37

TCP/SNA

update

March 2000

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/QNA
1301 West Highway 407, Suite 201-405
Lewisville, TX 75077-2150
USA
Telephone: 940 455 7050

Contributions

Articles published in *TCP/SNA Update* are paid for at the rate of £170 ($250) per 1000 words and £90 ($140) per 100 lines of code for original material. To find out more about contributing an article, without any obligation, please contact us at any of the addresses above and we will send you a copy of our *Notes for Contributors*.

***TCP/SNA Update* on-line**

Code from *TCP/SNA Update* can be downloaded from our Web site at http://www.xephon.com/tcpsnaupdate.html; you will need the user-id shown on your address label.

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 making any use of it.

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.

Printed in England.

IP Version 6 – a closer look

The Internet protocol (IP) was designed to be a simple but robust communication protocol, and was never intended for use in corporate networks. Since the beginning of the 1990s, however, the network of networks has been expanding at an incredible rate, and the Internet technology contained in the TCP/IP suite of networking protocols has become universal and omnipresent.

Because of the success of the Internet and intranets, IP Version 4 is having to be replaced by a new suite of protocols, able to simultaneously support both the open but loosely bound Internet and the high functional demands of corporate intranets. This new suite of protocols is IP Version 6 (IPV6), also called IP next generation (IPng).

IP Version 6 was recommended in 1994 by the Internet Engineering Task Force (IETF) and the Internet Engineering Steering Group (IESG); it became a Proposed Standard in 1995 and an Internet Draft in 1997. This means that the basic protocol should now be fairly stable, but that work is in progress and changes are still to be expected. The protocol is being developed by the IETF IPng working group, and will be used by hundreds of millions of computers. Over the next decade, IP Version 6 will become the most successful network protocol ever designed.

PROBLEMS WITH IP

A new version of the IP was needed for three main reasons, namely:

- Problems with the Internet.
- Problems with IP version 4.
- New functional demands caused by the emerging corporate intranets.

Each device (‘node’) currently connected to the Internet (using IP Version 4) has a globally unique Internet address which is 32 bits long. To manage the addresses and routing on the Internet, the current IP address is divided into a network address (‘netid’, class A: 8 bits, class B: 16 bits, or class C: 24 bits) and a host address (‘hostid’, occupying the rest of the 32 address bits).

Although the current theoretical IP address space contains about four billion addresses, Internet addresses are not used economically. During the 1980s, a number of organizations acquired an Internet class A address, each tying up 16 million host addresses. Today, these organizations probably use less than 1% of the addresses allocated to them. The same is true to a lesser extent for class B and C addresses.

This inefficiency means that between 150 and 250 million hosts can be effectively connected to the Internet. Although this seemed ample when the IP protocol was designed, it is totally inadequate for the future Internet; if the Internet keeps growing at the current rate, the IP address space will be exhausted before the end of this decade. Hiding entire corporate networks behind firewalls is no solution – the available stock of class C network addresses will be exhausted because of the huge demand for Internet connections. The only workable solution is the introduction of longer addresses into the Internet, thus enlarging the IP address space.

Routers at the top of the Internet backbone hierarchy need to know the routes to all networks connected to the Internet. Although this was no problem in the early days, by 1996 the number of connected networks had grown to more than 100,000. Because an IP Version 4 address is an identifier, not a locator, there is no relation between the address and the network topology; routers, and especially backbone routers, need to keep a separate entry for almost every network connected to the Internet, thus putting an enormous strain on their memory and processing power. Although the problem is currently being alleviated by allocating blocks of consecutive network addresses to Internet Service Providers (ISPs) and by summarizing routes (the CIDR scheme of address allocation and routing), the current scheme cannot support the future growth of the Internet. Only a change in the architecture can help.

There are also a number of problems with the IP protocol itself. The allocation of addresses to hosts is in principle manual, and is not an integral part of the protocol. This creates a high risk of configuration errors, and offers insufficient support for mobile users. Solutions such as the Dynamic Host Configuration Protocol (DHCP) solve part of the problem but are insufficiently integrated into the current IP protocol suite. The current version of IP also has an inefficient header structure: the length of the header is variable, it contains optional fields, and was not designed for high-speed processing.

Finally, as intranet applications expand and become mission-critical, new demands are being put on future Internet protocols. Corporate networks using IP should be able to distinguish between different traffic priorities. Any future protocol should also provide for security functions, such as encryption and authentication. And the emergence of new multimedia traffic requires the ability to reserve bandwidth on the Internet or an intranet in order to minimize delay and jitter.

IP VERSION 6

The answer to these problems comes with IP Version 6 (Version 5 was a little-known experimental protocol). Version 6 is a truly new networking protocol, with a new packet structure and new functions, and will be able to support the future growth of the Internet. However, Version 6 does more than just resolve the addressing problem: it is designed according to modern networking standards, is more efficient, and offers more functions than Version 4. The designers kept in mind that the new protocol had to be flexible and universal, supporting all kinds of devices, ranging from mainframe computers to mobile palmtop computers and even household appliances.

On the other hand, IP Version 6 remains an Internet protocol. It is compatible with Version 4, and starts from the same philosophy. Like Version 4, Version 6 is a routable protocol, uses IP-style source and destination addresses, and provides for best-effort packet delivery. The Version 6 address also identifies the connection of a device to the network (an ‘interface’), and the routing of Version 6 packets is very like the routing of Version 4 packets.

The structure of the IP Version 6 packet is quite simple: it consists of a header and a payload (see Figure 1). The header is always 40 bytes long, which is only twice the minimum length of an IP Version 4

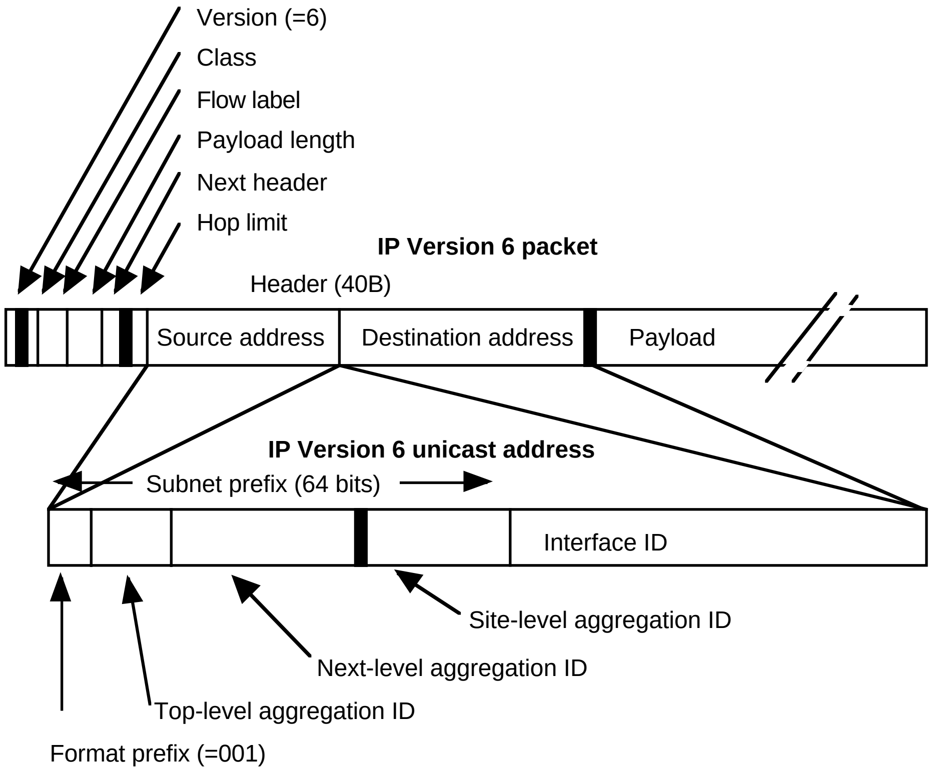

Figure 1: The IP Version 6 packet and the structure of a unicast address

header. A 'normal' payload has a maximum length of 64KB, although it can optionally be longer.

The header includes the IP version number (6), the class and flow label fields to manage the quality of service (see below), the payload length in bytes, the type of the next header, used to include optional features (see below), the hop limit ('time to live' in hop-counts), and the source and destination addresses.

Several of the Version 4 header fields were omitted from the Version 6 header. The header length (IHL) field is not required because the Version 6 header is always 40 bytes long. The Version 6 header contains no header checksum field: error correction is supposed to be done by the datalink layer of the communication. All Version 4 header fields, managing fragmentation, are omitted because Version 6 does not support fragmentation by routers: only fragmentation at the source and destination is supported. All intermediate routers must support a maximum transmission unit (MTU) of at least 576 bytes.

Larger MTU sizes can be supported by sending a path MTU discovery on a connection before sending data packets.

The source and destination addresses used in Version 6 are 128 bits long, spanning a virtually unlimited address space. The address space reportedly provides 665,570,793,348,866,943,898,599 IP addresses per square metre of surface on the earth. Of course, only a small fraction of this enormous address space is currently eligible for use, 85% of it being reserved for future use.

Version 6 addresses are represented as eight groups of four hex digits, separated by colons. Series of zeros may be replaced by a double colon and leading zeros may be omitted. For example, the address FEC0:0000:0000:0000:0000:0000:0A27:E54C may be abbreviated as FEC0::A27:E54C. The special address :: (ie all zeros) is called the unspecified address and should never be used. The address ::1 is the loopback address, used by a node to send packets to itself.

IP Version 6 discerns three types of address: unicast, multicast, and anycast. A unicast address identifies a single interface to the network and is the equivalent of a 'normal' Version 4 address. A multicast address identifies a set of interfaces (in most cases belonging to different devices). When a packet is sent to a multicast address, it is delivered to all interfaces identified by the multicast address. Multicast addresses replace the Version 4 broadcasts. The anycast address is similar to the multicast address, but a packet sent to an anycast address is delivered to just one of the interfaces identified by the address. Anycast addresses are typically used for addressing a set of routers.

Version 6 unicast addresses are split into two parts, which are roughly the equivalents of Version 4's netid and hostid (see Figure 1). The 64 high-order bits are called the subnet prefix. This part essentially has a locator function (it indicates where the network of subnets is) or indicates that the IP address is special. The 64 low-order bits are the interface identifier (interface ID). This identifies the connection of a device to the network (the 'interface'). Interface IDs can be allocated locally, while the subnet prefix is intended to have a global Internet-wide meaning. The MAC address of the network interface card can be used to construct the interface ID.

	Addresses starting with	
	from	**to**
IP Version 4 addresses	0:0:0:0:0:0: 0:0:0:0:0:FFFF:	0:0:0:0:0:0: 0:0:0:0:0:FFFF:
Reserved for OSI NSAP addresses	02	03
Reserved for IPX addresses	04	05
Aggregatable global unicast addresses	2	3
Link-local addresses	FE8	FEB
Site-local addresses	FEC	FEF
Multicast addresses	FF	FF

Figure 2: The planned allocation of the IP Version 6 address space.

The first bits of a Version 6 subnet prefix are called the format prefix, and identify to which part of the Version 6 address space the address belongs. The format prefix is used to divide the total address space into different kinds of addresses (see Figure 2 – note that all addresses beginning with values different from those given in the table are reserved).

Addresses starting with 2 or 3 are aggregatable global unicast addresses. These are the IP addresses that will commonly be used on the Internet. The subnet prefix of such an address is hierarchically structured and is in fact a locator of a network of subnets, connected to the Internet. The first 48 bits of the subnet prefix are allocated by ISPs to customer site connections. The last 16 bits of the subnet prefix can be used by each customer network manager to further subdivide the site network. The 64-bit interface ID is assigned locally to each connection within the site. It can be the MAC address of the network interface or another ‘local’ identification.

The first 48 bits of the subnet prefixes of aggregatable global unicast addresses reflect the routing topology of the Internet, the idea being that networks which are close to each other in the Internet routing topology have partly identical subnet prefixes. This allows for very efficient routing. By using variable-bit subnet masking, routers can summarize blocks of network addresses into a single entry in the routing table. This reduces the size of Internet backbone routing tables from potentially millions to a few thousand entries at most.

Addresses starting with FE8 to FEB are link-local addresses. Such unicast addresses should only be used locally on a communications link, and should never propagate on the Internet. Packets issued by a node with a link-local address are non-routable. The low-order 64 bits of a link-local address are the interface ID, which is locally assigned to each interface. Link-local addresses offer a powerful means of reusing Internet addresses.

IP Version 6 addresses starting with FEC to FEF are site-local addresses. These unicast addresses should be used only within a single site (or organization) and should never be found on the Internet. Routers should not forward packets with a site-local source address outside a site or organization. Site-local addresses are routable within a single site or organization, but not on the Internet. As in all unicast addresses, the low-order 64 bits are the interface ID, assigned locally to each interface. Site-local addresses are intended for use in networks that are not yet connected to the Internet. When the site or organization connects to the Internet, the ISP allocates a subnet prefix to the site. This subnet prefix can then be combined with each of the existing interface IDs to automatically provide a global unicast address for each node. This eliminates the need for massive manual node reconfiguration when an organization or a site connects to the Internet.

Any Version 6 address starting with 0:0:0:0:0:0 is Version 4 compatible, and a node on a Version 6 network will regard the low-order 32 bits of the address as the Version 4 address of a device that understands the Version 6 protocol. An address starting with 0:0:0:0:0:FFFF is a Version 4 mapped address, and a Version 6 node will regard the rest of the address as the Version 4 address of a device that is not Version 6-capable.

Finally, space is reserved in the Version 6 address space for OSI NSAP addresses and for IPX addresses.

Multicast addresses start with FF0 (permanent multicast addresses) or FF1 (transient multicast addresses). Anycast addresses are within the unicast address space.

AUTOMATIC ADDRESS ALLOCATION

IP Version 6 offers two ways to automatically allocate host addresses:

- Stateless configuration is an automatic allocation of IP addresses to stations. In this scheme, routers advertize IP network prefixes on networks to which they are connected. Each interface appends the advertized network prefix with its own unique interface ID to generate a global 128-bit IP address. Stateless address allocation is truly plug-and-play: it requires no manual intervention per station, no address server, and only a minimal router configuration.

All IP Version 6 implementations should support stateless configuration. It is especially useful for mobile devices and if there's no particular interest in precisely what IP addresses are used on the network.

