



30

TCP/SNA

June 1998

In this issue

- 3 Internet security
 - 8 Call for papers
 - 9 Programming in TCP/IP with REXX
 - 11 FTP under MVS in batch mode
 - 12 A mailbox system for SMTP under MVS TCP/IP – 2
 - 31 Implementing a Wide Area Network
 - 34 Link between NCPs – part 2
 - 53 Variable Bit Rate Voice over ATM
 - 60 TCP/SNA news
-

© Xephon plc 1998

update

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: t_eddolls@compuserve.com

North American office

Xephon/QNA
1301 West Highway 407, Suite 201-405
Lewisville, TX 75067
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>; you will need the user-id shown on your address label.

Editor

Trevor Eddolls

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 £125.00 in the UK; \$185.00 in the USA and Canada; £129.50 in Europe; £33.00 in Australasia and Japan; and £132.00 elsewhere. In all cases the price includes postage. Individual issues, starting with the March 1991 issue, are available separately to subscribers for £32.00 (\$47.00) each including postage.

© Xephon plc 1998. All rights reserved. None of the text in this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, without the prior permission of the copyright owner. Subscribers are free to copy any code reproduced in this publication for use in their own installations, but may not sell such code or incorporate it in any commercial product. No part of this publication may be used for any form of advertising, sales promotion, or publicity without the written permission of the publisher. Copying permits are available from Xephon in the form of pressure-sensitive labels, for application to individual copies. A pack of 240 labels costs \$36 (£24), giving a cost per copy of 15 cents (10 pence). To order, contact Xephon at any of the addresses above.

Printed in England.

Internet security

Internet security is a vital requirement for companies with a permanent Internet connection, whether the connection is primarily for e-mail, or for giving staff access to the external World Wide Web, or for running an in-house WWW server. This article outlines the need to be comprehensive in your security, and above all to actively test it, no matter how extensive and expensive your current Internet security is.

SECURITY IS DIFFERENT

Security systems differ from all other IT systems in that the users cannot act as testers. It is not uncommon for software vendors to release products that have a number of flaws, and the user community act as the testers. Typically, whenever any of your core IT systems in-house fail, the users will often be the first to spot the problem, and will clog your Help Desk with calls saying the database is slow or that the network to the USA is down! You can then fix the problem.

However, users cannot act as testers of your security – users will not be aware if there are major security vulnerabilities in your systems because, as far as they are concerned, things are still working.

For this reason, it is vital to take especial care over the implementation of security systems, and to plan for active testing.

The second principle to keep in mind is not to make assumptions about the security of your systems. It is all too easy to think, for example, that because work has just been done on applying the latest patch to your Internet e-mail server, there are no other security issues with it

NO PERIMETER SECURITY

Many organizations start by having no perimeter security at all – this was certainly the case a couple of years ago with most Internet connected sites. Many users, however, quickly realized that the costs associated with an Internet security incident could be high – bad PR,

the staff time wasted in re-installing all the company Internet systems (since any one may potentially have been corrupted by attackers), and the costs of not having Internet access for days or weeks while a problem was fixed. Companies then began to go down the road towards securing their connection.

FIRST STEP TO SECURITY

Often the first step to security means applying some filtering rules to the Internet router, between the Internet and their internal network. Virtually every site already has an Internet router, so in hardware terms this is a low cost option! Unfortunately, it is also often presented as the ‘best option’ by consultants, some of whom will make claims that it can do everything that a full firewall can.

Router filtering is better than nothing, but still leaves the site exposed to many commonly used attacks that are impossible to prevent using just a router.

STRONGER PROTECTION

The best-practice perimeter security is the use of a firewall.

Firewall definition

Note that the word firewall is often used erroneously to describe routers that have had some packet filtering applied. Such packet filtering is better than no network protection at all, but is inadequate to provide best-practice security. A firewall is defined as a network gateway device, which controls the flow of traffic in and out of the organization from the Internet, by running checking and control systems at the application layer – not simply at the packet layer.

