be allocated for dedicated IP addresses that are actually inactive.

Lastly, the TCB operand in the TCP/IP major node details the number of MVS subtasks that are used by VTAM to access TCP/IP. Each TCB can support up to 120 LU-LU sessions. When setting the TCB operand, take the number of expected LU-LU SNA-over-TCP/IP sessions, not exceeding 11,880, and then divide it by 120 to gain the optimum TCB value. If this number is less than the number of processors in your node, throughput can be optimized by coding the TCB operand to equal the number of processors.

CONCLUSION

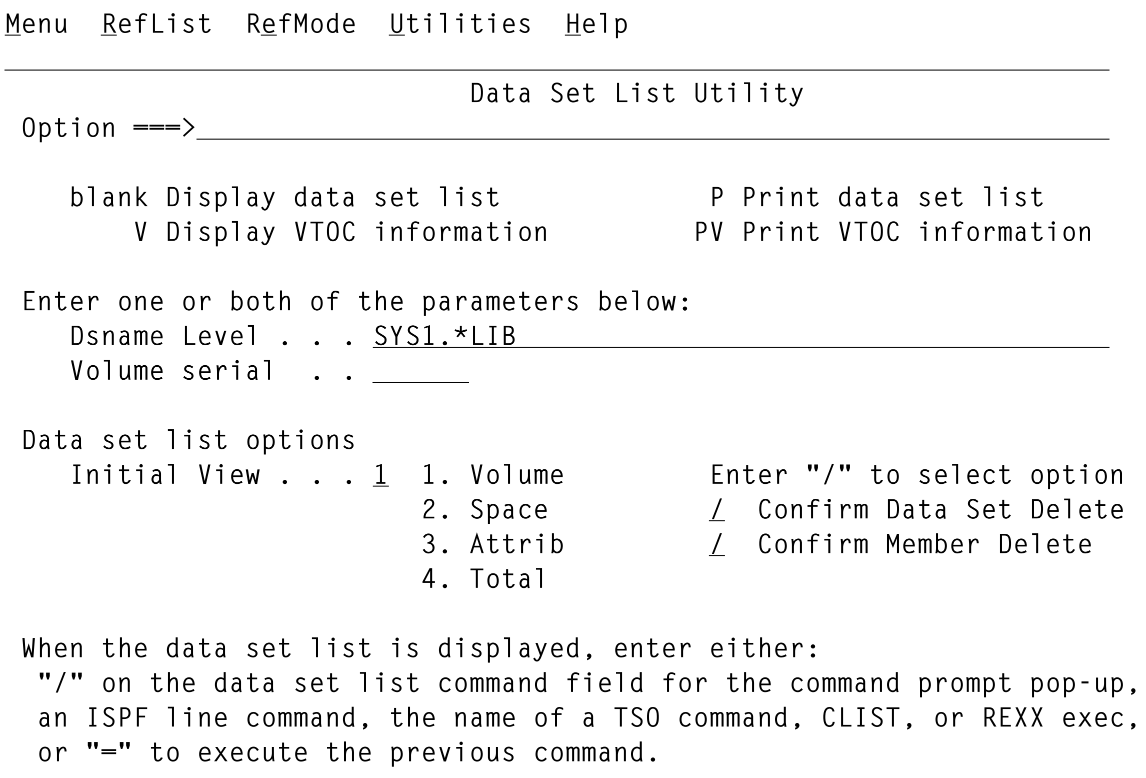
If you have a large investment in SNA networking, or have to link to companies or divisions of your organization that use SNA networks and applications, AnyNet can be an enabler for linking diverse networks and applications. It's an ideal mechanism to help IT personnel meet time-consuming and expensive challenges.

Elizabeth Bradley
Systems Programmer (UK)

Byte by byte through a 3270 datastream

Twenty years ago, I worked with a programmer who was always looking exclusively for examples, refusing to read the manual's description of how a feature worked. If you're like that programmer, or just need something to confirm your understanding after reading the manual, this article in the 3270 datastream series is for you.

It has now been ten years since I last waded through a VTAM I/O trace, trying to solve a CICS application programmer's problem. Reviewing the *3270 Data Stream Programmer's Reference* (GA23-0059), I found potential ambiguities that can only be solved by looking at a real 3270 data stream. So, that's what I've done here: chosen a popular ISPF panel (see Figure 1), and explored and explained its 3270 data stream byte by byte, sometimes bit by bit, as output to a standard size (24x80, ie Model 2) 3270 capable of handling 3270 Extended Attributes, including colour.



```
Menu  RefList  RefMode  Utilities  Help
------------------------------------------------------------------------------
                             Data Set List Utility
 Option ===> __________________________________________________________

    blank Display data set list             P Print data set list
        V Display VTOC information         PV Print VTOC information

 Enter one or both of the parameters below:
    Dsname Level . . . SYS1.*LIB
    Volume serial  . . ______

 Data set list options
    Initial View . . . 1  1. Volume          Enter "/" to select option
                          2. Space           /  Confirm Data Set Delete
                          3. Attrib          /  Confirm Member Delete
                          4. Total

 When the data set list is displayed, enter either:
  "/" on the data set list command field for the command prompt pop-up,
  an ISPF line command, the name of a TSO command, CLIST, or REXX exec,
  or "=" to execute the previous command.
```

Figure 1: ISPF option 3.4 (or DSLIST)

NOTATION

The 3270 datastream is broken down the way it is interpreted by 3270 hardware, as bits, bytes, or half words (two bytes), each expressed in the way that makes most sense, whether decimal, hexadecimal, binary, or EBCDIC character representation. Lacking a more familiar form of expression, Assembler notation will be used, with an upper-case letter followed by a value enclosed in single quotation marks. For example:

- H'21', where H means half word (2 bytes) integer shown with decimal value.
- X'5F', where X means hexadecimal; each digit is half a byte (4 bits).
- B'10010000', where B means binary; each digit is a bit (8 bits to a byte).
- C'Test', where C means character.
- FL1'255', where FL1 means 1-byte integer shown with decimal value.