- Stateful configuration is Version 6 of DHCP, in which servers distribute IP addresses and other configuration information to interfaces on request. Stateful configuration allows better control over the allocation of IP addresses than stateless configuration, where each interface chooses its own identifier. It also supports DNS autoregistration to automatically update the DNS entries of registered hosts.

The router broadcasts determine whether stations on a network segment should use stateless or stateful configuration. But even if no router is present, interfaces should be able to generate a link-local address through stateless configuration. And both configuration modes can be combined: interfaces can acquire an IP address through stateless configuration, and can get other configuration information from a DHCP server.

In addition, all allocated Version 6 addresses have a lifetime, which may extend up to infinity. When an address expires, it may be allocated to another interface. The expiration of an address is as follows: the address status changes from 'preferred' (its normal status) to 'deprecated', indicating that the address is still communicating on the network but should not be used any more. Then, the address becomes invalid and can no longer be used.

IP Version 6 also allows an address to be verified for uniqueness by the Duplicate Address Detection algorithm, putting an end to the problem of duplicate IP addresses on loosely managed networks.

The IP Version 6 addressing structure and automatic allocation are powerful and well designed. The tedious reconfiguration currently required when moving nodes or changing network topology can be eliminated using combinations of stateless and stateful configuration. And the fact that it is truly plug-and-play could bring enormous savings to large corporate networks, where moves of equipment and reconfigurations are all too frequent.

OPTIONS

Optional protocol features are implemented in Version 6 via extension headers. An IP Version 6 packet can contain several headers. The first is the actual 40 byte header. The next header field can point to a second header (which is actually contained in the payload of the Version 6 packet). This can contain extra protocol information, and may in its turn point to a third header, etc. The last header field contains the type of higher layer protocol (eg TCP).

Several options are defined in IP Version 6:

- Hop-by-hop options, to carry protocol information relevant to each router. The only such option currently defined is the Jumbo payload, allowing payloads larger than 64KB if the underlying network supports this.
- Routing, to send a packet to a destination via intermediate nodes.
- Fragment, to indicate to the destination that it should assemble fragments.
- Destination options, carrying specific instructions for the destination node.
- Encapsulating security payload, implementing the security framework of IP Version 6 (both authentication and encryption, see below).

The hop-by-hop options are processed by each router on the way. All other options are processed only in the source and destination nodes. Options are included in IP Version 6 in a well-architected manner. Their format is consistent, and there are rules for designing new options and for dealing with options that are not supported by routers or by destination nodes.

The class field in the Version 6 header allows the sending station to define the packet's priority relative to other packets from the same station. Separate priorities are defined for traffic that is or is not sensitive to delay. For both types, the priority field indicates that a router should drop low-priority packets in favour of high-priority ones from the same source. The priority has only a relative value within the packet stream coming from a single source.

An interesting aspect is that intermediate routers may change the class field, inserting their 'own' priorities. For example, an ISP could change the packet priorities according to the service levels agreed with each of its customers.

The flow label header field offers support for 'flows' – a single logical sequence of packets from one source to one destination (unicast or multicast). There may be several simultaneous flows between a source and a destination, each representing a different meaningful packet sequence.

Each flow is given a different flow label, chosen at random by the sending node. This is inserted in each packet of the corresponding flow. Intermediate routers can capture and cache the flow labels and the associated information (source and destination address, next hop, class, etc). The flow labels can be used in combination with a resource reservation protocol, such as RSVP, to reserve bandwidth for a given flow along a path through the network.

Flows are potentially a powerful way of managing traffic in the Internet and on intranets. The use of flow labels allows routers to keep track of logical sequences of packets and their properties and to act on these flows. This enables 'corporate-grade' networking using IP, implementing the separation of traffic by priority, and the definition of service levels and their implementation in routers. Because the actual priorities are part of the IP Version 6 protocol, prioritization and separation of traffic are standard, and implementations should be interoperable.

SECURITY

IP Version 6 offers a consistent means for enabling the authentication and encryption of traffic over an IP network. This is implemented by the authentication header (AH) and the Encapsulating Security Payload (ESP) header, both IP Version 6 extension headers.

By using the security features of IP Version 6, you can:

- Authenticate both parties in a communication.
- Guarantee the confidentiality of the communication, either by encrypting the whole packet (ie tunnelling the traffic) or by encrypting only the higher-layer payload (eg the TCP data).
- Implement non-repudiation (ie the receiver can prove that he/she received the information from the sender).

IP Version 6 does not actually contain new encryption or authentication algorithms; rather, the standard describes how the security features are integrated with the IP protocol, leaving the actual details of the encryption algorithm to the implementation. For authentication, the standard requires the minimal support of the MD5 algorithm with a 128-bit key.

The IP Version 6 security functions constitute the necessary technical basis for implementing large-scale electronic commerce over the Internet and other IP networks. Because authentication and encryption are part of the IP Version 6 protocol, they will be widely implemented, and different implementations will be interoperable. The IP Version 6 security functions should replace the current, often proprietary, security architectures used in secure Web communication and virtual networking over the Internet.

The combination of security and resource reservation in IP Version 6 opens up opportunities for virtual networking over the Internet, where networks at different locations of an organization are transparently interconnected. Version 6 offers a standard way to achieve encrypted tunnelling of packets over insecure networks, ensuring confidentiality. It will also hopefully be possible to reserve bandwidth on the Internet for the secure ‘tunnels’ between the different locations, ensuring good performance.

ISSUES

Although IP Version 6 is becoming a well-defined networking protocol, some issues still need to be resolved before it can replace the current protocol:

- The exact management of the Version 6 address space is still not clear. The unicast addressing architecture defines a hierarchy in the subnet prefixes, making the prefix a locator. The top-level hierarchy in the prefix is the Top-level Aggregation Identifier (TLA ID). The Internet Assigned Numbers Authority (IANA) will allocate blocks of TLA IDs to registries, who will in turn allocate IDs to ISPs. Although some general rules are defined for the allocation of TLA IDs, is not yet clear how TLA IDs will be allocated.

  Moreover, because the subnet prefix is a locator, it is tied to the ISP; if an organization wants to switch to another ISP, its IP subnet prefix has to change. Although such a change is handled smoothly by the IP protocol, some will consider this an unacceptable lock-in. Organizations that are connected to the 'Version 6' Internet through several ISPs will obtain a different subnet prefix from each of the providers they are connected to. They will have to cope with serious routing and addressing problems in their internal networks, for which there is no fundamental solution.

  On the other hand, if the subnet prefix were a pure identifier, all structure would be lost, and the Internet backbone routers would again have to retain an entry for each existing network. The presently proposed addressing architecture aims to minimize the impact of changes and conserve as much independence as possible from the ISPs.

- Although the basics of the Version 6 protocols are well defined, an enormous amount of work remains to be done. The complete TCP/IP protocol suite has to be adapted and expanded for IP Version 6.

- Another issue is the distribution and management of the security keys, for both authentication and encryption. Although fundamental for the smooth operation of a secure Internet, key management is not included in the IP Version 6 protocol.

- Version 6 might be considered too complex to become a successful networking protocol. In the future, the Internet is expected to be present everywhere from mainframe computer to electrical appliances; it is by no means certain that Version 6 will be simple enough to be incorporated in, say, a coffee machine. There are echoes of OSI, which included lots of features and was suitable for all applications, but failed to achieve overall success because it was too complicated to implement.

- The reservation of resources and the proper management of traffic priority on an unstructured network like the Internet is a fundamental problem. Although this issue is not related to IP Version 6 as such, the successful implementation of resource reservation and priorities will be an important factor for its success in corporate networks.

  The fundamental problem is that efficient resource reservation requires some form of connection-oriented networking. Resources, such as bandwidth or router buffers, have to be allocated along a certain path through the network. This contradicts the fundamental nature of the Internet and all other IP networks, which are by architecture connectionless. Many other resource reservation issues also remain unsolved, such as how to manage oversubscription and the pricing and metering of reserved resources.

  Although these issues can probably not be solved for the Internet, resource reservation may be successfully implemented in corporate IP networks, where network designers can better size routers and transmission bandwidth. They can exert control over traffic patterns, decide who gets priority, and design high- and low-priority paths through the network.

- Migration from IP Version 4 to IP Version 6 remains an issue. A ‘big bang’ conversion is impossible, and IP Version 6 is designed to co-exist with the present version of IP. However, it is not fully clear how long both protocols will have to co-exist on the

Internet, who will control the migration process, and who will ‘force’ the last Version 4 users to convert.

It is planned that IP Version 6 nodes will also contain a Version 4 protocol stack for compatibility. This dual stack may cause implementation and configuration problems. As the Version 4 address space nears exhaustion, Version 6-only nodes will inevitably appear. Gateways will be needed to access such nodes, as they cannot be part of the current Internet address space. Who will install and manage such gateways, and how will they work?

CONCLUSION

The IP Version 6 protocol suite solves many of the current problems with the Internet and the present IP protocol. Its address space is considerably larger, and should support not only the accelerating growth of the Internet but also all imaginable network addressing needs on earth.

IP Version 6 is a standard, versatile, and functionally rich protocol, built on the vast existing experience with the Internet and IP protocols. It is both a technical and an architectural improvement over IP Version 4, and introduces functions that appeal to corporate network managers, who prefer stability and manageability to openness. The header structure has been cleaned up, and communications options and automatic address allocation are architecturally included in the protocol. The address allocation can be fully automated, providing truly plug-and-play configuration of connecting nodes.

Because of the fixed header length, the simple structure, and the absence of error checking, routers can easily process Version 6 headers, making the protocol suitable for very fast ‘silicon’-routing, where the processing in routers is shifted to specialized hardware ‘switches’. The routing tables of Version 6 Internet backbone routers will remain fairly small, as hierarchically structured network address prefixes are used for backbone routing.

The protocol is designed to allow a smooth migration from existing IP networks, including the Internet. IP Version 6 is aware of the current IP Version 4 addressing structure and can discern between nodes that understand IP Version 6 and nodes that don’t. No ‘big bang’ scenario is needed to migrate a network (or the Internet) from the current version of IP.

At present, IP Version 6 is a well-defined but still experimental networking protocol. IP Version 6 is being developed by an IETF working group and is widely supported by the networking industry and the Internet community. Besides some technical issues, the main issue is whether mainstream Version 6 implementations will deliver on the promises contained in the standards documents. Finally, it is not clear when Version 6 will gain sufficient momentum in the Internet and in corporate networks: as some of the benefits of IP Version 6 are not obvious, Internet users may be slow to convert. Version 6 risks being considered as a ‘nice-to-have’ rather than a ‘need-to-have’ technology for some time. However, the advantages of IP Version 6 will ultimately become obvious, and its eventual widespread use on the Internet and in corporate networks is inevitable.

Claude Doom
CSC Computer Sciences (Belgium)

IP Version 6 glossary

The following is a glossary of common IP Version 6 terms.

Anycast address	An IP address that identifies a set of interfaces. Any packet sent to an anycast address will be delivered to any one of the interfaces in the set (in principle to the nearest interface).
Deprecated address	An IP address that has in principle expired. The use of deprecated addresses is discouraged but not forbidden.
Destination address	The address of an interface, or a set of interfaces, to which the IP packet is destined.
Global address	An IP address that is valid over the whole Internet.

Header or IP header
The first part of an IP packet, containing addressing and routing information.

Host
Any node that does not forward IP packets.

Interface
The connection of a node to an IP network. An interface is connected to a link.

Interface identifier
An identifier for an interface that should be unique within the link to which the interface is attached. Interface identifiers are in most cases drawn from a link-layer address. Interface identifiers are usually combined with a subnet prefix to construct an IP address.

Internet
The worldwide collection of networks using the IP protocol.

Invalid address
An IP address that is not assigned. It should not be used in any communication. A valid address may become invalid when its valid lifetime has expired.

IP
Internet Protocol. Defines the routing protocol for transfer of packets over the Internet or any IP network.

IP address
The identifier of a single interface or a set of interfaces within the IP protocol. IP (Version 6) addresses are 128 bits long.

IP header
See *Header*.

IP packet
See *Packet.*

IP payload
See *Payload*.

IP tunnel
Virtual connection over which IP packets are encapsulated for transport.

Link
A communications medium over which nodes can communicate on the link layer. Within the IP protocol, all nodes on the same link are neighbours and are considered to be directly

reachable. Examples of links are Ethernet, Token Ring, Frame Relay links, ATM link, and IP tunnels.

Link-layer address
The address of an interface on the link-layer (the MAC layer in the case of a LAN, a phone number, or an X.121 address).

Link-local address
An IP address that is used only on the link to which the interface is attached.

Multicast address
An IP address that identifies a set of interfaces. Any packet sent to a multicast address will be delivered to all interfaces in the set.

Neighbours
All nodes connected to the same link.

Node
Any device that implements IP. Nodes are either hosts or routers.

Packet or IP packet
A single amount of information, carried as a single unit in the IP protocol. A packet consists of a header and a payload.

Payload or IP payload
The actual contents of an IP packet, normally part of an upper layer protocol information exchange.

Preferred address
An IP address that has not expired. It may be used as a source address or a destination address.

Preferred lifetime
The length of time an allocated IP address remains a preferred address. After the preferred lifetime is exceeded, the address becomes a deprecated address.

Prefix
See *Subnet prefix*

Router
A node that forwards IP packets.

Site
A set of networked nodes that are recognized as a logical group and are collectively connected to the Internet.

Site-local address

An IP address that is unique within a single local site.

Source address

The IP address of the interface sending the IP packet.

Subnet prefix

The first part of an IP address. Usually subnet prefixes are advertised by routers. A subnet prefix is combined with an Interface identifier to form an IP address.

Unicast address

An IP address identifying a single interface.

Upper layer

A protocol layer on top of IP. IP packets are used to exchange the necessary content and protocol information of the upper layer protocol.

Valid address

An IP address that is either a preferred address or a deprecated address.

Valid lifetime

The length of time an IP address remains a valid address. When the valid lifetime is exceeded the address becomes an invalid address.