Note that a packet filter router runs at the packet-by-packet level. It can run a series of fixed ‘rules’ that are static, and they can only make decisions based on a per packet basis, using only the information within that packet – ie only the source and destination address, and the port numbers and protocol etc.

A firewall however, by running at the application layer, understands

the nature of the Internet protocols, so is able to control access on a historical basis (once a certain type of traffic connection is established, the firewall can allow that traffic to continue within certain boundaries), and with better handling of protocol issues.

Additional firewall benefits

In addition to the enhanced security, a firewall can also allow:

- Logging of activity types at the protocol (not the less useful per-packet) level.
- Alarms – certain traffic types can be configured to trigger alarms, allowing immediate staff attention.
- Virtual Private Network (VPN) – automatic encryption of traffic between two or more sites linked only over the Internet.
- Network Address Translation – hiding local IP numbers (illegal or otherwise) by translating numbers on-the-fly.

Necessary but not sufficient

Whilst a firewall is a necessary component of a best-practice secure implementation, it is not sufficient alone. Certain attack types can be carried through normal SMTP e-mail or HTTP Web packets, in the normal data content, and impact the e-mail or WWW servers behind the firewall – such content-driven attacks will pass unstopped through even a theoretically perfect firewall.

This means that Internet-visible servers must also be secure, and it is also necessary to secure routers, and manage the flow of DNS information from the site so as not to provide host lists to external attackers.

DESIGN CHOICES

There are two basic design philosophies for firewalls – the ‘proxy’ and the dynamic filter approach.

In the proxy approach (also known as ‘application relay’), all traffic is re-written in the firewall so that there is no through traffic;

application layer ‘proxy’ modules in the firewall perform this task.

In the dynamic filter approach (also known as stateful inspection), the firewall uses the speed of the static-filter (router style) technique, but this is overlaid with dynamically-allocated rules.

The main benefit of the proxy approach is that it is, from a purist design perspective, more secure – no packets are ever transferred through, and thus it intrinsically fails safe if the proxy modules fail, and attack types involving corrupted packets are ineffective, because the re-written packets are cleaned-up.

FIREWALL CERTIFICATION

The US-based NCSA compiles a list of products defined as firewalls that pass their basic test suite; currently there are about 20 listed. In addition, there is a more stringent certification process – the UK, Germany, France, and Netherlands supported ITSEC scheme. However, many of the 20 NCSA listed products have little market share or history. Since some of these ‘also-ran’ products will be difficult to support into the future because of their rarity, and will probably also lag behind in development, it is sensible to choose one of the main products, and for governmental sites, or those with governmental links, an ITSEC approved device is mandated.

WHY TEST?

It must be born in mind that even the most secure firewall product cannot protect a site when the Internet router is incorrectly configured, or alongside insecurely configured Internet applications such as SMTP or DNS. The organization’s perimeter defences are therefore a complex combination of router, firewall, and Internet servers – with plenty of scope for configuration errors or software bugs to invalidate the defences. It is vital to test your security systems once they are in place – or better still, before you make your final security rollout – to discover any holes that you were not aware of, so they can be fixed as part of your security activity.

HOW TO TEST

There are a couple of routes that can be used to test your site, the first choice being whether to use in-house or external testers. In-house testing requires some level of security skills, and sufficient time should be allowed firstly for planning and designing the test systems, and then for performing the testing, and lastly for translating the raw data output into useful reports that can be used as the basis for action and further security planning.

Testing tools such as Satan and ISS are available and are often the first port of call for those planning in-house testing, but they do require careful use and neither provide an 'all-in-one' test suite, so a considerable amount of engineering is likely to be needed and should be built into the planning.

External testing on the other hand makes use of a 'fresh eye', which may spot problems hidden by the implicit assumptions made by those on the 'inside'.

In addition, external testing, from IP ranges not part of your own allocation, ensure that the results are not distorted by being taken from a part of your own network.