SFE

The 3270 Start Field Extended (SFE) order is used so extensively in this ISPF panel that the nine different SFEs used are shown once here, and then referred to by number in the datastream itself:

```
SFE type 1
X'29' – Start Field Extended (SFE) order
X'03' – number of attribute type-value pairs that follow
X'C0' – type=3270 field attribute
B'01100000' – value=a byte of bit values
        B'01' – set to make the byte a displayable character
        B'1' – protected field
        B'0' – alphanumeric
        B'00' – display and not selector-pen-detectable
        B'0' – reserved bit
        B'0' – field has not been modified
X'C2' – type=field outlining attribute
X'00' – value=no outlining lines
X'42' – type=foreground colour attribute
X'F4' – value=green

SFE type 2
X'29' – Start Field Extended (SFE) order
```

X'03' – number of attribute type-value pairs that follow
X'C0' – type=3270 field attribute
B'01100000' – value=a byte of bit values

    B'01' – set to make the byte a displayable character
    B'1' – protected field
    B'0' – alphanumeric
    B'00' – display and not selector-pen-detectable
    B'0' – reserved bit
    B'0' – field has not been modified

X'C2' – type=field outlining attribute
X'00' – value=no outlining lines
X'42' – type=foreground colour attribute
X'F7' – value=white

SFE type 3
X'29' – Start Field Extended (SFE) order
X'03' – number of attribute type-value pairs that follow
X'C0' – type=3270 field attribute
B'01100000' – value=a byte of bit values

    B'01' – set to make the byte a displayable character
    B'1' – protected field
    B'0' – alphanumeric
    B'00' – display and not selector-pen-detectable
    B'0' – reserved bit
    B'0' – field has not been modified

X'C2' – type=field outlining attribute
X'00' – value=no outlining lines
X'42' – type=foreground colour attribute
X'F1' – value=blue

SFE type 4
X'29' – Start Field Extended (SFE) order
X'03' – number of attribute type-value pairs that follow
X'C0' – type=3270 field attribute
B'11101000' – value=a byte of bit values

    B'11' – set to make the byte a displayable character
    B'1' – protected field
    B'0' – alphanumeric
    B'10' – intensified display and selector-pen-detectable
    B'0' – reserved bit
    B'0' – field has not been modified

X'C2' – type=field outlining attribute
X'00' – value=no outlining lines
X'42' – type=foreground colour attribute
X'F1' – value=blue

SFE type 5
X'29' – Start Field Extended (SFE) order
X'03' – number of attribute type-value pairs that follow
X'C0' – type=3270 field attribute
B'11101000' – value=a byte of bit values
B'11' – set to make the byte a displayable character

B'1' – protected field
B'0' – alphanumeric
B'10' – intensified display and selector-pen-detectable
B'0' – reserved bit
B'0' – field has not been modified
X'C2' – type=field outlining attribute
X'00' – value=no outlining lines
X'42' – type=foreground colour attribute
X'F5' – value=turquoise

SFE type 6
X'29' – Start Field Extended (SFE) order
X'03' – number of attribute type-value pairs that follow
X'C0' – type=3270 field attribute
B'11101000' – value=a byte of bit values

- B'11' – set to make the byte a displayable character
- B'1' – protected field
- B'0' – alphanumeric
- B'10' – intensified display and selector-pen-detectable
- B'0' – reserved bit
- B'0' – field has not been modified

X'C2' – type=field outlining attribute
X'00' – value=no outlining lines
X'42' – type=foreground colour attribute
X'F7' – value=white

SFE type 7
X'29' – Start Field Extended (SFE) order
X'03' – number of attribute type-value pairs that follow
X'C0' – type=3270 field attribute
B'11110000' – value=a byte of bit values

- B'11' – set to make the byte a displayable character
- B'1' – protected field
- B'1' – numeric
- B'00' – display and not selector-pen-detectable
- B'0' – reserved bit
- B'0' – field has not been modified

X'C2' – type=field outlining attribute
X'00' – value=no outlining lines
X'42' – type=foreground colour attribute
X'F1' – value=blue

SFE type 8
X'29' – Start Field Extended (SFE) order
X'03' – number of attribute type-value pairs that follow
X'C0' – type=3270 field attribute
B'11110000' – value=a byte of bit values

- B'11' – set to make the byte a displayable character
- B'1' – protected field
- B'1' – numeric
- B'00' – display and not selector-pen-detectable
- B'0' – reserved bit

B'0' – field has not been modified
X'C2' – type=field outlining attribute
X'00' – value=no outlining lines
X'42' – type=foreground colour attribute
X'F4' – value=green

SFE type 9
X'29' – Start Field Extended (SFE) order
X'04' – number of attribute type-value pairs that follow
X'C0' – type=3270 field attribute
B'01000000' – value=a byte of bit values
B'01' – set to make the byte a displayable character
B'0' – unprotected field
B'0' – alphanumeric
B'00' – display and not selector-pen-detectable
B'0' – reserved bit
B'0' – field has not been modified
X'C2' – type=field outlining attribute
X'00' – value=no outlining lines
X'41' – type=extended highlighting attribute
X'F4' – value=underscore
X'42' – type=foreground colour attribute
X'F5' – value=turquoise

Although it's clear that the (unnecessarily) large number of SFEs within ISPF indicates that the 3270 datastream was generated by a program rather than a programmer, it should be mentioned that each SFE requires a character position on the 3270 screen. Therefore, although it might be tempting to remove all but the last in a series of consecutive SFEs, this would mean that the screen position of all subsequent text would change.

THE ACTUAL DATA STREAM

X'F5' – Erase/Write (EW) command code
B'11000011' – the Write Control Character (WCC) byte
B'11' – set to make the byte a displayable character
B'00' – reserved
B'0' – do not initiate print of screen
B'0' – do not sound alarm
B'1' – unlock keyboard
B'1' – reset Modified Data Tag (MDT) bits in the field attributes
X'11' – Set Buffer Address (SBA) order
H'0' – row one, column one (screen byte zero)
see SFE #1 – start protected green field
see SFE #6 – start protected field, bright if monochrome, white if colour
C' ' – text displayed (a single blank)
X'28' – Set Attribute (SA) order
X'41' – type=Extended Highlighting attribute

X'F4' – value=Underscore
C'M' – text displayed
X'28' – Set Attribute (SA) order
X'00' – type=all character attributes
X'00' – value=reset all attribute types
C'enu' – text displayed
see SFE #6 – start protected field, bright if monochrome, white if colour
C' ' – a blank displayed
X'28' – Set Attribute (SA) order
X'41' – type=Extended Highlighting attribute
X'F4' – value=Underscore
C'R' – text displayed
X'28' – Set Attribute (SA) order
X'00' – type=All character attributes
X'00' – value=reset all attribute types
C'efList' – text displayed
see SFE #6 – start protected field, bright if monochrome, white if colour
C' R' – text displayed
X'28' – Set Attribute (SA) order
X'41' – type=Extended Highlighting attribute
X'F4' – value=Underscore
C'e' – text displayed
X'28' – Set Attribute (SA) order
X'00' – type=All character attributes
X'00' – value=reset all attribute types
C'fMode' – text displayed
see SFE #6 – start protected field, bright if monochrome, white if colour
C' ' – blank displayed
X'28' – Set Attribute (SA) order
X'41' – type=Extended Highlighting attribute
X'F4' – value=Underscore
C'U' – text displayed
X'28' – Set Attribute (SA) order
X'00' – type=All character attributes
X'00' – value=reset all attribute types
C'tilities' – text displayed
see SFE #6 – start protected field, bright if monochrome, white if colour
C' ' – blank displayed
X'28' – Set Attribute (SA) order
X'41' – type=Extended Highlighting attribute
X'F4' – value=Underscore
C'H' – text displayed
X'28' – Set Attribute (SA) order
X'00' – type=All character attributes
X'00' – value=reset all attribute types
C'elp' – text displayed
see SFE #1 – start protected green field
X'3C' – Repeat to Address (RA) order
H'80' – stop address=column 80
C' ' – character to be repeated=blank
see SFE #3 – start protected blue field