A mailbox system for SMTP under MVS TCP/IP – concluded

Here, we conclude the code for the implementation of a mailbox system for SMTP, based on ISPF functions.

```
*  ESTAE EXIT ROUTINE
RECOVERY EQU   *
         PUSH  USING                    SAVE PREVIOUS BASE REGS
         USING *,R15                    SET UP BASE REGISTER
         USING SDWA,R1                  SET UP ADDRESSABILITY TO SDWA
         LA    R4,12                    PUT 12 IN REGISTER FOR COMPARE
         CR    RØ,R4                    IS SDWA PRESENT?
         BNE   HAVESDWA                 YES, BR TO PROCESS WITH SDWA
         L     RØ,Ø(R2)                 LOAD RETRY ADDR FROM PARM LIST
         LA    R15,4                    SET RC TO RETRY ADDR IN RØ
         BR    R14                      RETURN WITH RETRY ADDR
HAVESDWA EQU   *                        ENTER HERE IF SDWA PRESENT
         ST    R14,12(R13)              SAVE RETURN ADDRESS
         L     R2,SDWAPARM              LOAD PARAM LIST ADDR FROM SDWA
         ST    R2,SDWASRØ1              SAVE POINTER TO ESTAE PARM LIST
         L     R2,4(R2)                 LOAD RETRY ADDRESS
         SETRP RC=4,,RETADDR=(2),RETREGS=YES,FRESDWA=YES,REGS=(14)
         DROP  R15,R1                   DROP LOCAL ADRESSABILITY
         POP   USING                    RESTORE PREVIOUS BASE REGS
*
RTRYRTN1 EQU   *                        RETRY ROUTINE WITH NO SDWA
RTRYRTN2 EQU   *                        ESTAE RETRY ROUTINE WITH SDWA
         LM    R12,R13,8(R1)            LOAD REGS FOR ESTAE PARM LIST
         LA    R15,16                   SET SEVERE ERROR
         STH   R15,ERROR                INDICATE SEVERE ERROR
         B     QUICKOUT                 AND EXIT
STAXEXIT EQU   *
         USING *,R15                    ADDRESS TEMPORAILY
         SAVE  (14,12)                  SAVE REGS
         BALR  R12,Ø                    SET UP BASE
STAXBASE EQU   *
         L     R15,STAXOFFS             SET UP BASE OFFSET
         SR    R12,R15                  SET UP REAL BASE
         DROP  R15                      LEAVE TEMPORARY ADDRESSING
* CLEAN UP WHAT NEED TO
         DROP  R13                      LEAVE ADDRESSING WORKAREA
         USING WORKAREA,R9              ADDRESS WORKAREA
         L     R9,8(R1)                 GET USER DATA
         OI    OPTIONS,ATTN             SET ATTN FLAG
         L     R11,CIBADDR              GET ADDR OF CIB
         QEDIT ORIGIN=COMCIBPT,CIBCTR=Ø DONT ALLOW MODIFIES
         QEDIT ORIGIN=COMCIBPT,BLOCK=(R11) FREE CIB
         POST  TIMEECB                  POST WAIT COMPLETED
         DROP  R9                       LEAVE LOCAL ADDR TO WORKAREA
         USING WORKAREA,R13             ADDRESS WORKAREA NORMALLY AGAIN
         RETURN (14,12),RC=8            RETURN
*
TIMEOFFS DC    A(TIMEBASE-&ID)          TIME BASE OFFSET
STAXOFFS DC    A(STAXBASE-&ID)          STAX BASE OFFSET
COMM     DC    16A(Ø)                   ANSWER ADRESSES
ECBLIST  DS    ØF                       ECBLIST
COMMECBA DC    A(Ø)                     COMMUNICATIONS ECB ADDR
TIMEECBA DC    A(TIMEECB)               SLEEP ECB ADDR
TIMEECB  DC    F'Ø'                     SLEEP ECB
PARMLIST DC    A(VARLEN)                PARMLIST FOR CLIST VAR CREA
         DC    A(MODDATA)               PARMLIST FOR CLIST VAR CREA
         DC    A(VARNLEN)               PARMLIST FOR CLIST VAR CREA
         DC    A(VARNAME)               PARMLIST FOR CLIST VAR CREA
VARLEN   DC    H'Ø'                     LENGTH OF VARIABLE FOR VAR CREA
VARNLEN  DC    Y(L'VARNAME)             LENGTH OF VARIABLE NAME
VARNAME  DC    C'HALT'                  NAME OF CLIST VARIABLE
MODDATA  DS    CL256                    RECEIVING DATA
STOPVARC DC    C'STOP'                  SHOW STOP IN CLIST VAR
TIMEVARC DC    C'TIME EXPIRATION'       SHOW STOP IN CLIST VAR
STOPCM   DC    CL12Ø'STOP ISSUED'       SHOW STOP ISSUED
TIMECM   DC    CL12Ø'SLEEP TIME EXPIRED' SHOW TIME EXPIRATION
MODIFYCM DC    CL12Ø'MODIFY ISSUED; CONTENTS OF MODIFY COMMAND IS:'
PACK     PACK  DW(Ø),Ø(Ø,R8)            EXECUTED PACK
* DEFINE ESTAE AND STAX LIST FORM
ESTAEL   ESTAE MF=L                     CREATE MODEL ESTAE PARM LIST
LESTAEL  EQU   *-ESTAEL                 NAME ITS LENGTH
STAXL    STAX  STAXEXIT,MF=L            STAX LIST FORM
STAXLEN  EQU   *-STAXL                  LENGTH OF STAX
         SYSPRINT
         LTORG
         END

// EXEC ASMCL,MEMBER=TSOLINE1,
// PARM.ASM=RENT,
// PARM.LKED='XREF,LET,LIST,RENT,REUS,REFR'
*
*  CLEAR SCREEN UNDER TSO, USING STLINENO LINE=1
*  THIS TYPE OF CLEARING SCREEN IS LESS DRAMATIC
*  THAN THE FULL SCREEN WRITE IMPLEMENTED IN THE PROGRAM "CLRSCRN"
*  IE IT HAS LESS IMPACT UPON ISPF.
*
*  THE PROGRAM CAN BE USED TO REMOVE THE PAGING CONDITION FROM VTAM
*  WHEN CALLING CLISTS UNDER TSO/ISPF.
*
         PRINT NOGEN
         CVT   DSECT=YES,PREFIX=YES,LIST=NO
         PRINT NOGEN
         IHAASCB
         IHAASXB
         IHAPSA
         USING PSA,RØ
         IKJTCB
         IHAACEE
         IEZJSCB
         IKJPSCB
         IEFAJCTB
         IEFTCT
         IKJTSB
         IEESMCA
         IEFUCBOB PREFIX=YES
UCBPFLEN EQU   UCBCMSEG-UCB
         IEFJESCT .                     JESCT
         IEFJSCVT .                     JSCVT (SSCT)
         IHASDWA DSECT=YES              SDWA FOR ESTAE/SETRP MACRO
         PRINT GEN
WORKAREA DSECT                          GETMAINED WORKARE
SAVEAREA DS    CL72                     SAVE AREA
STAXD    STAX  STAXEXIT,MF=L            STAX LIST FORM
ESTAEW   DS    XL(LESTAEL)              ESTAE PARM LIST AREA
ESTAPARM DS    4F                       PARM LIST TO RETRY ROUTINE:
RETCODE  DS    A                        RETURN CODE
PARMADDR DS    A                        ADDR OF PARMLIST
OPTIONS  DS    C                        EXECUTION OPTIONS
ATTN     EQU   X'8Ø'                    ATTN FLAG SET
WORKLEN  EQU   *-WORKAREA               LENGTH TO GETMAIN
*
&ID      SETC  'TSOLINE1'
&IDLEN   SETA  K'&ID
&ID      INITR SIZE=WORKLEN,AMODE=31,RMODE=ANY,SUBPOOL=Ø,CLEAR=YES
         USING WORKAREA,R13             ADDRESS WORKAREA
         ST    R1,PARMADDR              SAVE ADDR OF PARMLIST
         LA    RØ,RTRYRTN1              RETRY ROUTINE - NO SDWA
         ST    RØ,ESTAPARM              STORE IN PARAMETER LIST
         LA    RØ,RTRYRTN2              RETRY ROUTINE WITH SDWA
         ST    RØ,ESTAPARM+4            STORE IN PARAMETER LIST
         STM   R12,R13,ESTAPARM+8       STORE BASE & DATA REG IN PARM
         MVC   ESTAEW(LESTAEL),ESTAEL   MOVE IN ESTAE PARAMETER LIST
         ESTAE RECOVERY,CT,PARAM=ESTAPARM,MF=(E,ESTAEW) SETUP RCVRY
         MVC   STAXD(STAXLEN),STAXL     MOVE IN STAX LIST TO GETMAINED
         STAX  STAXEXIT,USADDR=WORKAREA,MF=(E,STAXD) SET ATTN EXIT
*
*  NORMAL PROCESSING
*
         TM    OPTIONS,ATTN             IS ATTN FLAG SET
         BO    EXIT                     RETURN IF ATTN
         STLINENO LINE=1                CLEAR SCREEN
         B     EXIT                     RETURN
*
EXITRC4  EQU   *
         MVC   RETCODE,=F'4'            SET RETURNCODE 4
         B     EXIT                     GO EXIT
EXITRC8  EQU   *
         MVC   RETCODE,=F'8'            SET RETURNCODE 8
         B     EXIT                     GO EXIT
EXITRC12 EQU   *
         MVC   RETCODE,=F'12'           SET RETURNCODE 12
         B     EXIT                     GO EXIT
EXIT     EQU   *
         ESTAE Ø                        CANCEL ESTAE EXIT
QUICKOUT EQU   *
         L     R15,RETCODE              GET RETURN CODE
         EXITR RC=(R15)                 RETURN TO CALLER
*
*  ESTAE EXIT ROUTINE
*
RECOVERY EQU   *
         PUSH  USING                    SAVE PREVIOUS BASE REGS
         USING *,R15                    SET UP BASE REGISTER
         USING SDWA,R1                  SET UP ADDRESSABILITY TO SDWA
         LA    R4,12                    PUT 12 IN REGISTER FOR COMPARE
         CR    RØ,R4                    IS SDWA PRESENT?
         BNE   HAVESDWA                 YES, BR TO PROCESS WITH SDWA
         L     RØ,Ø(R2)                 LOAD RETRY ADDR FROM PARM LIST
         LA    R15,4                    SET RC TO RETRY ADDR IN RØ
         BR    R14                      RETURN WITH RETRY ADDR
HAVESDWA EQU   *                        ENTER HERE IF SDWA PRESENT
         ST    R14,12(R13)              SAVE RETURN ADDRESS
         L     R2,SDWAPARM              LOAD PARAM LIST ADDR FROM SDWA
         ST    R2,SDWASRØ1              SAVE POINTER TO ESTAE PARM LIST
         L     R2,4(R2)                 LOAD RETRY ADDRESS
         SETRP RC=4,,RETADDR=(2),RETREGS=YES,FRESDWA=YES,REGS=(14)
         DROP  R15,R1                   DROP LOCAL ADDRESSABILITY
         POP   USING                    RESTORE PREVIOUS BASE REGS
*
RTRYRTN1 EQU   *                        RETRY ROUTINE WITH NO SDWA
RTRYRTN2 EQU   *                        ESTAE RETRY ROUTINE WITH SDWA
         LM    R12,R13,8(R1)            LOAD REGS FOR ESTAE PARM LIST
         LA    R15,16                   SET SEVERE ERROR
         ST    R15,RETCODE              INDICATE SEVERE ERROR
         B     QUICKOUT                 AND EXIT
*
*  STAX ATTENTION EXIT
*
STAXEXIT EQU   *
         PUSH  USING                    SAVE PREVIOUS BASE REGS
         USING *,R15                    ADDRESS TEMPORARILY
         SAVE  (14,12)                  SAVE REGS
         BALR  R12,Ø                    SET UP BASE
STAXBASE EQU   *
         L     R15,STAXOFFS             SET UP BASE OFFSET
         SR    R12,R15                  SET UP REAL BASE
         DROP  R15                      LEAVE TEMPORARY ADDRESSING
         POP   USING                    RESTORE PREVIOUS BASE REGS
* CLEAN UP WHAT NEED TO
         DROP  R13                      LEAVE ADDRESSING WORKAREA
         USING WORKAREA,R9              ADDRESS WORKAREA
         L     R9,8(R1)                 GET USER DATA
         OI    OPTIONS,ATTN             SET ATTN FLAG
         DROP  R9                       LEAVE LOCAL ADDR TO WORKAREA
         USING WORKAREA,R13             ADDRESS WORKAREA NORMALLY AGAIN
         RETURN (14,12),RC=8            RETURN
STAXOFFS DC    A(STAXBASE-&ID)          STAX BASE OFFSET
* DEFINE ESTAE AND STAX LIST FORM
ESTAEL   ESTAE MF=L                     CREATE MODEL ESTAE PARM LIST
LESTAEL  EQU   *-ESTAEL                 NAME ITS LENGTH
STAXL    STAX  STAXEXIT,MF=L            STAX LIST FORM
STAXLEN  EQU   *-STAXL                  LENGTH OF STAX
*
         LTORG
         END

// EXEC ASMCL,MEMBER=ASVTFIND
// PARM.LKED='XREF,LET,LIST,AC=1'    ALSO MENTION IN TSO AUTH TABLES
*
*    VERIFY THE EXISTENCE OF AN ADDRESS SPACE FROM ITS NAME
*
*    TSO COMMAND: ASVTFIND ADDR-SPACE-NAME
*
*    RC 4: ADDRESS SPACE NAME FOUND
*    RC 8: ADDRESS SPACE NAME NOT FOUND
*    RC 12: NO VALID PARAMETER
*
*    MUST RUN AUTHORIZED
*
&ID      SETC  'ASVTFIND'
&IDLEN   SETA  K'&ID
&ID      INITR
         L     R2,Ø(R1)                 GET PARM ADDR
         LH    R3,Ø(R2)                 GET PARM LENGTH
         CH    R3,=AL2(5+&IDLEN)        TEST FOR ZERO DATA
         BNH   EXITR12                  IF ZERO, IGNORE
         LR    R15,R3                   SAVE LENGTH
         SH    R15,=AL2(5+&IDLEN)       REDUCE BY COMMAND HEADER
         LA    R14,&IDLEN+5(R2)         POINT TO FIRST DATA
```

```
RECYCLE  EQU   *
         CH    R15,=H'256'              MORE THAN 256 TO XLATE
         BNH   XLATELST                 GO TO LAST XLATE
         TR    Ø(256,R14),TRTAB         XLATE A BATCH OF 256 BYTES
         SH    R15,=H'256'              COUNT DOWN ALREADY XLATED
         LA    R14,256(R14)             STEP BEHIND
         B     RECYCLE                  AND RECYCLE
XLATELST EQU   *
         BCTR  R15,Ø                    REDUCE FOR EXECUTE
         EX    R15,TRLATE               TRANSLATE TO UPPER
         LOAD  EP=ASVTSCAN,ERRET=EXITR12   GET ASVT SCAN SUBROUTINE
         ST    RØ,ASVTADDR              SAVE ADDR OF ASVTSCAN
         XC    ASCBNO,ASCBNO            CLEAR ADDR SPACE ID
RESCAN   EQU   *
         L     R14,ASCBNO               GET ADDR SPACE ID
         LA    R14,1(R14)               STEP ADDR SPACE ID UP BY ONE
         ST    R14,ID                   AND SAVE IT FOR CALL
         ST    R14,ASCBNO               AND SAVE IT FOR CALL
         L     R15,ASVTADDR             GET ADDR OF ASVT SCAN SUBROUT
         CALLXA (15),(ADDRNAME,TYPE,ASCBADDR) CALL ASVT SCAN SUBR
         CH    R15,=H'8'                TEST FOR END
         BE    EXITR8                   THEN FINISHED
         CH    R15,=H'4'                TEST FOR EMPTY POINTER
         BE    RESCAN                   THEN RECYCLE
         LR    R11,R3                   SAVE LENGTH
         SH    R11,=AL2(5+&IDLEN+1)     REDUCE BY HEADER + 1
         CH    R11,=Y(L'USERNAME-1)     TEST FOR TOO LONG
         BNH   LENOK                    LENGTH ACCEPTABLE
         LA    R11,L'USERNAME-1         USE MAX LENGTH
LENOK    EQU   *
         MVC   USERNAME,=CL8' '         BLANK BEFORE MOVE
         EX    R11,*+4                  MOVE USERID
         MVC   USERNAME(Ø),&IDLEN+5(R2) MOVE USERID
         CLC   USERNAME,ADDRNAME        MATCH JOBNAME
         BNE   RESCAN                   IF NOT, THEN RECYCLE
         EXITR RC=4                     FOUND
EXITR8   EQU   *
         EXITR RC=8                     NOT FOUND
EXITR12  EQU   *
         EXITR RC=12                    INVALID PARAMETER
ASVTADDR DS    F                        ADDR OF ASVT SCAN SUBROUTINE
ASCBNO   DC    F'Ø'                     ADDR SPACE NO
ID       DS    ØF                       ADDR SPACE ID
ADDRNAME DC    CL8' '                   ADDR SPACE NAME
TYPE     DC    CL4' '                   ADDR SPACE TYPE
ASCBADDR DS    A                        ADDR OF ASCB
USERNAME DC    CL8' '                   ADDR SPACE NAME
TRLATE   TR    Ø(Ø,R14),TRTAB           TRANSL INPUT
TRTAB    DC    256AL1(*-TRTAB)          TRANSLATE TABLE USED BY
         ORG   TRTAB+C':'                XLATE OF INPUT AND
         DC    C'@'                      XLATE OF INVALID JOBNAME
         ORG   TRTAB+C'{'
         DC    C'#'
         ORG   TRTAB+C'}'
         DC    C'$'
         ORG   TRTAB+C'a'
         DC    C'ABCDEFGHI'
         ORG   TRTAB+C'j'
         DC    C'JKLMNOPQR'
         ORG   TRTAB+C's'
         DC    C'STUVWXYZ'
         ORG
         LTORG
         END
// EXEC ASMCL,MEMBER=ASVTSCAN,
// PARM.LKED='XREF,LET,LIST,AC=1'    ALSO MENTION IN TSO AUTH TABLES
*
*    GENERAL ROUTINE TO SCAN ASVT FOR ADDRESS SPACES
*
*   INPUT PARM 1: EITHER ADDR SPACE ID (FULLWORD) OR
*                 ADDR SPACE NAME (JOBNAME) 8 BYTES LEFT ALIGNED
*
*   OUTPUT PARM 1: RETURNS JOB NAME (8 BYTES) FOR CALL WITH ID
*                  RETURNS ID (FULLWORD) FOR CALL WITH JOBNAME
*
*          PARM 2: TYPE: STC, JOB, TSU OR INIT (4 BYTES)
*
*          PARM 3: ADDR OF ASCB, OR ZERO
*
*          RETURN CODE: Ø FOR FOUND
*                       4 FOR EMPTY POINTER
*                       8 FOR END
*
*    MUST RUN AUTHORIZED
*
         PRINT NOGEN
         CVT   DSECT=YES
         USING CVT,R11                  SET ADDRESSABILITY TO CVT
         IHAASVT
         USING ASVT,R2                  SET ADDRESSABILITY TO ASVT
         IHAASCB
         USING ASCB,R4                  SET ADDRESSABILITY TO ASCB
         IHAPSA
         USING PSA,RØ                   DUMMY USING FOR PSA
         IKJTSB
         PRINT GEN
         DSECT
ID       DS    ØF                       ADDR SPACE ID
ADDRNAME DS    CL8                      ADDR SPACE NAME
```

```
         USING ID,R5                    ADDRESS ID/ADDRNAME
         DSECT
TYPE     DS    CL4                      ADDR SPACE TYPE
         USING TYPE,R6                  ADDRESS TYPE
ASCBADDR DS    A                        ADDR OF ASCB
         USING ASCBADDR,R9              ADDRESS ASCB ADDR
ASVTSCAN INITR
         LM    R5,R6,Ø(R1)              GET PARAMETERS
         L     R9,8(R1)                 GET PARAMETERS
         XC    ASCBADDR,ASCBADDR        CLEAR ASCB POINTER
         XC    ASCBNO,ASCBNO            INITIATE ASCBNO
         MVI   IND,SEEKID               SET SEARCH FOR NAME
         CLI   Ø(R5),Ø                  NUMERIC CALL (ID)
         BE    CALLNUM                  THEN PROCEED