CHOOSING A TESTING HOUSE

There are two kinds of external test houses – the consultancy-based approach and the engineered service approach.

Consultancy-based testing will be charged on a per day basis, and for a complete site-wide analysis five to ten days' duration is typical. Some agencies such as the UK government-linked DRA may carry some prestige and be desirable, but other consultancies, whether large or small, may better fit your needs. Be sure that the organization used can provide the right level of 'on the ground' testing, and is not simply an 'overview of security' type house.

Engineered testing is not based on a per-day chargeable basis, but on a defined and pre-agreed specification, set with a fixed price. The end user can predict the costs, and can evaluate a template report in advance to see exactly what kind of a service they will get for the

money. Comparing such testing houses is a straightforward business of comparing their template reports. Be sure again to find a test house that does not simply run basic Satan-type host-by-host tests, but can genuinely produce a site-wide result. Also check that the written report is structured in a user-friendly manner, with non-jargon overviews for senior managers as well as detailed technical content.

CONCLUSION

Security is a constantly changing field, and there is no fix-and-forget solution – although many managers would prefer this! It requires on-going effort, to keep abreast of the latest security threats, to keep software versions patched, to review perimeter configurations, and take active testing audits on a regular basis.

None of this effort will guarantee 100% security. What it will do is greatly reduce the likelihood of an Internet security breach, and, with the number of opportunistic Internet attacks rising to over 100 published breaches a month, that must make sense.

Deri Jones
Security Specialist
NTA Monitor Ltd (UK)

© NTA Monitor Ltd 1998

Call for papers

Why not share your expertise and earn money at the same time? *TCP/SNA Update* is looking for REXX EXECs, macros, CLISTs, program code, etc, that experienced networkers have written to make their life, or the lives of their users, easier. We will publish it (after vetting by our expert panel) and send you a cheque when the article is published. Why not call now for a free copy of our *Notes for contributors*?

Programming in TCP/IP with REXX

With OS/390 and TCP/IP, I use REXX to program in client/server mode.

Program TCPIPCCLE1 is the client program which communicates with the server program TCPIPSE1. The server program should be started before the client one.

This REXX EXEC sends and receives messages in client/server mode. TCPIPCL2 and TCPIPSE2 are two other examples of programming TCP/IP with REXX under OS/390.

The TCP/IP port number 8000 is used, which you can change it if it's already in use at your site.

You have to change the IP address 155.3.1.41 to your own address.

TCPIPCCLE1

```
parse value socket('initialize','testtcp2') with cr id md s
say 'initialize' cr id md s
parse value socket('socket') with cr sock
say 'socket' cr sock
parse value socket('connect',sock,'2 8000 155.3.1.41') with cr reste
say 'connect' cr reste
parse value socket('send',sock,'Coucou') with cr reste
say 'send' cr reste
parse value socket('recv',sock) with cr reste
say 'recv' cr reste
parse value socket('terminate') with cr id2
say 'terminate' cr id2
```

TCPIPCL2

```
parse value socket('initialize','testtcp2') with cr id md s
say 'initialize' cr id md s
parse value socket('socket','2','DATAGRAM','UDP') with cr sock
say 'socket' cr sock
parse value socket('bind',sock,'2 8001 155.3.1.41') with cr reste
say 'bind' cr reste
parse value socket('fcntl',sock,'F_GETFL') with cr reste
say 'fcntl' cr reste
```

```

say 'Debut'
parse pull texte
do while texte≠''
  parse value socket('sendto',sock,texte,, '2 8000 155.3.1.41') ,
    with cr reste
  say 'sendto' cr reste
  parse value socket('recvfrom',sock) with cr reste
  say 'recvfrom' cr reste
  parse pull texte
end
parse value socket('terminate') with cr id2
say 'terminate' cr id2