X'08' – Graphic Escape (GE) order
X'A2' – selects a (single character) horizontal line in the centre of the row from the alternate character set (APL2)
The above two bytes are repeated here nearly 80 times.
C' ' – blank displayed
see SFE #1 – start protected green field
X'3C' – Repeat to Address (RA) order
H'188' – stop address=188
C' ' – character to be repeated=blank
see SFE #3 – start protected blue field
C'Data Set List Utility' – text displayed
see SFE #1 – start protected green field
X'3C' – Repeat to Address (RA) order
H'239' – stop address=239
C' ' – character to be repeated=blank
see SFE #1 – start protected green field
see SFE #1 – start protected green field
C'Option ' – text displayed
X'3C' – Repeat to Address (RA) order
H'251' – stop address=239
C'=' – character to be repeated=equals sign
C'>' – text displayed
see SFE #9 – start unprotected green turquoise field with underlining
X'3C' – Repeat to Address (RA) order
H'319' – stop address=screen byte 319
X'00' – character to be repeated=Null character
see SFE #1 – start protected green field
X'3C' – Repeat to Address (RA) order
H'387' – stop address=screen byte 387
C' ' – character to be repeated=blank
see SFE #9 – start unprotected green turquoise field with underlining
X'3C' – Repeat to Address (RA) order
H'399- stop address=screen byte 399
C' ' – character to be repeated=blank
see SFE #1 – start protected green field
see SFE #1 – start protected green field
C' ' – blank displayed
see SFE #1 – start protected green field
see SFE #2 – start protected white field
C'blank' – text displayed
see SFE #6 – start protected field, bright if monochrome, white if colour
C'Display data set list' – text displayed
X'3C' – Repeat to Address (RA) order
H'435' – stop address=screen byte 435
X'00' – character to be repeated=Null
see SFE #1 – start protected green field
C' ' – four blanks displayed
see SFE #1 – start protected green field
see SFE #2 – start protected white field
C' ' – blank displayed
X'3C' – Repeat to Address (RA) order

H'446' – stop address=screen byte 446
C' ' – character to be repeated=blank
C'P' – character to display
see SFE #5 – start protected field, bright if monochrome, turquoise if colour
C'Print Data Set List' – text displayed
X'3C' – Repeat to Address (RA) order
H'473' – stop address=screen byte 473
X'00' – character to be repeated=Null
see SFE #1 – start protected green field
C' ' – four blanks displayed
see SFE #1 – start protected green field
see SFE #1 – start protected green field
C' ' – blank displayed
see SFE #1 – start protected green field
see SFE #2 – start protected white field
C' ' – blank displayed
X'3C' – Repeat to Address (RA) order
H'488' – stop address=screen byte 488
C' ' – character to be repeated=blank
C'V' – character to be displayed
see SFE #5 – start protected field, bright if monochrome, turquoise if colour
C'Display VTOC information' – text displayed
X'00' – a single null character displayed
see SFE #1 – start protected green field
C' ' – four blanks to be displayed
see SFE #1 – start protected green field
see SFE #2 – start protected white field
C' ' – blank displayed
X'3C' – Repeat to Address (RA) order
H'525' – stop address=screen byte 525
C' ' – character to be repeated=blank
C'PV' – characters to be displayed
see SFE #5 – start protected field, bright if monochrome, turquoise if colour
C'Print VTOC information' – text displayed
X'3C' – Repeat to Address (RA) order
H'553' – stop address=screen byte 553
X'00' – character to be repeated=null
see SFE #1 – start protected green field
C' ' – four blanks to be displayed
see SFE #1 – start protected green field
C' ' – blank displayed
see SFE #1 – start protected green field
X'3C' – Repeat to Address (RA) order
H'640' – stop address=screen byte 640
C' ' – character to be repeated=blank
see SFE #4 – start protected field, bright if monochrome, blue if colour
C'Enter one or both of the parameters below:' – text displayed
see SFE #4 – start protected field, bright if monochrome, blue if colour
X'3C' – Repeat to Address (RA) order
H'719' – stop address=screen byte 719
C' ' – character to be repeated=blank

see SFE #1 – start protected green field
see SFE #1 – start protected green field
C' ' – blank displayed
see SFE #1 – start protected green field
see SFE #1 – start protected green field
C'Dsname Level . . .' – text displayed
see SFE #9 – start unprotected green turquoise field with underlining
C'SYS1.*LIB' – text displayed
X'3C' – Repeat to Address (RA) order
H'787' – stop address=screen byte 787
X'00' – character to be repeated=null
see SFE #1 – start protected green field
X'3C' – Repeat to Address (RA) order
H'800' – stop address=screen byte 800
C' ' – character to be repeated=blank
see SFE #1 – start protected green field
C' ' – blank displayed
see SFE #1 – start protected green field
see SFE #1 – start protected green field
C'Volume serial . .' – text displayed
see SFE #9 – start unprotected green turquoise field with underlining
X'3C' – Repeat to Address (RA) order
H'829' – stop address=screen byte 829
X'00' – character to be repeated=null
see SFE #1 – start protected green field
see SFE #1 – start protected green field
C' ' – blank displayed
see SFE #1 – start protected green field
see SFE #7 – start protected numeric blue field
see SFE #7 – start protected numeric blue field
nnnC' ' – a large number of blanks displayed (about 130)
see SFE #4 – start protected field, bright if monochrome, blue if colour
C'Data set list options' – text displayed
X'3C' – Repeat to Address (RA) order
H'1040' – stop address=screen byte 1040
X'00' – character to be repeated=null
see SFE #1 – start protected green field
C' ' – blank displayed
see SFE #1 – start protected green field
see SFE #1 – start protected green field
C'Initial View . . .' – text displayed
see SFE #9 – start unprotected green turquoise field with underlining
C'1' – text displayed
see SFE #8 – start protected numeric green field
see SFE #2 – start protected white field
C'1.' – text displayed
see SFE #2 – start protected white field
C'Volume' – text displayed
X'3C' – Repeat to Address (RA) order
H'1081' – stop address=screen byte 1081
C' ' – character to be repeated=blank