         MVI   IND,SEEKNAME             SET SEARCH FOR NAME
CALLNUM  EQU   *
         MODESET KEY=ZERO               GET INTO KEY ZERO
         L     R11,CVTPTR               GET ADDRESS OF CVT
         L     R2,CVTASVT               GET ADDRESS OF ASVT
         DROP  R11                      DONT ADDRESS CVT ANY LONGER
         L     R7,ASVTMAXU              GET MAX NUMBER OF ASCBS
         TM    IND,SEEKNAME             TEST SEARCH BY ID
         BO    NOTID1                   IF SO PROCEED
         C     R7,ID                    ARE WE ABOVE MAX
         BL    NOTFND                   THEN EXIT WITH CC 8
NOTID1   EQU   *
         SLL   R7,2                     COMPUTE LENGTH OF ASCB POINTERS
         LA    R7,ASVTENTY(R7)          POINT BEHIND
         LA    R3,ASVTENTY              GET ADDR OF 1ST ASCB POINTER
         USING ASVTENTY,R3              GET ADDRESSABILITY
NEXTASCB EQU   *
         L     R15,ASCBNO               GET ASCB NO
         LA    R15,1(R15)               STEP UP BY ONE
         ST    R15,ASCBNO               AND SAVE IT AGAIN
         L     R15,ASCBNO
         TM    IND,SEEKNAME             TEST SEARCH BY NAME
         BO    NOTID2                   GO PROCESS NAME SEARCH
         CLC   ASCBNO,ID                DO WE HAVE A HIT
         BL    STEPFORW                 THEN TRY NEXT
         MVC   JOBNAME,=CL8' '          BLANK IT
         L     R4,ASVTENTY              GET ASCB POINTER
         LTR   R4,R4                    IS IT NOT AVAILABLE
         BNP   EMPTY                    IGNORE IT
         MVC   TYPE,=CL4'JOB'           SAY IT IS A JOB
         L     R15,ASCBJBNI             GET POINTER TO JOB NAME
         LTR   R15,R15                  IS IT AVAILABLE
         BNZ   JOBFND                   IT IS A JOB
         L     R15,ASCBJBNS             GET POINTER TO STC/LOGON
         MVC   TYPE,=CL4'STC'           SAY IT IS AN STC
         CLC   Ø(8,R15),=CL8'INIT'      TEST FOR INITIATOR
         BNE   NOTINIT                  IF NOT, PROCEED
         MVC   TYPE,=CL4'INIT'          SAY IT IS AN INITIATOR
NOTINIT  EQU   *
         L     R1,ASCBTSB               GET TSB ADDRESS
         USING TSB,R1                   SET ADDRESABILITY TO ASCB
         LTR   R1,R1                    TEST FOR TSO USER
         BZ    JOBFND                   NOT A TSO USER
         TM    TSBFLG5,TSBVTAM          IS IT A VTAM TSB
         DROP  R1                       DROP ADDR TO TSB
         BNO   JOBFND                   NOT A VTAM USER
         MVC   TYPE,=CL4'TSU'           SAY IT IS A TSU
         B     JOBFND                   WE GOT IT FOR SEARCH BY ID
JOBFND   EQU   *
         MVC   JOBNAME,Ø(R15)           GET USER ID
         TR    JOBNAME,TRTAB            TRANSL IN CASE OF BAD POINTER
         CLC   JOBNAME,=CL8' '          BLANK JOBNAME
         BE    EMPTY                    IGNORE IT
         MVC   ADDRNAME,JOBNAME         RETURN JOBNAME
         B     EXIT                     THEN EXIT
NOTID2   EQU   *
         L     R4,ASVTENTY              GET ASCB POINTER
         LTR   R4,R4                    IS IT NOT AVAILABLE
         BNP   STEPFORW                 PROCEED TO NEXT
         MVC   TYPE,=CL4'JOB'           SAY IT IS A JOB
         L     R15,ASCBJBNI             GET POINTER TO JOB NAME
         LTR   R15,R15                  IS IT AVAILABLE
         BNZ   JOBFND2                  IT IS A JOB
         L     R15,ASCBJBNS             GET POINTER TO STC/LOGON
         MVC   TYPE,=CL4'STC'           SAY IT IS AN STC
         CLC   Ø(8,R15),=CL8'INIT'      TEST FOR INITIATOR
         BNE   NOTINIT2                 IF NOT, PROCEED
         MVC   TYPE,=CL4'INIT'          SAY IT IS AN INITIATOR
NOTINIT2 EQU   *
         L     R1,ASCBTSB               GET TSB ADDRESS
         USING TSB,R1                   SET ADDRESSABILITY TO ASCB
         LTR   R1,R1                    TEST FOR TSO USER
         BZ    JOBFND2                  NOT A TSO USER
         TM    TSBFLG5,TSBVTAM          IS IT A VTAM TSB
         DROP  R1                       DROP ADDR TO TSB
         BNO   JOBFND2                  NOT A VTAM USER
         MVC   TYPE,=CL4'TSU'           SAY IT IS A JOB
JOBFND2  EQU   *
         MVC   JOBNAME,Ø(R15)           GET USER ID
         TR    JOBNAME,TRTAB            TRANSL IN CASE OD BAD POINTER
         CLC   JOBNAME,ADDRNAME         DO WE HAVE A HIT
         BNE   STEPFORW                 LOOK FOR NEXT
         MVC   ID,ASCBNO                RETURN ASCB NO
         B     EXIT                     AND RETURN
STEPFORW EQU   *
         LA    R3,4(R3)                 POINT TO ASCB POINTER
         CR    R3,R7                    IS IT THE END
         BL    NEXTASCB                 NO RECYCLE
         B     NOTFND                   NO HIT THIS TIME
EXIT     EQU   *
         MODESET KEY=NZERO              GET INTO USER KEY
         ST    R4,ASCBADDR              RETURN ASCB ADDR
         EXITR                          RETURN WITH CC Ø
EMPTY    EQU   *
         MODESET KEY=NZERO              GET INTO USER KEY
         EXITR RC=4                     RETURN WITH CC 4
NOTFND   EQU   *
         MODESET KEY=NZERO              GET INTO USER KEY
         EXITR RC=8                     RETURN WITH CC 8
ASCBNO   DS    F                        ASCB ID NO
JOBNAME  DS    CL8                      ASID NAME
IND      DC    X'ØØ'                    INDICATOR
SEEKID   EQU   X'Ø1'                    SEARCH FOR ID
SEEKNAME EQU   X'Ø2'                    SEARCH FOR NAME
TRLATE   TR    Ø(Ø,R14),TRTAB           TRANSL INPUT
TRTAB    DC    256C' '                  TRANSLATE TABLE USED TO
         ORG   TRTAB+C'@'               TRANSLATE AWAY INVALID CHARS
         DC    C'@'
         ORG   TRTAB+C'#'
         DC    C'#'
         ORG   TRTAB+C'$'
         DC    C'$'
         ORG   TRTAB+C'A'
         DC    C'ABCDEFGHI'
         ORG   TRTAB+C'J'
         DC    C'JKLMNOPQR'
         ORG   TRTAB+C'S'
         DC    C'STUVWXYZ'
         ORG   TRTAB+C'Ø'
         DC    C'Ø123456789'
         ORG
         LTORG
         END
```

```
// EXEC ASMCL,MEMBER=BREAK
BREAK    INITR
*
*      TSO COMMAND ONLY
*          ALLOWS USE OF PA1 AGAIN AFTER INHIBIT FROM COMMAND NOBREAK
*
         STTMPMD ON,KEYS=NO
         EXITR
         LTORG
         END
```

```
// EXEC ASMCL,MEMBER=NOBREAK
```

```
*
*      TSO COMMAND
*          DISABLES THE USE OF PA1 IN TSO
*
NOBREAK  INITR
         STTMPMD OFF,KEYS=ALL
         EXITR
         LTORG
         END

// EXEC ASMCL,MEMBER=INSØ7Ø
*
*    ROUTINE TO CREATE CLIST/REXX/ISPF USER VARIABLE
*     WHEN RUNNING TSO FOREGROUND OR BACKGROUND CREATE CLIST/REXX VAR.
*     WHEN RUNNING TSO FOREGROUND OR BACKGROUND ISPF, ALSO CREATE ISPF
*     VARIABLE UNDER SAME NAME (SAME NAME IF POSSIBLE DUE TO LENGTH).
*     CLIST VARIABLES WILL BE CREATED UP TO LENGTH OF 252, BUT THE
*     CORRESPONDING ISPF VARIABLE WILL BE TRUNCATED TO MAX LENGTH 8.
*
*     PARM 1: LENGTH OF CONTENTS IN HALFWORD (ZERO WILL CLEAR VARIABLE)
*     PARM 2: CONTENTS OF VARIABLE
*     PARM 3: LENGTH OF NAME OF USER VARIABLE IN HALFWORD, MAX 252
*     PARM 4: NAME OF USER VARIABLE
*
*    CAN BE CALLED BY ANY PROGRAM WHICH SHOULD EXECUTE IN A CLIST/REXX
*     TO RETURN A CLIST/REXX/ISPF VARIABLE TO A CLIST/REXX
*
*     RETURN CODE: Ø FOR OK
*                  4 IF WRONG PARAMETERS I.E
*                    LENGTH OF VARIABLE NAME > 252 OR CONTENTS > 256
*                    LENGTH OF VARIABLE NAME = Ø
*                  4 IF SERVICE ROUTINES IKJCT441 OR ISPLINK NOT FOUND
*                  IF ERROR FROM IKJCT441: RC FROM IKJCT441
*                  IF ERROR FROM ISPLINK : RC FROM ISPLINK
*
         GBLC  &ID
         GBLA  &IDLEN
*
         PRINT NOGEN
         IKJTSVT
         USING TSVT,R1Ø                SET ADDRESSABILITY TO TSVT
         PRINT GEN
*
INSØ7Ø   INITR GENCODE=YES,XLATE=NO,STAX=NO,ESTAE=NO,SCANDATA=NO
*
&ID      CSECT
         XC    RETCODE,RETCODE          CLEAR RETURNCODE
         TM    OPTIONR,IKJEFTØ1         TEST FOR ANY PSCB (TSO FG/BG)
         BZ    EXIT                     IF NOT TSO IN FG OR BATCH, EXIT
         MVC   NAME,=CL256' '           CLEAR VARIABLE NAME
         MVC   VALUE,=CL256' '          CLEAR VARIABLE VALUE
         L     R1,PARMADDR              GET ADDR TO PARAMETER LIST
         LM    R2,R5,Ø(R1)              GET PARAMETERS
         XR    R7,R7                    CLEAR LENGTH OF CONTENTS
         ICM   R7,3,Ø(R2)               GET LENGTH OF CONTENTS
         BZ    EMPTY                    EMPTY VARIABLE
         CH    R7,=H'256'               TEST FOR TOO HIGH
         BH    EXITRC4                  IF SO, ERROR
         ST    R7,VALUELEN              BUILD LENGTH OF CONTENTS
         BCTR  R7,Ø                     REDUCE FOR EXECUTE
         EX    R7,*+4                   MOVE THE STUFF
         MVC   VALUE(Ø),Ø(R3)           MOVE THE STUFF
EMPTY    EQU   *
         XR    R8,R8                    CLEAR LENGTH OF VARIABLE NAME
         ICM   R8,3,Ø(R4)               GET LENGTH OF VARIABLE NAME
         BZ    EXITRC4                  EXIT WITH RC 4 IF ZERO
         CH    R8,=H'252'               TEST FOR TOO HIGH
         BH    EXITRC4                  IF SO, ERROR
         ST    R8,NAMELEN               BUILD LENGTH OF VARIABLE
         BCTR  R8,Ø                     REDUCE FOR EXECUTE
         EX    R8,*+4                   MOVE THE STUFF
         MVC   NAME(Ø),Ø(R5)            MOVE THE STUFF
*
*  USE ALWAYS LINK, THEN CORRECT ADDRESSING MODE IS ALWAYS SET UP
*
*        L     R11,CVTPTR               GET ADDRESS OF CVT
*        L     R1Ø,CVTTVT               GET ADDRESS OF TSVT
*        L     R15,TSVTVACC             GET ADDR OF VARIABLE ACCESS RT
*        CALL  (15),                                                   *
               (ECCODE,                                                *
               NAMEPTR,                                                *
               NAMELEN,                                                *
               VALUEPTR,                                               *
               VALUELEN,                                               *
               TOKEN),                                                 *
               VL                       CALL SERVICE ROUTINE
         LINK  EP=IKJCT441,                                            *
               PARAM=(ECCODE,                                          *
               NAMEPTR,                                                *
               NAMELEN,                                                *
               VALUEPTR,                                               *
               VALUELEN,                                               *
               TOKEN),                                                 *
               VL=1,ERRET=EXITRC4       CALL SERVICE ROUTINE
         ST    R15,RETCODE              SET RETURN CODE
         LTR   R15,R15                  TEST FOR GOOD RC
         BNZ   EXIT                     ELSE ERROR
         LINK  EP=ISPQRY,ERRET=EXIT     ISPF ACTIVE
         LTR   R15,R15                  TEST FOR GOOD RC
         BNZ   EXIT                     IF NOT, THEN NO ISPF AVAIL
         LINK  EP=ISPLINK,              VDEFINE FUNCTION VARIABLE      *
               PARAM=(VDEFINE,                                          *
               NAME,                                                    *
               VALUE,                                                   *
               FORMAT,                                                  *
               VALUELEN),                                               *
               VL=1,ERRET=EXITRC4       CALL SERVICE ROUTINE
         ST    R15,RETCODE              SET RETURN CODE
         LTR   R15,R15                  TEST FOR GOOD RC
         BNZ   EXIT                     IF NOT, NOT ABLE TO VDEFINE
         LINK  EP=ISPLINK,              VERASE, IF EXISTING IN SHARED  *
               PARAM=(VERASE,                                           *
               NAME,                                                    *
               SHARED),                                                 *
               VL=1,ERRET=PUT           CALL SERVICE ROUTINE
PUT      EQU   *
         LINK  EP=ISPLINK,              VPUT TO SHARED                 *
               PARAM=(VPUT,                                             *
               NAME,                                                    *
               SHARED),                                                 *
               VL=1,ERRET=DELETE        CALL SERVICE ROUTINE
         ST    R15,RETCODE              SET RETURN CODE
DELETE   EQU   *
         LINK  EP=ISPLINK,              VDELETE FUNCTION VARIABLE      *
               PARAM=(VDELETE,                                          *
               NAME),                                                   *
               VL=1,ERRET=ENDDEL        CALL SERVICE ROUTINE
ENDDEL   EQU   *
         EXITR                          RETURN WITH RETURN CODE
NAME     DC    CL252' '                 NAME OF VARIABLE
NAMELEN  DC    A(Ø)                     LENGTH OF NAME
VALUE    DC    CL256' '                 VARIABLE CONTENTS
VALUELEN DC    A(Ø)                     LENGTH OF CONTENTS
NAMEPTR  DC    A(NAME)                  ADDR OF NAME
VALUEPTR DC    A(VALUE)                 ADDR OF CONTENTS
TOKEN    DC    A(Ø)                     TOKEN, UNUSED
ECCODE   DC    A(TSVEUPDT)              ENTRY CODE FOR SETTING VALUES
VDEFINE  DC    CL8'VDEFINE'             VDEFINE SERVICE
VDELETE  DC    CL8'VDELETE'             VDELETE SERVICE
VERASE   DC    CL8'VERASE'              VERASE SERVICE
VPUT     DC    CL8'VPUT'                VPUT SERVICE
FORMAT   DC    CL8'CHAR'                FORMAT FOR DATA IN ISPF SERV
ASIS     DC    CL8'ASIS'                SHARED OR PROFILE POOL
SHARED   DC    CL8'SHARED'              SHARED PROFILE POOL
         LTORG
         END
```

```
// EXEC ASMCL,MEMBER=SMFID
*
*    CREATE SMFID IN &SMFID
```

```
*
*     TSO COMMAND
*
*     RETURNS SMFID IN &SMFID
*
         CVT   DSECT=YES
         USING CVT,R11                 SET ADDRESSABILITY TO CVT
         IEESMCA
         USING SMCABASE,R1Ø            SET ADDRESSABILITY TO SMCA
SMFID    INITR
         L     R11,CVTPTR              GET ADDRESS OF CVT
         L     R1Ø,CVTSMCA             GET ADDRESS OF TSVT
         MVC   ID,SMCASID              GET SMFID
         LOAD  EP=INSØ7Ø,ERRET=EXITRC8  GET CLIST VAR SUBR
         LR    R15,RØ                  GET ADDR OF SUBR
         CALLXA (15),(LENGTH,ID,VARLEN,VAR) CALL SUBRUTINE
         DELETE EP=INSØ7Ø              DELETE SUBR AGAIN
EXIT     EQU   *
         EXITR                         RETURN
EXITRC8  EQU   *
         EXITR RC=8                    RETURN WITH ERROR
ID       DC    CL256' '                SMFID TO SUBROUTINE
LENGTH   DC    AL2(L'SMCASID)          LENGTH FOR SUBROUTINE
VAR      DC    C'SMFID'                CLIST VARIABLE
VARLEN   DC    AL2(L'VAR)              LENGTH OF VARIABLE
         LTORG
         END
//*

// EXEC ASMCL,MEMBER=INTSYSIN
*
*   CALLED PROGRAM ONLY
*       CREATES A ONE RECORD DATASET FROM THE PARAMETER FIELD
*
INTSYSIN INITR AMODE=24,RMODE=24        AMODE=24 BECAUSE OF IO
         L     R2,Ø(R1)                GET PARM ADDR
         LH    R3,Ø(R2)                GET LENGTH OF PARM
         LTR   R3,R3                   TEST LENGTH FOR ZERO
         BZ    ERROR                   ERROR IF ZERO
         OPEN  (SYSUT2,(OUTPUT))       OPEN DS
         BCTR  R3,Ø                    REDUCE FOR EXECUTE
         EX    R3,*+4                  MOVE DATA
         MVC   DATA(Ø),2(R2)           MOVE DATA
         LA    RØ,DATA                 GET DATA
         PUT   SYSUT2,(RØ)             PUT PARM AS DATA REC
         CLOSE SYSUT2                  CLOSE OUTPUTDS
         EXITR                         END PROGRAM
ERROR    EXITR RC=12                   SET UP ERROR CODE
DATA     DC    CL1ØØ' '                DATA AREA, MAX 1ØØ BYTES
SYSUT2   DCB   DDNAME=SYSUT2,DSORG=PS,MACRF=(PM),BLKSIZE=32ØØ,         X
```

```
               RECFM=FB,LRECL=8Ø
         LTORG
         END
//*
```