see SFE #1 – start protected green field
see SFE #5 – start protected field, bright if monochrome, turquoise if colour
C' ' – blank displayed
see SFE #1 – start protected green field
see SFE #1 – start protected green field
C'Enter "/" to selection option' – text displayed
X'3C' – Repeat to Address (RA) order
H'1116' – stop address=screen byte 1116
C' ' – character to be repeated=blank
see SFE #1 – start protected green field
C' ' – three blanks being displayed
see SFE #1 – start protected green field
C' ' – blank displayed
see SFE #1 – start protected green field
X'3C' – Repeat to Address (RA) order
H'1145' – stop address=screen byte 1145
C' ' – character to be repeated=blank
see SFE #2 – start protected white field
C'2.' – text displayed
see SFE #2 – start protected white field
C'Space' – text displayed
X'3C' – Repeat to Address (RA) order
H'1161' – stop address=screen byte 1161
C' ' – character to be repeated=blank
see SFE #1 – start protected green field
C' ' – three blanks being displayed
see SFE #9 – start unprotected green turquoise field with underlining
C'/' – text displayed
see SFE #1 – start protected green field
see SFE #2 – start protected white field
C'Confirm Data Set Delete' – text displayed
X'3C' – Repeat to Address (RA) order
H'1197' – stop address=screen byte 1197
C' ' – character to be repeated=blank
see SFE #1 – start protected green field
C' ' – two blanks being displayed
see SFE #1 – start protected green field
C' ' – blank displayed
see SFE #1 – start protected green field
X'3C' – Repeat to Address (RA) order
H'1225' – stop address=screen byte 1225
C' ' – character to be repeated=blank
see SFE #2 – start protected white field
C'3.' – text displayed
see SFE #2 – start protected white field
C'Attrib' – text displayed
X'3C' – Repeat to Address (RA) order
H'1241' – stop address=screen byte 1241
C' ' – character to be repeated=blank
see SFE #1 – start protected green field
C' ' – three blanks being displayed

see SFE #9 – start unprotected green turquoise field with underlining
C'/' – text displayed
see SFE #1 – start protected green field
see SFE #2 – start protected white field
C'Confirm Member Delete' – text displayed
X'3C' – Repeat to Address (RA) order
H'1277' – stop address=screen byte 1277
C' ' – character to be repeated=blank
see SFE #1 – start protected green field
C' ' – two blanks being displayed
see SFE #1 – start protected green field
C' ' – blank displayed
see SFE #1 – start protected green field
X'3C' – Repeat to Address (RA) order
H'1305' – stop address=screen byte 1305
C' ' – character to be repeated=blank
see SFE #2 – start protected white field
C'4.' – text displayed
see SFE #2 – start protected white field
C'Total' – text displayed
X'3C' – Repeat to Address (RA) order
H'1321' – stop address=screen byte 1321
C' ' – character to be repeated=blank
see SFE #1 – start protected green field
X'3C' – Repeat to Address (RA) order
H'1360' – stop address=screen byte 1360
C' ' – character to be repeated=blank
nnC' ' – a large number of blanks displayed (about 75)
see SFE #4 – start protected field, bright if monochrome, blue if colour
C'When the data set list is displayed, enter either:' – text displayed
see SFE #4 – start protected field, bright if monochrome, blue if colour
X'3C' – Repeat to Address (RA) order
H'1519' – stop address=screen byte 1519
C' ' – character to be repeated=blank
see SFE #1 – start protected green field
see SFE #1 – start protected green field
see SFE #1 – start protected green field
see SFE #5 – start protected field, bright if monochrome, turquoise if colour
C'"/"' – text displayed
see SFE #1 – start protected green field
C'on the data set list command field for the command prompt pop-up,' – text displayed
see SFE #1 – start protected green field
X'3C' – Repeat to Address (RA) order
H'1599' – stop address=screen byte 1599
C' ' – character to be repeated=blank
see SFE #1 – start protected green field
see SFE #1 – start protected green field
see SFE #1 – start protected green field
see SFE #1 – start protected green field
C'an ISPF line command, the name of a TSO command, CLIST or REXX exec,

```
or' – text displayed
see SFE #1 – start protected green field
C' ' – three blanks displayed
see SFE #1 – start protected green field
see SFE #1 – start protected green field
see SFE #1 – start protected green field
see SFE #5 – start protected field, bright if monochrome, turquoise if colour
C'"="' – text displayed
see SFE #1 – start protected green field
C'to execute the previous command.' – text displayed
see SFE #1 – start protected green field
X'3C' – Repeat to Address (RA) order
H'1759' – stop address=screen byte 1759
C' ' – character to be repeated=blank
see SFE #1 – start protected green field
X'3C' – Repeat to Address (RA) order
H'0' – stop address=screen byte 0
C' ' – character to be repeated=blank
X'11' – Set Buffer Address (SBA)
H'253' – buffer address=screen byte 253
X'13' – Insert Cursor (IC) order
```

READ PARTITION

Before the actual datastream for the ISPF panel itself, as shown above, the following Read Partition structured field was sent to the 3270 terminal as part of a Write Structured Field (WSF) command:

```
X'F3' – Write Structured Field (WSF) command
H'6' – length of first structured field
X'40' – Outbound 3270DS structure field
X'00' – Partition ID (PID)
X'F1' – Write partition command
B'11000011' – the Write Control Character (WCC) byte
        B'11' – set to make the byte a displayable character
        B'00' – reserved
        B'0' – do not initiate print of screen
        B'0' – do not sound alarm
        B'1' – unlock keyboard
        B'1' – reset Modified Data Tag (MDT) bits in the field attributes
H'5' – length of structured field
X'01' – Read Partition structured field
X'FF' – partition identifier must be set to X'FF' for query operations
X'02' – Query operation
```

Note that the above X'FF' will have a second X'FF' right after it in all telnet protocols, including tn3270 and tn3270e. This doubling prevents it from being processed by telnet protocol as the Interpret As Command (IAC) code.

QUERY RESPONSE

In response to this Read Partition structured field, the 3270 terminal responded with the following inbound datastream of structured fields:

X'88' – Attention Identifier (AID) that precedes all inbound structured fields
H'14' – length of structured field
X'81' – Query Reply structured field type
X'80' – QCODE=Summary Query Reply
X'80' – Summary QCODE is supported
X'81' – Usable Area QCODE is supported
X'85' – Character Sets QCODE is supported
X'86' – Colour QCODE is supported
X'87' – Highlighting QCODE is supported
X'88' – Reply Modes QCODE is supported
X'99' – Auxiliary Devices QCODE is supported
X'A6' – Implicit Partition QCODE is supported
X'8C' – Field Outlining QCODE is supported
X'97' – Document Interchange Architecture (DIA) QCODE is supported

H'23' – length of structured field
X'81' – Query Reply structured field type
X'81' – QCODE=Usable Area
B'0' – reserved
B'0' – non-page printer
B'0' – reserved
B'0' – not a hard copy device
X'3' – 12/14/16-bit addressing allowed
B'0' – variable cells not supported
B'0' – matrix character
B'0' – values in next four bytes are cells (not pels)
B'00000' – reserved
H'80' – width of usable area=80 cells
H'24' – height of usable area=24 cells
X'01' – units of measure of pels=millimetres
H'1',H'3' – distance between pel centres in X direction=1/3 mm.
H'100',H'196' – distance between pel centres in Y direction=100/196 mm.
FL1'9' – number of X units in default cell
FL1'16' – number of Y units in default cell
H'1920' – character buffer size=1920 bytes

H'27' – length of structured field
X'81' – Query Reply structured field type
X'85' – QCODE=Character Sets
B'1' – Graphic Escape (GE) supported
B'0' – multiple LCIDs are not supported
B'0' – Load PSSF is not supported
B'0' – Load PS EXTENDED is not supported
B'0' – only one character slot size is supported
B'0' – two byte coded character sets are not supported
B'0' – CGCSGID is not present
B'00' – reserved

B'0' – LOAD PS slot size match required
B'0' – reserved
B'0' – CCSID not present
B'0010' – reserved
FL1'0' – default character slot width=0
FL1'9' – default character slot height=9
B'00010000000000000000000000000000' – only format type 4 is supported
H'7' – length of each character set descriptor
X'00' – device default character set
B'0' – non-loadable character set
B'0' – single-plane character set
B'0' – single-byte coded character set
B'0' – Local Character set ID (LCID) compare
B'0000' – reserved
X'00' – LCID=00
H'101' – Coded Graphic Character Set Global Identifier (CGCSGID) character set identifier=101
H'37' – CGCSGID code page identifier=37
X'01' – device-specific character set ID=01
B'0' – non-loadable character set
B'0' – single-plane character set
B'0' – single-byte coded character set
B'0' – Local Character set ID (LCID) compare
B'0000' – reserved
X'F1' – LCID=F1
H'963' – CGCSGID character set identifier=963
H'310' – CGCSGID code page identifier=310

H'22' – length of structured field
X'81' – Query Reply structured field
X'86' – QCODE=Colour
B'00000000' – no flags set
X'08' – number of colour attribute value/identifier pairs
X'00' – attribute=device default colour
X'F4' – value=green
X'F1' – colour attribute
X'F1' – value=blue
X'F2' – colour attribute
X'F2' – value=red
X'F3' – colour attribute
X'F3' – value=pink
X'F4' – colour attribute
X'F4' – value=green
X'F5' – colour attribute
X'F5' – value=turquoise
X'F6' – colour attribute
X'F6' – value=yellow
X'F7' – colour attribute
X'F7' – value=white ("neutral")

H'13' – length of structured field
X'81' – Query Reply structured field

X'87' – QCODE=Highlight
FL1'4' – number of attribute value/action pairs=4
X'00' – default attribute value
X'F0' – action=normal highlight
X'F1' – highlight attribute
X'F1' – action=blink
X'F2' – highlight attribute
X'F2' – action=reverse video
X'F4' – highlight attribute
X'F4' – action=underscore

H'7' – length of structured field
X'81' – Query Reply structured field
X'88' – QCODE=Reply Modes
X'00' – field mode is supported
X'01' – extended field mode is supported
X'02' – character mode is supported

H'6' – length of structured field
X'81' – Query Reply structured field
X'99' – QCODE=Auxiliary Device (AUXDA)
B'000000000000000000' – reserved

H'17' – length of structured field
X'81' – Query Reply structured field
X'A6' – QCODE=Implicit Partition
B'000000000000000000' – reserved
FL1'11' – length of parameter
X'01' – implicit partition sizes
B'00000000' – reserved
H'80' – width of the implicit partition default screen size=80
H'24' – height of the implicit partition default screen size=24
H'80' – width of the implicit partition alternate screen size=80
H'24' – height of the implicit partition alternate screen size=24

H'10' – length of structured field
X'81' – Query Reply structured field
X'8C' – QCODE=Field Outlining
B'00000000' – reserved
B'0' – separation of underlining and outlining not supported
B'0000000' – reserved
FL1'0' – location of vertical line=0
FL1'0' – location of overline/underline=0
FL1'0' – location of overline in case of separation=0
FL1'0' – location of underline in case of separation=0

H'18' – length of structured field
X'81' – Query Reply structured field
X'97' – QCODE=DIA
B'000000000000000000' – reserved
H'4096' – maximum DIA bytes/transmissions allowed inbound
H'4096' – maximum DIA bytes/transmissions allowed outbound
FL1'1' – number of three byte function set identifiers that follow=1
X'01000B' – DIA function set identifier

```
FL1'4' – parameter length=4
X'01' – Direct Access ID=01
X'CAFE' – Destination/Origin identification (DOID)
```

The identity tables in some of these structured fields, where attribute values are assigned to themselves, eg X'F5' is defined as X'F5', mean that you can go directly to the published device tables. For example, for colour attribute values, see Table 4-7 of the current edition (1992) of the *3270 Data Stream Programmer's Reference*. A direct URL to the table on the Internet is: http://publibz.boulder.ibm.com/cgi-bin/bookmgr_OS390/BOOKS/CN7P4000/4.4.6.4#TBLCOLRARC

The character sets, including CGCSGID values, are shown in Topic 5 of the *3174 Character Set Reference* (GA27-3831), available on the Internet at: http://publibz.boulder.ibm.com/cgi-bin/bookmgr_OS390/BOOKS/CN7H6001/5.0

EXAMPLES

I spent five years in DEC PDP 11 minicomputer environments and noted how DEC manuals had the most difficult examples. It seemed as if the software developer, who also wrote the manual, wanted to show how smart his program was by demonstrating a capability completely unrelated to the purpose of the software. With that in mind, I have tried to show the 3270 datastream behind a very popular ISPF panel, rather than creating an artificial example of my own.

George Walker
(Canada)

3270 terminal emulation using RUMBA

Continuing our series on 3270 mainframe terminal emulators, we look at RUMBA Office 2000 Version 6.