JOBNAME MACRO

The JOBNAME macro shown below returns the name of the current address space:

```
*
*   RETURNS JOBNAME POINTER IN REGISTER, DEFAULT TO R15
*    CAN BE OVERWRITTEN BY JOBNAME (RX)
*
         MACRO
&NAME    JOBNAME &REG
         LCLC  &JOBFND,&JOBNFND,&RNULL
&JOBFND  SETC  'JN1'.'&SYSNDX'
&JOBNFND SETC  'JN2'.'&SYSNDX'
&RNULL   SETC  'JN3'.'&SYSNDX'
         AIF   ('&REG' EQ '').RNULL
         AIF   ('&REG'(1,1) EQ '(').AREG
         AGO   .RNULL
AREG    ANOP
&REGR    SETC  '&REG(1)'
         AGO   .REG
RNULL   ANOP
&REGR    SETC  '15'
REG     ANOP
&NAME    DS    ØH .
         L     &REGR,X'224'             GET ASCB ADDR
         CLC   &RNULL,X'AC'(&REGR)      ANY JOBNAME ADDR
         BE    &JOBNFND                 NO
         L     &REGR,X'AC'(&REGR)       GET JOBNAME IF JOB
         B     &JOBFND                  YES
&JOBNFND DS    ØH .
         L     &REGR,X'BØ'(&REGR)       ELSE STARTED TASK OR TSO
         B     &JOBFND                  YES
&RNULL   DC    AL4(Ø)                   NULL ADDRESS
&JOBFND  DS    ØH .
         MEXIT
         MEND
```