RUMBA Office 2000 Version 6 was released in February 1999 and therefore doesn’t reflect NetManage’s acquisition of Wall Data a year later (for example, the initial install menu still bears the Wall Data logo). This and other issues will be resolved in Version 7, due in June: it’s promised that “RUMBA7.0 will be Windows 2000 certified and will provide users with live interactive help desk support and strengthened security through SSL.”

NetManage has a number of other products in both the ViewNow and RUMBA lines that also provide 3270 terminal emulation. It all depends on what environments you need to support:

- Any other types of hosts?
- What workstation operating systems?
- How you are connected to the host(s)?
- Windows GUI, Web browser, or thin client?

Price varies with functionality, and how many different host types and client workstation environments need to be supported.

Before we move on, we should also note that this is our first sweep of products, where we focus on those with Windows-based GUIs. RUMBA Web-to-Host will be reviewed in our second sweep.

INSTALLING

Loading the installation CD-ROM into a drive, a menu comes up with the following categories and choices:

```
Rumba 2000
        Install
        Getting Started Help
Administrator’s Kit
        Install
        View
Exit
```

I chose the first Install item and was prompted for an 11-character serial number and 10-character product key (found on a white sticker on the back of the CD sleeve). At first glance, the separation of each into multiple fields on the screen looked like a recipe for trouble, but the fields are fixed length, allowing continuous entry of the entire value, automatically moving between fields. And the code behind it all is smart enough to ignore the hyphen in the product key shown on the sleeve but not required during entry.

The next two dialogue boxes ask for acceptance of the licence agreement and allow you to correct your name and company. The *Select the Setup Options* dialogue box offers three options: express, minimal, and custom.

If you choose 'Express', C:\Program Files\WallData is suggested as an installation directory. A Destination Drive Space section lists 113356 K required and > 2097151 K available.

A Setup alert dialogue box appears immediately after: *The directory "C:\Program Files\WallData" does not exist, would you like to create it?* Yes and No buttons are provided.

Similarly, a *Select Private Directories* dialogue box suggests:

- C:\Program Files\WallData for Private Directories, where user profiles are stored.
- C:\Program Files\WallData\SYSTEM for Interfaces Private Directory, where interface configuration files are stored.

Another Setup alert dialogue box appears for the creation of the SYSTEM directory.

REGISTERING

No prompts occur for more than ten minutes, until the "Wall Data Registration" dialog box. You choose whether or not to register now and can also get information on ONESTEP Support Services.

Following registration, a prompt offers to show you the README file. The README Contents are divided into three sections:

- New features, enhancements, and changes in function.

- Set-up installation program.
- Restrictions.

The first section begins with a subsection detailing changes to the entire RUMBA product family, and is then divided between products.

Nothing caught my eye in the Table of Contents, so I closed the README file and a *Restart Windows* dialog box immediately appeared, with the statement: “Some files could not be updated because they are in use by other programs. Files in use will be updated the next time you restart your computer.”

I was given the choice of restarting now or later, chose now, and removed the installation CD-ROM. The restart took no longer than a normal reboot.

CUTTING IT DOWN TO SIZE

What had changed? A RUMBA 2000 Folder had been added to both the Desktop and to the end of the Start-Programs menu, and available disk space had dropped by 103MB. That may sound a lot, but RUMBA Office 2000 also supports AS/400, VAX, Unix, and HP hosts, and I did choose the Express install option.

Looking at the subfolder RUMBA Administrative Tools, an *Add or Remove Components* brings up the familiar Set-up windows seen during Installation, but does not require the installation CD-ROM. I chose to remove AS/400, Unix, HP, RUMBA Internet Companion, ARPEGGIO Data Access. (I found out what the last two were by single-clicking on them – a description appears on the right.)

The Destination Drive Space showed:

- Required: –49743 K

Remove Shared File? dialogue boxes soon appeared, indicating shared files in C:\WINNT\System32 that were “no longer used by any programs”: SNBD6W9S.DLL, MSXB3032.DLL, ielabel.ocx, and iemenu.ocx.

The Recycle Bin was still empty, and the disk space saving was 40.7MB. The total disk space used by RUMBA was now 63MB.

HOW LONG?

Installation and testing were performed with Windows 2000 Professional on a 300MHz Pentium II processor with 64MB RAM, ATA33 IDE hard drive formatted with NTFS, a CD-RW drive with 6X read, and a 10Mbps NIC connected to 2.5/1Mbps ADSL high-speed Internet.

Assuming that you spend a minute answering the initial prompts, including serial number and product key, it takes about 12 minutes to get to the Registration step. *Gathering File Information* occurs immediately after those initial prompts and lasts almost a full minute. Using the Internet option, the actual electronic registration is all over in five seconds, displaying its progress through the process.

GETTING STARTED

Reviewing the contents of the RUMBA 2000 Folder, a Getting Started on-line book in the subfolder RUMBA Administration Tools seemed the most promising place to start. The Table of Contents reads as follows:

- Introduction
- Installation planning
- Installing RUMBA software
- After installing RUMBA software
- Other Help resources
- Copyrights and trademarks.

The mainframe-relevant sections of the ‘After installing RUMBA software’ chapter are:

- Using the RUMBA 2000 folder icons
- Mainframe host configuration information
- Mainframe host general configuration procedures.

The middle section gives three choices for configuring a connection (only the first two of which were relevant to me):

- RUMBA Assist
- Connection Configuration dialogue box
- APPC Configuration Utility.

Unfortunately, the description of RUMBA Assist left me thinking that it only led you through practice sessions. In fact, it's a sophisticated approach that guides you through configuration and other processes by:

- Greying out all fields of each dialogue box except those where you need to enter or select information.
- After a short period of time, highlighting the menu bar choice you should be making.

I only discovered that I was working with the real thing when I exited a RUMBA Assist session to see the dialogue box I was working with remain on the screen with the changes I had made.

When you first initiate RUMBA by the only logical choice within the RUMBA 2000 Folder, *Mainframe Display*, RUMBA Assist Fast Start comes up first. Just as RUMBA Assist walks you through a process, RUMBA Assist Fast Start walks you through a series of RUMBA Assist processes. The next time you start RUMBA, Assist Fast Start will not appear. You can, however, select from a menu of RUMBA Assist sessions by selecting Help-RUMBA Assist from the menu bar. For example, Fast Start's first Assist can be found by mouse clicking *Connecting to the Host* then *Configuring a Host Connection*.

CONNECTION CONFIGURATION

The first step this leads you through is Connection-Configure from the menu bar and selecting from the Installed Interfaces. To simplify the process, 'Installed Interfaces' is the only area of the dialogue box that is not greyed out. I chose TN3270 because tn3270e was not listed.

Pushing Assist's Next arrow, I was instructed to click the TN3270 tab, but, when I did, the Destination Name/Address box momentarily became greyed out like the rest of the dialogue box. Pushing Assist's Exit button left the TN3270 tab of the Connection Configuration dialogue box still displayed, so I continued by pushing the Insert button as Assist had indicated I should.

A TELNET: New IP Name/Address dialogue box appeared and I filled in the Destination Name/Address field with the numeric IP address of my Internet mainframe host and pushed the OK button. I left the rest of the fields as they were, including blank Device Name and Terminal Type fields.

Clicking the TN3270 Advanced tab, I was reassured by the section title of TN3270/TN3270E Options, and the default check mark beside Extended Attributes. Much later, after considerable searching, I found that this was also the place to choose 3270 monochrome operation (eg CICS where the terminal type matters) by placing a check mark in the *Do Only TN3270* box.

Clicking the General Tab, I returned to the initial display where I had already selected TN3270 from the Installed Interfaces list. This time, I focused on the Auto Connection Options. I selected Auto Connect, but right-clicked on Auto Disconnect and clicked on What's This? from the one entry pop-up menu. I found the displayed statement confusing: "Specifies whether the software automatically terminates a host profile when you close the RUMBA window".

All other attempts to get Help from within the dialogue box gave the same statement, so I pushed the OK button and selected Help-RUMBA Help Topics from the menu bar. The Search tab found no index entry for 'Auto Disconnect', the Connection Configuration dialogue box entry led to an overview, and the 'automatic connections' entry just repeated the same statement above.

GETTING CONNECTED

To see if I had done enough, I selected Connection-Connect and watched what happened. It connected quickly. I maximized the window for ease of readability, but still found the font a bit hard to read. I followed the log-on procedure, noting that the PC <-' Enter' key is 3270 Enter. Everything looked right. The Fn keys were the 3270 PF keys.

Searching for the New Line key was a bit more difficult. Right-Ctrl did nothing, and Shift-Enter was also 3270 Enter. From the menu bar, Options-Keyboard... displays the PC keyboard. The lower left has two fields, with 3270 Emulation Keys selected in the upper field and Alternate Cursor in the lower. Scrolling down the list in the lower field and selecting New Line changes the PC keyboard display to highlight the Caps Lock and Num Lock keys in red and the numeric keypad 'plus' sign in fluorescent green. Exiting and testing verified that the keypad plus key is indeed the 3270 New Line key.

KEYBOARD MAPPING

To move the 3270 Enter key to the right Ctrl on the PC keyboard, and New Line to PC Enter:

- From the menu bar, select Option-Keyboard, or key Ctrl-Shift-K.
- Select Enter from the 3270 Emulation Keys list in the bottom left corner of the Keyboard Settings dialogue box.
- A list of four PC keyboard combinations are displayed to the right, under the heading Mapped To...:
  - Enter
  - Shift+Enter
  - NumEnter
  - Shift+NumEnter
- To delete the first two, you must select each individually and click the Delete button for each; multiple selection is not allowed.
- Push the New button.
- On the PC keyboard displayed at the top of the dialogue box, single-mouse-click on the right Ctrl key.
- Push the Accept button.
- Repeat the process for Shift-Right Ctrl by also single-mouse-clicking the Shift key on the PC keyboard displayed at the top of the dialogue box.
- Select New Line from the 3270 Emulation Keys list, push the

New button, then Enter from the displayed PC keyboard and the Accept button.

- Repeat the process for Shift+Enter.
- Push the OK button at the bottom of the window.
- A *Save Map File?* dialogue box will appear with the message *Save changes to untitled map file?*
- Push the Yes button.
- A *Save Map File As* dialogue box will appear.
- Choose a name for the .map file to be stored in the C:\Program Files\WallData\MFRAME directory.

Assuming that you choose to save the changes to the profile when you exit RUMBA, this keyboard map now becomes the one used in the current profile. During keyboard mapping, if you choose a PC key that's already assigned to a 3270 function, a Warning dialogue box appears with the message "This key sequence is mapped to" followed by the 3270 key name, then "Overwrite?" and Yes and No buttons.

USING PROFILES

After logging off and pushing the Connection icon in the toolbar (to disconnect), clicking the X (exit) button in the upper right corner of the window displays the *Save Profile?* dialogue box with the message "Save untitled session as a profile?" and three buttons: Yes, No, and Cancel. Choosing Yes takes you to a Save Profile dialogue box with a default directory of C:\Program Files\WallData\MFRAME. The File Name field is blank and the Save as type field shows Display Profile. Enter a file name and push the Save button, and RUMBA exits.

You can create a shortcut to the profile you just created, put it on the Desktop or Start menu, and use it to start up RUMBA with the settings you've saved. Alternatively, you can save your settings into the default (untitled) profile that's loaded whenever you click on Mainframe Display from the RUMBA 2000 Folder.

RUMBAAssist will also help you automatically log in to the mainframe.

From the menu bar, go to Help-RUMBA Assist-Simplifying Your Work-Automatically Logging On. If you're currently connected to the host, even if you're logged off, you'll get the message "Please disconnect before running this lesson".

Assist does an excellent job of walking you through the process of recording your log in keystrokes into a macro, saving it, and having it automatically played whenever you connect to the host. The only minor fault I could find in the step-by-step assistance provided was that it seemed to get into a loop on the last step: selecting Exit from the File menu. Passwords (and all hidden fields) are stored in the macro, but are not displayed in the macro processor's window as the macro runs.

CONCLUSION

When PC workstations originally replaced 3270 terminals, RUMBA was the most popular choice in terminal emulation software. Ten years later, there's still nothing wrong with RUMBA as a high-end product. It will be interesting to see, when I review the Web-to-Host RUMBA product later in this series, how far NetManage has moved the product ahead since acquiring it and owner Wall Data.

Armand Minet
(Canada)

3270 terminal emulation using ZOC shareware

When I was first given access to a mainframe test site late last year, I was in a hurry to ensure that I could get connected. I quickly looked through ZDNet, TUCOWS, and CNet for shareware most likely to work for tn3270e. For my purposes, ZOC seemed to have the best combination of reviews and functionality. Since then, there have been significant improvements to the product. In fact, Version 4.00 was released just three days before I did this review of it. I chose a fully functional evaluation copy directly downloadable from the developer’s Web site at http://www.emtec.com/download.html#zocfiles

INSTALLATION AND GETTING CONNECTED

The .exe file is just over 1MB in size and I ran it directly, rather than saving the download. When set-up completes, you’re left in the newly-created ZOC Terminal folder, with 10 icons displayed. To start the product, you double-click the tenth icon, labelled ZOC. The same ZOC Terminal folder has also been added to the Start-Programs menu.