ISPFBAT

The ISPFBAT procedure shown below is for running ISPF in batch.

```
//*
//*      PROCEDURE FOR RUNNING ISPF IN BATCH
//*
//ISPFBAT  PROC
//S1       EXEC PGM=INTSYSIN,
// PARM=' COPY INDD=SYSUT1,OUTDD=SYSUT2'
//SYSUT2   DD   DISP=(,PASS),UNIT=VIO,SPACE=(TRK,(1,1))
//*
//*  AVOID ENQ PROBLEMS ON ISPF PROFILE MEMBERS I.E MSG ISPTØ36
//*
//S2       EXEC PGM=IEBCOPY,REGION=8M
//SYSPRINT DD   DUMMY
//SYSUT1   DD   DSN=INST.ISPPROF,DISP=SHR INSTALLATION ISPF DEFAULTS
//SYSUT2   DD   DISP=(,PASS),UNIT=VIO,DSNTYPE=PDS,
// SPACE=(CYL,(1,1,45)),DCB=(*.SYSUT1,BLKSIZE=2792Ø)
//* DUMMY STATEMENT TO ALLOW DOUBLE REFER BACK TO SAME PASSED DATASET
//DUMMY    DD   DSN=*.SYSUT2,DISP=(OLD,PASS),VOL=REF=*.SYSUT2
//SYSUT3   DD   UNIT=VIO,SPACE=(CYL,(1,1))
//SYSUT4   DD   UNIT=WORK,SPACE=(CYL,(1,1)),DCB=KEYLEN=8
//SYSIN    DD   DSN=*.S1.SYSUT2,DISP=(SHR,PASS)
//*
//*  AVOID ENQ PROBLEMS ON SKELETONS
//*
//S3       EXEC PGM=IEBCOPY,REGION=8M
//SYSPRINT DD   DUMMY
//SYSUT1   DD   DSN=INST.ISPSLIB,DISP=SHR INSTALLATION SKELETONS
//SYSUT2   DD   DISP=(,PASS),UNIT=VIO,DSNTYPE=PDS,
// SPACE=(CYL,(1,1,45)),DCB=(*.SYSUT1,BLKSIZE=2792Ø)
//SYSUT3   DD   UNIT=VIO,SPACE=(CYL,(1,1))
//SYSUT4   DD   UNIT=WORK,SPACE=(CYL,(1,1)),DCB=KEYLEN=8
//SYSIN    DD   DSN=*.S1.SYSUT2,DISP=(OLD,DELETE)
//*
//ISPFBAT  EXEC PGM=IKJEFTØ1,DYNAMNBR=1Ø24,TIME=1439,REGION=ØM
//SYSPROC  DD   DSN=INST.CLIST,DISP=SHR
//         DD   DSN=INST.PARMLIB,DISP=SHR
//         DD   DSN=SYS1.SISPCLIB,DISP=SHR
//         DD   DSN=SYS1.SISPEXEC,DISP=SHR
//SYSHELP  DD   DSN=SYS1.HELP,DISP=SHR
//SYSUADS  DD   DSN=SYS1.UADS,DISP=SHR
//SYSLBC   DD   DSN=SYS1.BRODCAST,DISP=SHR
//SYSPRINT DD   SYSOUT=*
//SYSTERM  DD   SYSOUT=*
//SYSIN    DD   DUMMY
//ISPPROF  DD   DSN=*.S2.SYSUT2,DISP=(OLD,PASS)
//SYSTSPRT DD   SYSOUT=*
//SYSTSIN  DD   DUMMY
//ISPLOG   DD   DUMMY,DCB=(RECFM=VA,LRECL=125,BLKSIZE=129)
//ISPCTLØ  DD   UNIT=VIO,SPACE=(CYL,(Ø,1)),
// DCB=(LRECL=8Ø,BLKSIZE=8ØØ,RECFM=FB)
//ISPCTL1  DD   UNIT=WORK,SPACE=(CYL,(Ø,1)),   SKEL SUPPORT REQ REAL
```

```
// DCB=(LRECL=8Ø,BLKSIZE=8ØØ,RECFM=FB)         DEVICE AND DISTINGUISH
//ISPCTL2  DD   UNIT=WORK,SPACE=(CYL,(Ø,1)),   DSNAMES FROM SUBMITTING
// DCB=(LRECL=8Ø,BLKSIZE=8ØØ,RECFM=FB)         TSOUSER.
//ISPTABL  DD   DSN=INST.ISPTLIB,DISP=SHR      INSTALLATION ISPTLIB
//ISPPLIB  DD   DSN=INST.ISPPLIB,DISP=SHR      INSTALLATION ISPPLIB
//         DD   DSN=SYS1.SISPPENU,DISP=SHR
//ISPMLIB  DD   DSN=INST.ISPMLIB,DISP=SHR      INSTALLATION ISPMLIB
//         DD   DSN=SYS1.SISPMENU,DISP=SHR
//ISPTLIB  DD   DSN=*.S2.SYSUT2,DISP=(OLD,PASS)
//         DD   DSN=INST.ISPTLIB,DISP=SHR      INSTALLATION ISPTLIB
//         DD   DSN=SYS1.SISPTENU,DISP=SHR
//ISPSLIB  DD   DSN=*.S3.SYSUT2,DISP=(OLD,DELETE)
//         DD   DSN=SYS1.SISPSENU,DISP=SHR
//         DD   DSN=SYS1.SISPSLIB,DISP=SHR
//*
//*  DELETE DOUBLE REFER BACK DATASETS
//*
// IF (S2.RUN = TRUE      )
// THEN
//*
//S4       EXEC PGM=IEFBR14,COND=EVEN
//DD1      DD   DSN=*.S2.SYSUT2,DISP=(OLD,DELETE)
//DD2      DD   DSN=*.S2.DUMMY,DISP=(OLD,DELETE)   MUST DELETE
//*
//*  IF DATASET IS NOT EXPLICITLY DELETED, THE MSG
//*  IEC143I 213-Ø4 OCCURS AT DOUBLE INVOCATION OF THIS
//*  PROCEDURE IN THE SAME JOB WHEN DATASET IS SMS MANAGED.
//*
// ENDIF
//*
```

(Editor’s note: This article is now concluded.)

Nils Plum
Systems Programmer (Denmark)

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*, *SQL Server 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 which would be of interest to our readers, 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 on fionah@xephon.com.

From NetWare SAA to telnet with Reflection

It sounds so simple – NetWare SAA is no longer reliable, so change the transport type to telnet. After all, Reflection supports both. Unfortunately, our experience was that a lot of other changes were required before the user complaints died down to acceptable levels.

THE ENVIRONMENT

Just a year before the change, a four-month program began to replace the 5,000 workstations in the enterprise with Windows NT 4.0. Most had been MS-DOS 6 and Windows 3.11; there were a sizeable number of Windows 95 workstations, but virtually no NT. Despite the fact that Rumba was used for most terminal emulation, Reflection was chosen for Unix, VAX/VMS, and IBM mainframe hosts.

For the mainframe, Reflection for IBM for Windows NT Version 6.10 was initially installed, but new software installs are currently Version 6.20. NetWare SAA was used as the transport.

Access to the IBM mainframe is mostly TSO and IMS/TM. There are also a few CICS applications. Part of the program to standardize the desktop included replacing OfficeVision/VM (PROFS) with Microsoft Exchange and Outlook.

Just a few years earlier, the organization had gone through a very painful and time-consuming project to standardize the Network Operating System (NOS) to NetWare. Despite the fact that there were strong technical reasons to simultaneously move from NetWare to NT as a NOS, the previous negative NOS experience vetoed the idea.

SYMPTOMS OF THE PROBLEM

A year later, the frequency of Reflection errors on the mainframe went from mildly annoying to frustrating. Closely located groups of users were *each* reporting several errors per hour. These fell into three categories:

- Reflection was going into 100% CPU usage, usually during a

System Wait (clock symbol showing) while waiting for a host response.

- Connection errors were being experienced during a session. The host session would be lost, and reconnection (eg S beside Reconnect in TSO full screen logon) was failing because VTAM thought the Logical Unit (LU) was still attached. Users could either wait half an hour for the TSO inactive session timeout to occur, or wait on hold for the Help Desk who would then take several minutes to contact the mainframe console operator to cancel the host session.
- A few incorrect characters would be randomly displayed on the screen, but no error condition would be reported.

The third situation was the most alarming, since it wasn't clear what occurred during full-screen displays in the ISPF Editor and in similar situations in other environments. Would the incorrect characters be transmitted back to the host and unnoticed changes occur within the file being edited? Or would this occur only if other changes were being made to the same line because of the 3270 Modify Data Tag (MDT) protocol?

THE SOLUTION

It sounded so simple when the Help Desk proposed it: change the transport type from NetWare SAA to Telnet. The changeover was very carefully documented so that it could be taught to new mainframe users:

- Select Connection from the Reflection menu bar and Disconnect from the drop down menu
- Connection – session setup
- Look in the Transport area of the window
- Change the Type box to telnet
- Change the 'Host Name or IP Address' box to tn3270
- Click the OK button

- Select Connection – Connect
- Select File – Save to retain this setting for future sessions.

And when we were greeted by a somewhat different version of the company's standard log-on screen, there was every indication that it had worked. But those of us who had previously changed the default Reflection colour scheme ran into problems as we logged on to TSO for the first time. The background screen colour remained the same, but the colours displayed on that background had changed. In some cases, text was difficult to see because of poor contrast with the background colour, or was missing altogether because it was the same colour as the background!

It took quite a while for the mystery to be solved. Another division of the company has its own mainframe, and the few of us who had used both had already seen this problem.

There are two sets of colours that can be set in Reflection: monochrome, such as Protected Normal Alpha, and colour, such as Host Turquoise. In the past, to change the Reflection colour scheme, we had changed the monochrome set; now we had to change the colour set. Sometimes the colour changes seemed bizarre, like having Host White display in Black – necessary if the background has been changed from the black default to white or possibly even bright grey or dull white.

Moving from monochrome to colour also meant moving to support for the 3270 extended attributes. Unfortunately, it took a while to realize this fact and its implications, and even longer to come up with a partial set of solutions.

The first indication of a problem was when several programmers reported that there were now horizontal lines in previously blank fields, and that they wanted to get rid of them. After visiting each, it was determined that they were all using ISPF, and ISPF was using the underscore extended attribute to highlight certain fields.

Since there didn't seem to be any way to solve this problem in Reflection, it was reconsidered as an ISPF problem. Solutions were determined within ISPF, but, of course, that didn't address other environments. Once these solutions had been implemented, a CICS user heard about them, and asked to have the horizontal lines removed from her screen. Another attempt was made to solve the problem directly through Reflection – changing the Model ID from Model 2 24x80 Extended to *non-extended* – but this hung the VTAM connection and no one was able to access that particular remote CICS system because they kept getting the same hung LU.

ISPF SOLUTIONS

Most ISPF panels were cleared up from ISPF Option 0. On the Menu bar, Colours is selected, then 2 CUA Attributes from the drop down menu. This displays a CUA Attribute Change Utility panel, on which the colour, intensity, and highlighting of most panel elements can be set. There are quite a few panel elements, so the list is scrollable with PF8 (DOWN) and PF7 (UP). Any occurrences of USCORE in the Highlight field were changed to NONE. PF3 (EXIT) saves the changes.

Unfortunately, not all panel fields are controlled from here. For example, the Member List also has its own Colour Change Utility, accessible in any member list (including option 3.4) from the Functions menu bar item, by selecting drop down menu item 2 Change Colours.

Personally, I ran into a different problem. In the ISPF Editor, using the mouse with Reflection to select text for cutting or copying became confusing since the same shading was used by ISPF to highlight whatever it was I had been searching for with the FIND command.

In the Edit panel (ie when you are actually editing a member or sequential dataset, whether through option 2 or 3.4), select Edit from the menu bar, then 3 Hilite from the drop-down menu. At this point, you have a choice. You can remove the slash beside ‘Highlight FIND strings’ and not have the FIND string highlighted at all – a reasonable approach given that it had not been highlighted previously, as a monochrome 3270 without extended attributes.

Alternatively, if, like me, you find the highlighting a useful visual aid, you can select a different type of highlighting. From the Edit Colour Settings panel displayed as a result of menu item 3 Hilite selected above, select Colours from the menu bar, then 2 Find String Colour... from the drop-down menu. The current values are shown for colour and highlighting method to distinguish any occurrences of the string you last did a FIND on. In my case, I chose 1 Red and 3 Underscore. The Shading I had confused with Reflection's mouse selection of text was 2 Reverse. Hit PF3 (EXIT) to save the changes.

CAUTIONS AND CONCLUSIONS

One word of caution: do not change the ISPF Global Colours. However, the panel that would be used to change Global Colours is quite useful. It is also accessed from ISPF option 0 Select Colours from the menu bar, then 1 Global Colours from the drop-down menu. The resultant panel is titled ISPF Settings – Global Colour Change Utility.

At the bottom of the panel is a list entitled ISPF Default Colour. Each colour is listed on the left, and the fields on the right allow you to change the colour that ISPF displays instead. They should all be left blank since Reflection is the preferred method of doing colour mapping. But it's a good thing to have the panel displayed as you change Reflection's colours, since you have all of the colours that ISPF uses all on one screen.

In my own case, wanting a fairly bright background to eliminate screen glare, I chose bright grey or dull white as a Reflection background colour, rather than the default black. Within Reflection, I changed Host Blue to dull blue, Green to dull green, Turquoise to blue, Yellow to dull yellow and White to black. I left Red and Pink unchanged, but note that the colour labelled Pink by both Reflection and ISPF appears more like red-violet.

All in all, it was not an enjoyable exercise, but a few months later, the results are great. For the first few days, the telnet 3270 server was not as stable as it needed to be, but it is now better managed and very reliable.

Since then, NT has been announced as the replacement for NetWare. So we would have had to make the transition anyway, and the effort was not wasted.

Jon E Pearkins
(Canada)

Using packet-switching

The dramatic increase in the use of three packet-switching technologies for data communications has significantly increased interest in their use to carry intra-organizational voice. However, the costs and attributes of these technologies vary widely.

In the beginning, there were switched circuits, which were used for the very first telephone calls through manual switchboards, and are still used for dialled telephone calls. Then, customers demanded and telcos provided leased circuits, which were dedicated between two fixed points.

The earliest use of the public telephone company services to carry data communications involved both switched and leased circuits. However, as this type of traffic increased, it was carried primarily using leased circuits, often with the use of time-division and statistical multiplexers. Then, as the data communications industry developed, an increasing percentage of data was carried in packets, with packets to different destinations being carried in sequence, often through a single channel of a leased carrier circuit.

Although the cost of long-distance carriage has been steadily decreasing, it has not fallen nearly as much as the cost of computing. It is this differential that has increased the attractiveness of using computing to packetize, prioritize, buffer, and transmit packets of data and hence reduce the overall bandwidth, or capacity, of the leased circuits used to carry this data.

THREE SWITCHING TECHNOLOGIES

In recent years, three packet switching technologies have been widely deployed to carry data between sites in wide area networks: Frame Relay, asynchronous transfer mode (ATM), and Internet protocol (IP). And, as many large (in both geographic spread and total bandwidth) networks using these technologies have been deployed, they have increasingly been used to carry all of an organization’s wide area communications – and that includes voice.

Note that these technologies are not limited to the carriage of voice and data traffic for intra-organizational and inter-organizational communications. ATM is fast becoming the core switching technology for carriers and IP the network protocol for the Internet. But this article focuses on the carriage of intra-organizational voice; other applications are outside its scope.

That is not to say that the Internet is not and will not be used for corporate voice carriage. In *Internet Telephony Grows Up* (1997), Forrester Research found a strong interest in the potential for savings possible from the use of Internet telephony, but a reluctance to use it until issues about quality, reliability, manageability, and technology and service maturity were resolved.

FRAME RELAY

Frame Relay (FR) was designed as a carrier service to interconnect customer LANs at multiple locations. Before its introduction, LANs were mainly interconnected using leased E1 (2.048 Mbps) and T1 (1.544 Mbps) services, although both higher and lower bit-rate services were also used.

The primary disadvantage is that they are point-to-point services. This was not much of an issue for a network of a few nodes, but, for organizations with larger networks, such links required the construction of large networks, typically hybrids of mesh, star, and ring topologies, requiring high-capacity routers at major nodes, and a high level of management to minimize the occurrence of congestion. And the requirement for packets on some routes to travel through a number of intermediate nodes increased total network traffic and hence cost, and introduced additional delays to those packets. In addition, since there were, and still are, sizeable increments between the different capacities offered by carriers, users often had to lease capacity far in excess of what was actually required.