When you start ZOC, a Licence Agreement dialogue box appears with an International Licence Agreement displayed. When you ‘agree’, an EmTec ZOC help dialogue box appears, opened to a section entitled *Quick Start Guides*, with six listed: Telnet, ISDN, Modem, Scripting/ programming, System/network administrators, and Upgrading from earlier versions.

Almost immediately after, a *Manual Connection* dialogue box appears, under and partially obscured by the help dialogue box. Click on it and the help dialogue box disappears. Enter the IP address of the mainframe host in the *Connect to* field, either as four numbers separated by periods or as a domain name. Leave the *Device* field with the default value of Telnet, and change the *Emulation* field to IBM 3270. Leave the checkmark in the *Show this window when starting the program box*, and push the OK button.

VISIBLE NULLS

Unlike the older version of ZOC that I tested a few months ago, the initial VTAM log-on panel of my mainframe host is now visible with Version 4.00, but with one obvious abnormality not present in the previous version: much of this and all subsequent screens is filled with large solid dots in blue, white, and green, wherever Null characters, not blanks, are displayed, including the status line (Line 25). However, the good news is that the default keyboard mapping has the PC Enter key as 3270 New Line and the right Ctrl as Enter.

Even more annoying is the fact that the dots in a field change colour as soon as you start entering information into that field, presumably tracking the status of the Modify Data Tag (MDT) attribute (bit) for the field. Despite a thorough and time-consuming search of Help, I was unable to figure out how to get rid of those dots.

Some time later it occurred to me that, since the dots had not been there in the relatively recent version of the product I had tested a few months ago, it would be worth looking through a version history of the product. Version histories are provided in both the ZOC folder and on the Web site.

Sure enough, the following entry was discovered:

```
[19.02.01] NEW: 3270 option to display null characters
```

making it part of:

```
[22.02.01] WIN-REL: VERSION 3.94 [last beta before 3.99 release
candidate]
```

Clearly the Help and other documentation files have not kept up with the latest changes.

After a lot of looking at dialogue boxes, I finally figured out how to get rid of those pesky dots:

- From the menu bar, select *Options* then *Session Settings*
- Click the *Emulation* tab
- Remove the check mark from the *Show Null Characters* box.

KEYBOARD MAPPING

This is also where you can easily change the 3270 Enter key assignment. A check mark in the *Swap Ctrl and Enter* box will assign the right Ctrl key to 3270 New Line and Enter to Enter. Although I didn't test it, a Keyboard Mapping facility is provided: from the menu bar, select *Options* then *Manage Key Maps...*

I could only find documentation in Help for the default mapping of a few of the more common 3270 keys, including:

- Reset is Esc
- Clear is Shift+Esc
- Erase EOF is Ctrl+End.

And there is no easy way to display the rest from the keyboard mapping facility.

NO COLOUR SUPPORT

The biggest issue for most will be the fact that there is no support for 3270 colour. In *Help*, under *Emulation Options*, you will find the following statement: "ZOC does not support 3270-colour coding but offers a range of predefined colour styles (based on common 3270 emulators)."

For many users, this may not be an issue. An obvious exception is the mainframe developer of on-line applications who will need to test in both colour and monochrome. A less obvious exception is the CICS user. In the past, I have hung the next available VTAM Logical Unit (LU), effectively hanging an entire CICS network, by changing a 3270 emulator from colour to monochrome terminal type. CICS can be quite picky about 3270 terminal types matching the way they were defined.

Even in monochrome, an intermittent error occurred during the display of ISPF Option 3.4. Sometimes the last 'e' in the word 'Member' in the phrase *Confirm Member Delete* would be white when all the rest of the phrase was blue.

Beyond colour, ZOC supports the common screen sizes: 3270 Models 2-5. Finally, even if you are logged off your mainframe host, you cannot exit without disconnecting. From the menu bar, select *Device* then *Disconnect*.

CONCLUSIONS

Like many of the high-end shareware communications programs, ZOC provides all forms of terminal emulation for power users, including the traditional PC BBS audience via VT100 and dial-up. The product was created in Germany and you will see some evidence of the fact. For example, *Help-Help Contents-Contents* tab lists *Menu Kommands* as the second entry.

Although I didn’t take the time to test them, ZOC has many of the features of the most popular high-end products used for 3270 emulation in large organizations. It may be a viable choice if you can live without 3270 colour support – and that’s a big “if”.

Any plan to deploy this product would definitely have to include turning off the default display of Nulls as big dots. The default assignment of 3270 Enter to the PC right Ctrl key is a big plus for projects that replace 3270 terminals, but it will probably have to be changed for PC users new to the mainframe or currently using another 3270 emulator where Enter is Enter.

Armand Minet
(Canada)

TCP/SNA news

IBM has made a number of announcements, including the following:

1 Issuing details of z/OS V1R2, due in Q3. Relevant enhancements include allowing easy definition of TCP/IP configuration files, and automated restart across TCP/IP network outages. The tn3270 function, in conjunction with client access software, will support the use of digital certificates in place of user IDs and passwords to sign the user on to SNA applications. Host On Demand users will be able to sign on to multiple SNA applications with a single digital certificate.

z/OS Communications Server will get parallel sysplex qualities of service and workload distribution functions, TCP/IP restart, and storage management enhancements. Convergence to IP networks will be supported through compatibility with leading networking infrastructure providers, improved migration to dynamic routing protocols, consistent name resolution, updated DNS support (BIND9), and multiple FTP enhancements.

2 Features for the VSE/ESA V2R6, which include added TCP/IP support for the CICS ECI. Internet security is also targeted with the addition of SSL to TCP/IP for VSE/ESA.

3 Version 3 Release 1 of z/VM. Enhancements include an MPROUTE server for the TCP/IP feature, promising greater efficiency within a TCP/IP network, and integration of the TCP/IP Kerberos DES Feature.

4 Release 4 of its Tivoli NetView for OS/390, adding TCP/IP management services for OS/390 and z/OS.

5 Version 1.1 of its Host Access Client Package bundle.

URLs:
http://www.ibm.com/servers/eserver/zseries/
http://www.ibm.com
http://ibm.com/eserver/zseries/zvm
http://www.tivoli.com
http://www.ibm.com/software/network/

* * *

Microsoft Visio Enterprise Network Tools now offers Visio AutoDiscovery and Layout. This provides users with an SNMP discovery engine for automated inventory of devices and network connectivity.

URL: http://www.microsoft.com/office/visio/

* * *

Computer Associates has begun shipping Version 6.1 of its NetworkIT
NetMaster for TCP/IP network management tool for the OS/390.

URL: http://www.ca.com

* * *