FR is a transmission technology that carries and switches variable-length packets at Layer 2 of the seven-level OSI model. It was designed to be offered by carriers to their customers to carry data between LANs. The FR design borrowed many aspects of the design of X.25, also a packet-switched technology ideally suited to the transmission of data at bit rates up to about 64 Kbps.

FR networks thus switch packets between access points using virtual circuits (VCs), which, according to the ITU-T standard I.122, can be either permanent (PVCs) or switched (SVCs). Each VC links an originating data link control identifier (DLCI) with a terminating DLCI, each of which identifies the VC to the Frame Relay access device (FRAD). The use of VCs ensures that packets between two DLCIs always use the same path through the network and thus arrive in the correct sequence, and the network element address, which is in fact a sequence number, verifies their receipt. An FR network therefore allows a user to access this network at a number of sites and, using PVCs from each point of access to each other, operate a virtual mesh network without any requirement to transit data through intermediate sites or a central hub, although fully meshed networks are not always the optimum configuration.

Since it was designed for the interconnection of LANs, another significant characteristic of FR is that it allows packets of up to 4,096 bytes of data. This large packet size maximizes throughput when large volumes of data are being transferred – the six-byte header comprises only 0.15% of a packet that is carrying 4,096 bytes of data. LAN traffic also involves large numbers of much smaller transmissions, which, because of FR's variable packet length, are carried more efficiently than they would be using technology employing fixed-length packets. And, because of the low processing overhead, FR is generally agreed to remain efficient at bit rates of up to about 2.048 Mbps.

FR networks are primarily public networks operated by carriers and to which customers subscribe. Users lease an access to the FR network at each of their sites at a cost which is a function of the bit rate of the local access (the 'access rate', AR). The user also leases PVCs between pairs of sites. Each PVC has a committed information rate (CIR), the minimum bit rate that is guaranteed to be carried by the network without packet loss. PVCs are tariffed on the CIR, and, for some carriers, on distance.

Accesses to public FR networks often have CIRs on the PVCs, which, when totalled, are well below the AR of the access. The FRAD, the customer's interface to the FR network, may thus transmit data on one or more PVCs in excess of the CIR. However, when there is a peak in traffic within the network, packets will be discarded, although the exact mechanism varies widely from one carrier to another – the FR specification I.112 has defined the discard eligibility (DE) bit which allows a FRAD to specify and a FR network to identify which packets can be discarded and which cannot. The level of packet discard needs to be monitored, in order to set the CIR on each PVC to carry the traffic at a minimum cost without too many packets being discarded.

FR development took a big step forward with the establishment, in 1991, of the Frame Relay Forum, which now has over 300 members, including computing and data communications suppliers. Although FR networks also use ITU-T standards, the Frame Relay Forum has been promoting the deployment of FR as well as developing standards for its use. These standards include those for the transmission of voice through FR networks.

On a macroscopic level, voice traffic is bi-directional, non-bursty, and uses a constant and well-known bit rate. On a microscopic level, however, it consists of pauses and drawn-out or redundant sounds as well – analysis by the Frame Relay Forum suggests that only 22% of voice communication is actually distinct information. The removal of redundant or repetitive sounds and pauses requires digital speech interpolation, a form of non-linear compression, and the recreation of these repetitive sounds and pauses when the analogue voice is recreated at the receiving end. FRADs must use both digital speech interpolation and echo cancellation, necessary if the delay exceeds 25 msec, and many also support voice compression.

The Frame Relay Forum implementation agreement (IA) FRF.11, ratified in May 1997, defines two classes of voice compliance. Class 1 compliance uses ADPCM (adaptive differential pulse code modulation), compressing voice to 32 Kbps according to ITU-T standard G.727. Class 2 compliance uses CS-ACELP (conjugate-structure algebraic-code-excited linear-prediction) to compress voice to 8 Kbps according to either of the ITU-T standards G.729 and G.729A (note that the latter is of a lower quality). FRF.11 caters for the transparent transmission of Group 3 fax traffic by transmitting such traffic not as compressed voice, which may prevent the signal from being satisfactorily reconstructed, but by converting it back to digital traffic for transmission.

Just as for the other technologies, the biggest issues for the transmission of voice through FR networks are delay and jitter. The process of packetizing and queueing traffic for transmission through such a network inherently introduces transmission delay, especially if the same network is also carrying LAN traffic. Jitter, inconsistent delay, can result from the varying levels of both voice and non-voice that occur from one second to the next.

FRADs minimize delay using prioritization and fragmentation. The FR specifications do not include a priority indicator, and frames are routed through an FR network in order of receipt. The FRAD implements prioritization by having two or more buffers, one for each level of priority, and preferentially transmitting frames according to the priority of the buffer. Voice traffic, video traffic, and any other delay-sensitive traffic are assigned the highest priority.

Fragmentation (or ‘segmentation’) counters the delay caused by transmitting very large data packets. Fragmentation breaks such packets into smaller packets, thereby preventing any one packet from significantly delaying voice packets, which are typically only 128 bytes long. The exact fragmentation method has, to date, been proprietary to the FRAD manufacturer, preventing different FRADs from being able to interwork, but the Frame Relay Forum’s implementation agreement F.12 prescribes a standard fragmentation method.

FRADs attempt to eliminate jitter by the use of a jitter buffer. This holds incoming packets as they arrive, releasing them to the DSP, which, in turn, recreates the voice signal, after a delay which varies so that the delay each packet experiences from FRAD to FRAD is consistent.

Although the FR specifications do not include a priority flag as part of the frame header, this header does have the discard eligibility (DE) bit. As it cannot tolerate the disruption introduced by this process, the

DE bit must not be set for voice traffic. Data applications such as file transfer, by contrast, which are not delay sensitive, use higher-layer protocols which can request the transmission of discarded packets and thus reconstruct the original transmission. So, when the volume of traffic exceeds the CIR, the DE must be set for such lower priority traffic so that voice packets are not discarded.

Congestion within an FR network can be signalled to sending FRADs using the BECN (backward explicit congestion notification) bit, and to receiving FRADs using the FECN (forward explicit congestion notification) bit. FRADs receiving the BECN bit should be able to reduce their transmission volume.

ASYNCHRONOUS TRANSFER MODE

Asynchronous transfer mode (ATM) has been under development for about as long as FR, with this development coordinated and promoted by the ATM Forum and, to a lesser extent, the ITU-T. It was developed largely from the telecommunications industry, and, as a result of substantial carrier input, is targeted at all types of traffic and designed to operate at bit rates up to a gigabit (ATM networks operating at 155 Mbps). ATM defines both access interfaces to an ATM network and the network architecture. As with FR, the ATM specifications define both PVCs and SVCs.

ATM switches cells which have a fixed length of 53 bytes, comprising a five-byte header and 48 bytes of data. ATM is thus more likely to be used as the backbone network for large corporations and government departments, and is favoured by carriers themselves, some of whom have implemented ATM networks to carry both ATM and FR traffic for their customers.

As ATM has been designed to carry voice as well as data, it has some inherent advantages. First, having a relatively small and fixed packet length eliminates the problem of voice being delayed by large data packets and means that customer ATM equipment does not have to implement fragmentation.

ATM switches and multiplexers nevertheless use prioritization to minimize the delay of voice traffic. They can also use traffic shaping

ITU-T service categories	ATM Forum service categories
Deterministic bit rate (DBR)	Constant bit rate (CBR)
Variable bit rate (VBR-RT)	Real-time variable bit rate (rt-VBR)
Statistical bit rate (SBR)	Non-real-time variable bit rate (nrt-VBR)
(No equivalent)	Guaranteed frame rate (GFR)
Available bit rate (ABR)	Available bit rate (ABR)
(No equivalent)	Unspecified bit rate (UBR)
ATM block transfer (ABT)	(No equivalent)

Figure 1: ITU-T and ATM service categories

procedures to regulate the rate of cell transfer into the network. Other factors impacting the transmission of voice through ATM switches are buffer size (more is better), buffer allocation (large output or pooled buffers are preferable), and the use of a matrix instead of a ring or bus for internal cell switching.

Unlike FR, ATM was originally conceived as a multi-service technology. The ITU-T definition includes the ATM Adaptation Layer (AAL) to enhance the services provided by the ATM layer. Within this are a number of standards, with AAL1, AAL2, AAL3/4, and AAL5 the most widely supported. There are also five classes of service, A to D and X. But because these do not provide the complete specification for the carriage of different traffic types, both the ITU-T and the ATM Forum have devised and issued sets of service categories. The ITU-T's are in recommendation I.371, 'ATM Transfer Capability', and the ATM Forum's in 'TM4.0, ATM Service Category' (see Figure 1).

When a VC is established, the AAL and bit rate are requested for that particular circuit by the subscriber and granted by the provider. Those of PVCs are established when they are requested; those of SVCs are requested at call set-up.

Early implementations of voice over ATM used constant bit rate VCs for voice. However, as this service category provides a circuit emulation service (CES), which requires the provider to allocate a fixed amount of bandwidth for the duration of the call, CBR VCs monopolize bandwidth and don't use it efficiently. Variable bit rate has therefore become the service category of choice for voice over ATM, not just for its efficiency, but also because it allows voice from more than one channel on the ATM multiplexer to be inserted into a cell. It also supports voice compression and silence suppression.

Each access to an ATM service has an access rate (AR), and each VC has the sustainable cell rate (SCR), equivalent to FR's CIR. Each cell has a cell loss priority (CLP) bit; when the volume of traffic on a given VC exceeds the SCR, the carrier may discard the excess traffic. As the cells for which the CLP bit has been set will be discarded first, it should not be set for cells carrying voice.

Both ATM and FR have been designed to be implemented by carriers and offered to customers via points of access, and most non-niche carriers offer both. ATM is also designed to be implemented by organizations as their own private network backbone, although it is prohibitively expensive for most.

INTERNET PROTOCOL

Although the least suited to carrying voice of the three packet-switch technologies discussed here, the sheer size of IP's installed base and its use as the network protocol for the Internet have generated a significant demand for its use to carry voice.

Unlike ATM and FR, the applications for voice over IP (VoIP) networks fall into three groups:

- Carriage of voice through an organization's private IP network in addition to data, for which it was most likely to have been constructed.
- Carriage of voice through the Internet, from the organization's contacts into its contact centre or through ISPs providing an interconnect service.
- Carriage of voice between terminals of a LAN-based telephone system within the organization's site.

Note that the characteristics, cost benefits, and applications of transmitted voice through the Internet (and using IP) are very different from those for transmitting voice through a private IP network.

Most organizations currently operate private data networks using routers and IP, which have typically been installed to function as a backbone data network onto which as many data communications applications as possible are migrated. Voice can be incrementally added to such a network without incurring major capital expenditure.

Cost is significant because a major attraction of carrying voice through a private network is to eliminate long-distance call charges. But cost-effectiveness depends on a number of factors, including long-distance tariffs, leased-line tariffs, spare capacity in the existing network, and call volumes.

Savings are achieved not just from having voice piggy-back a data network; the process of packetizing voice achieves the same savings from silence suppression as achieved with FR, and most VoIP products also use compression. Most products use ITU-T recommendation G.723.1, which compresses voice to either 5.3 or 6.3 Kbps, although some use G.729, which compresses it to 8 Kbps. Only G.729 can reliably transmit DTMF tones.

Putting voice over a private IP network requires the use of gateways, which are typically either a device built specifically for this application, a PC chassis with VoIP cards inserted, or a function of a router.

Using a private IP network to carry voice is not without its problems, the most significant of which is delay. This is introduced by the packetization process itself, with most IP gateways requiring about 32 milliseconds to packetize 30 milliseconds of voice, with the destination gateway requiring another 32 milliseconds to perform the depacketization. As this happens in each direction, packetization and depacketization introduce a round-trip delay of about 170 milliseconds.

Delay is also introduced by the resequencing buffer. As IP has no equivalent of VCs, IP packets don’t necessarily follow the same path through the network and may arrive out of order. Received packets are held in this buffer to resequence them and remove jitter before reconstructing voice, which adds 50 milliseconds. Then there are the delays introduced by encoding, decoding, and the compression process itself, which may be another 200 milliseconds. The total round-trip delay is thus in the order of 500 milliseconds, close to that for a satellite link.

As with FR, large IP packets carrying data can delay voice packets, and some routers employ fragmentation. However, because of the size of the IP header, this significantly increases overall traffic, reducing efficiency.

The IP header contains no priority or discard eligibility bit, so the gateways have to perform prioritization. Most routers can do this, but if non-router voice gateways are used, they are adding voice packets to a LAN and cannot prioritize them. In this scenario, the gateways could be connected to their own voice-only LAN, which, with the data LAN, is connected to a switch, which gives priority to traffic from the voice-only LAN. However, although such methods give IP packets carrying voice priority when entering the network, once these packets traverse the network and pass through intermediate routers en route to their destination, there is no identification that they are intended to be priority packets. Some suppliers therefore implement proprietary header settings to achieve this result.

One attempt to address this led to the RSVP protocol, which allowed the sender to request a certain capacity at each intermediate router. But RSVP implementations rarely achieved this, and it has not been widely adopted. More recently, the Internet Engineering Task Force's (IETF's) Intserv working group has developed the (as yet unratified) Differentiated Services Model which uses the IP header's Type of Service octet field to classify traffic.

H.323

H.323 is a standard of the ITU-T for packet-based multimedia communication systems. It defines a set of channel set-up, call control, and codec specifications for transmitting real-time voice, video, and data over networks, such as IP networks, that do not guarantee delivery of quality of service.

H.323 itself references several other ITU-T recommendations, which collectively define four network components which interoperate to provide multimedia communication with other compliant components. Because of the high level of interest and uptake of VoIP, most applications of which must inherently interoperate, H.323 has been enthusiastically supported by suppliers of VoIP-capable products.

Version 1 was ratified in March 1996 and Version 2 in February 1998. Products claiming compliance with Version 1 must comply with all of the mandatory requirements of the 1996 version of H.323 and the 1996 versions of H.225.0 and H.245. To claim compliance with Version 2 of H.323, a product must comply with all of the mandatory requirements of the 1998 version of H.323 as well as the 1998 versions of H.225.0 and H.245.

H.323 allows a number of optional capabilities, none of which are required in order to claim H.323 compliance, and hence will actually work only if supported by both endpoints. One result is that calls between two endpoints that are H.323 compliant, but that comply with only the basic requirements of H.323, will deliver only the same functionality as a standard telephone call.

The H.323 standard defines four components: endpoints, gatekeepers, gateways, and multipoint control units (MCUs). Endpoints (the only component required in all H.323 networks) are the clients or terminating units on an H.323-compliant network, and are typically IP telephones, voice cards in PCs, or video codecs. H.323 references the G.711 standard for voice compression (although the VoIP consortium recommends G.723.1). Once the connection, or socket, between sender and receiver has been set up using a protocol such as TCP, it is the responsibility of real-time protocols such as H.225 or RTP, both referenced by H.323, to transmit traffic.

The optional gatekeeper is used for:

- Routing, and alternate routing if an endpoint is congested or unavailable.
- Allocating bandwidth by refusing additional connection requests above a threshold.
- Providing address translation from LAN aliases for terminals, and gateways to IP addresses.
- Providing call authorization and call accounting.
- Performing network management.

Larger H.323 networks may be divided into zones, each of which requires its own gatekeeper.

An H.323 MCU controls a conference call involving three or more endpoints. Both endpoints and MCUs can initiate such conference calls.

Gateways interface H.323 networks to non-H.323 networks such as the PSTN and ISDN, and to non-H.323 terminals such as analogue handsets, PABXs, and key systems. To perform this function, gateways provide the protocol translation, multiplexing, compression, decompression, and common channel signalling messages as required.

As comprehensive as it is, H.323 does not address the issues of resource reservation or quality of service (QoS) control, and hence does not guarantee any QoS in terms of delay or non-loss of packets, although it does guarantee that, even with congestion, some voice and data will get through. Some suppliers are, however, putting QoS request/negotiation/enforcement mechanisms in place using the RSVP and ISSLL specifications at the network layer.

H.323 has not escaped the attention of the ATM Forum, which has developed the Realtime Multimedia over ATM (RMOA) specification that defines how ATM's variable bit rate (VBR) service category interfaces with H.323.

INTERFACING TO THE PABX

Whichever of the technologies discussed here is used, the issue of interfacing the gateway or multiplexer to PABXs must be considered. Although it is quite possible with both FR and ATM to allocate a PVC to two specific PABX channels, one at each end, this is not actually done in practice. With ATM, one VC is typically allocated to voice between each pair of PABXs; with FR, one VC is typically allocated between each pair of sites. Some carriers recommend a single VC for all voice and data traffic; others recommend one VC for voice and another for data. IP does not have VCs, so the packets are simply sent.

As the gateway or multiplexer will perform the call routing, it must be able to understand the protocol used by the PABX to signal dialled digits. This is usually not an issue. However, as a trunk or channel on one PABX is no longer associated with a specific trunk or channel on another specific PABX, in order to achieve PABX-to-PABX extension feature transparency, a separate signalling path must be supported and the destination PABX must be able to match the signalling to the incoming call - not all of them can.

Another problem could occur when the circuit into the local gateway is idle but there are no idle circuits on the destination PABX. The local gateway will accept the call, but the ‘busy’ condition will be detected only when the call is routed through the network to the destination gateway. Because the local gateway has accepted the call, the originating PABX cannot alternate-route the call through the PSTN. The caller hears an engaged tone, and must dial a different telephone number. This is being addressed by the H.450.3 standard referenced by H.323. Some PABXs can, through the signalling channel, look ahead to the destination PABX to ascertain an available circuit – check with your vendor.

THE FUTURE

All three packet switching technologies have sizeable installed bases and are enjoying very healthy growth. In early 1998, IDC estimated that IP telephony worldwide represented US$1.9 billion a year for 5.8 billion minutes of use, and forecast IP telephony to expand to $24.4 billion and 151.7 billion minutes in 2002, representing 11% of total call volume. But by mid-year, it had revised its estimate to between 20% and 25% of total call volume. In *The Future of Broadband Networking* (1997), Ovum forecast that the IP market would grow at more than 70% annually between 1998 and 2002.

In *US Packet/Cell-Based Services Market Share and Forecast*, 1996-2001, IDC forecast FR to grow annually at 46%, from $2.2 billion in 1997 to $7.6 billion in 2001. Over the same period, ATM is forecast to grow from $230 million to $2.0 billion, a CAGR of 92%. ATM’s share of the combined market is thus forecast to grow from 9.5% to 20.5% over this period. Ovum forecast the ATM market to grow around 100% annually between 1998 and 2002. Note that the figures for FR and ATM are, however, for the market as a whole, not for the use of these technologies for carrying voice.

IP is so ubiquitous and so inexpensive that its future is assured. In *The Future of Broadband Networking*, Ovum forecast that, by 2002, the use of TCP/IP will exceed that of all other protocols. Given this market presence and the non-private network applications for VoIP, this technology will be increasingly developed and deployed by both carriers and organizations of all sizes throughout the developed world.

ATM addresses a very different sphere of the market, and, given the basis of its development, will increasingly become the core switching technology of carrier networks. Several carriers, including AT&T, BT, and Bell Canada, are no longer buying circuit switches for their transit switching networks.

As a voice transport technology, FR occupies a compromise position. Nevertheless, it will continue to have a market presence for some time yet, and many FR networks will continue to carry voice.

Stephen Coates
(Australia)

Information Point – reviews

This on-going series of Information Point articles looks at where else you might go to supplement the information you find in each issue of *TCP/SNA Update*.

VM/ESA TCP/IP PERFORMANCE

The speaker's notes and overheads from three excellent presentations are available at

http://www.vm.ibm.com/dEVPAGES/Bitner/presentations/tcpip

All three focus on TCP/IP performance in the VM/ESA environment.

Two include charts of benchmarks that demonstrated higher throughput with TCP/IP than SNA.

The author is IBM’s Bill Bitner, a frequent speaker at SHARE. Bill’s home page is at

http://www.vm.ibm.com/dEVPAGES/Bitner

and includes links to other VM-related information, as well as a caricature of Bill, complete with typical dialogue: “Why does VM perform that way?” “It depends ...”

NPPS

The Network Professionals of Puget Sound (NPPS) is a Seattle-based group that split from the Network Professional Association, and maintains a virtual bookstore through Amazon.com. It provides its own short reviews of most of the books it sells. Books are grouped by publisher, but there’s also a search feature that takes you directly into Amazon.com.

You can get there through the NPPS home page at

http://www.npps.org

by choosing ‘Bookstore’ from the left sidebar. Or, if you feel the frames approach takes up too much screen real estate,

http://www.npps.org/bookstore/main.html

gives you a frames-free view.

MORE BOOKS

Another Amazon.com-based bookstore can be found at

http://a-ten.com/books.

There are few reviews here, mostly just lists of books on specific topics. Clicking on the book title takes you to the book’s listing in Amazon.com, which often includes a review.

From the left sidebar of the home page, under the ‘Computers’ title, click on TCP/IP and you will see a large list of book titles. However, the copyright at the bottom of the page suggests that the list has not been updated since 1998, and there are no listings for books published in 1999 or 2000.

THE...FORMERLY KNOWN AS VTAM

If you still can't get used to calling VTAM by its new name, SecureWay Communications Server, you may not have found its home page, at

http://www.ibm.com/software/network/commserver

A link on the word SecureWay takes you to the SecureWay home page at

http://www.ibm.com/software/secureway

Selecting 'Support' from the left sidebar gives you links to:

- Downloadable product code fixes.
- The on-line technical database, where you can search for tips and fixes for known bugs.
- Newsgroups.
- Problem report submission.
- Support options.
- Supported versions and end-of-service dates.

All this from a single Web page.

Selecting 'Library' gives you links to:

- White papers
- Manuals and brochures
- Presentations
- Newsletters.

Selecting 'Case Studies' displays links to details of actual customer and business partner experience using the products around the world. Some are in the form of press releases, while others are in a more traditional case study format.

'More information' provides links to platform-specific versions of the product, where improvements made by the most recent release are listed.

As these pages indicate, SecureWay Communications Server is not just SNA (VTAM), but also includes TCP/IP. Note that, apart from a quick mention on the home page under the heading 'Operating systems', VSE and VM seem conspicuously absent from these pages.

TEK-TIPS

http://www.tek-tips.com was built to provide free, non-commercial technical forums. There are a large number of different forums, most dedicated to a specific hardware or software product, though there are also some on management issues and communications protocols. Some are quite popular, but others are not.

Topics are divided into four categories:

- Software
- Hardware
- Desktop systems
- Corporate applications.

However, it can be hard to figure out where to go at the next level. For example, TCP/IP is under 'Hardware', then 'Wiring Closet' and 'Communication Protocols'.

The 'Find a Forum' search box on the home page solves that by listing any forums that match your search string. Selecting one of the forums shows you, at the top of the page, how you could have got there through the hierarchy. So, in the case of TCP/IP, it reads 'Top -> Wiring Closet -> Communications Protocols'.

There are some features beyond the discussion threads within forums that deserve attention. Most forums show the results of a poll among members as to what they think of the product or technology. For TCP/IP, 33% liked it, 67% were neutral, and nobody hated it. However, there’s no indication of how many people actually voted, and those percentages could of course result from as few as three votes.

Links

From a forum, clicking ‘Links’ in the black title bar for the forum displays a list of links to other sites. Some topics, including TCP/IP, have no topic-specific links, but the ‘General Industry Links’ that follow are an interesting mix of material. This situation may improve over time, with members encouraged to add their own links.

To access links directly, without going to the forum, click on ‘Link Library’ near the top of the home page. The same hierarchy as that used for Forums is used, but there is no search function. You could, of course, use the Forum search function to get directly to the forum, and then use its Links link. Alternatively, you could determine where in the hierarchy you want to be, write it down or start up a second browser window, and navigate the Link Library hierarchy in that way.

Overall, the topics have a decidedly non-mainframe emphasis. But there are still some topics beyond TCP/IP that fall within our subject area. Here are a few:

- Microsoft SNA Server – little discussion, but a fair number of links.
- Novell SAA – no discussion, but a few links.
- Tivoli TME10 – some discussion and a few links.
- HP OpenView – some discussion and a few links.
- Attachmate – little discussion and a few links.
- Frame relay – some discussion, but no links.
- Token ring – little discussion and no links.

MESSAGING

The Business Quality Messaging (BQM) forum was founded in 1997 by AT&T, Compaq, IBM, Intel, and Microsoft. Today, there are almost 40 member organizations. You can find it at

http://www.bqm.org

Select 'Members', then 'Member Organizations' to see the full list.

Oddly, it is the 'About' link that has the most information to offer:

- Mission statement.
- Fact sheet.
- A lengthy white paper from Sept. 1998 by the Burton Group.
- Three sets of detailed messaging specifications, including one for ISPs.

The 'News' link provides access to BQM's recent press releases. And don't overlook the 'Related Links' link on the left sidebar of the News page – it lists eight great sites on messaging.

Finally, although the 'Events' link does list BQM meetings, its main use is the list of upcoming conferences covering messaging topics.

MQSERIES

MQSeries is IBM's middleware product in this area, and is covered by Xephon's own *MQ Update*. There's additional information on the Xephon Web site at

http://www.xephon.com/mqupdate.html

The 'MQSeries news' link lists recent product announcements, and 'MQSeries links to explore' provides an annotated list of interesting Web sites. Under the 'Visitors' heading, 'More information about MQ Update' links to a table of contents of recent issues and details on how to subscribe.

But perhaps the most important feature for subscribers is the (password-protected) ability to download source code from all issues of the journal.

TCP/SNA UPDATE

This journal has similar information on the Xephon site, including topical Web links. The site also offers a complete issue of a past edition for on-line viewing through Adobe Acrobat. And articles containing code from before 1998 are available for browsing, either by date or title of article.

BASIC NETWORK EDUCATION

Lamar University (Texas) Professor Dr Donald L Jordan makes available detailed material from his *MIS 334 Telecommunications Fundamentals and Network Applications* course at

http://mis334.lamar.edu

The ‘Lecture’ link gives class notes for all 30 lessons, in enough detail to be used directly as a self-study course.

The ‘Handouts’ link gives the same overheads Dr Jordan uses in his classes. They are divided into seven topics, with a table of contents for each, should you wish to get directly to an individual slide, rather than going through them sequentially.

Jon E Pearkins
(Canada)

TCP/SNA news

Better On-line Solutions has announced Version 5.0 of its BOSaNOVA TCP/IP TN5250e emulation software, which provides Windows users on a local or remote IP network with AS/400 connectivity.

For further information, contact:
Better On-line Solutions, 7762 E Gray Road, #300 Scottsdale, AZ 85260-6957, USA.
Tel: (602) 596 8332.
URL: http://www.bosWeb.com

* * *

William Data Systems has previewed Version 1.1 of its FTPalert, promising to overcome the major integrity and control problems that arise when TCP/IP's File Transfer Protocol is used to transfer data to or from OS/390 mainframes.

The company has also announced version 3.3 of its Exigence network diagnostic tool for OS/390.

For further information, contact:
William Data Systems, Rosemullion House, Holmesdale Road, South Nutfield, RH1 4JE, UK.
Tel: (01737) 822342.
URL:http://www.willdata.com

* * *

InfoExpress has announced the FireWalker virtual private networking suite for secure remote access to corporate user communities.

For further information, contact:
InfoExpress, 425 1st St, E Los Altos, CA 94022-3674, USA.
Tel: (650) 969 6924.
URL:http://www.infoexpress.com

* * *

ICL is to incorporate NetManage TCP/IP connectivity software in its Fujitsu TeamPad 7100 handheld computers following a worldwide agreement.

For further information, contact:
ICL, ICL House, 1 High Street, Putney, London, SW15 1SW, UK.
Tel: (0181) 788 7272.
URL:http://www.icl.com

* * *

Hewlett-Packard has announced its CIFS/9000 Common Internet File System for HP-UX 11.

For further information, contact:
Hewlett-Packard, 3000 Hanover St, Palo Alto, CA 94304, USA.
Tel: (650) 857 1501.
Hewlett-Packard, 1 Cain Road, Bracknell, Berks, RG12 1HN, UK.
Tel: (01344) 360000.
URL: http://www.software.hp.com

* * *