```

TCPIPSE1

```

parse value socket('initialize','testtcp1') with cr id md s
say 'initialize' cr id md s
parse value socket('socket') with cr sock
say 'socket' cr sock
parse value socket('bind',sock,'2 8000 155.3.1.41') with cr reste
say 'bind' cr reste
parse value socket('listen',sock) with cr reste
say 'listen' cr reste
parse value socket('accept',sock) with cr sock2 reste
say 'accept' cr sock2 reste
parse value socket('recv',sock2) with cr reste
say 'recv' cr reste
parse value socket('send',sock2,'Salut') with cr reste
say 'send' cr reste
parse value socket('terminate') with cr id2
say 'terminate' cr id2

```

TCPIPSE2

```

parse value socket('initialize','testtcp1') with cr id md s
say 'initialize' cr id md s
parse value socket('socket','2','DATAGRAM','UDP') with cr sock
say 'socket' cr sock
parse value socket('bind',sock,'2 8000 155.3.1.41') with cr reste
say 'bind' cr reste
do forever
  parse value socket('recvfrom',sock) with cr type port adr long texte
  say 'recvfrom' cr type port adr long texte
  parse value socket('sendto',sock,'Echo:' texte , , ,
    type port adr) ,
    with cr reste

```

```
say 'sendto' cr reste
upper texte; if texte='FIN' then leave
end
parse value socket('terminate') with cr id2
say 'terminate' cr id2
```

Claude Dunand (France)

© Xephon 1998

FTP under MVS in batch mode

If you need to transport files from MVS to another computing environment that supports TCP/IP, you can use FTP. I use an FTP batch utility to simplify this task

In this example, the source dataset is SYS1.PARMLIB, and we are going to transmit (copy) this file to the IP address 155.3.2.128. It could have been a host-id name like MVS001.

First you must enter a valid user-id and a password.

The binary option is only required when doing an FTP transfer to a non-390 system (ie not MVS or VM).

```
FTPMVS EXEC PGM=FTP,REGION=2048K,
//          PARM='155.3.2.128 (EXIT '
//STEPLIB DD DSN=SYS1.SEZALINK,DISP=SHR
//SYSTCPD DD DSN=TCPIP.PARMLIB(TCPDATA),DISP=SHR
//SYSFCPD DD DSN=TCPIP.PARMLIB(FCPDATA),DISP=SHR
//SYSPRINT DD SYSOUT=*
//OUTPUT DD SYSOUT=*
//INPUT DD *
syslbd @@@@
TYPE E
MODE B
CD 'SYS1.PARMLIB'
LCD 'SYS1.PARMLIB'
DIR
LCD
PWD
CLOSE
QUIT
/*
```

Claude Dunand (France)

© Xephon 1998

A mailbox system for SMTP under MVS TCP/IP – 2

This issue we continue the code for the implementation of a mailbox system for SMTP, based on ISPF functions.

INST81C

The panel INST81C is used to display the mail directory:

```
)ATTR
! TYPE(OUTPUT) COLOR(GREEN) CAPS(off) JUST(LEFT) PAD(' ') SKIP(on)
\ TYPE(OUTPUT) COLOR(RED) CAPS(OFF)
¬ TYPE(INPUT) INTENS(LOW) CAPS(ON) JUST(LEFT) PAD(' ')
/ TYPE(INPUT) INTENS(NON)
" TYPE(TEXT) COLOR(GREEN) INTENS(HIGH)
` TYPE(OUTPUT) COLOR(PINK) CAPS(off)
' TYPE(TEXT) COLOR(YELLOW) INTENS(HIGH)
; TYPE(TEXT) COLOR(RED) INTENS(HIGH)
? TYPE(TEXT) COLOR(PINK) INTENS(HIGH)
TYPE(INPUT ) INTENS(HIGH) COLOR(RED) PAD(_) CAPS(ON)
% TYPE(TEXT) COLOR(WHITE) SKIP(OFF)
+ TYPE(TEXT) COLOR(BLUE) SKIP(ON)
# TYPE(OUTPUT) COLOR(RED)
$ TYPE(OUTPUT) COLOR(YELLOW) CAPS(off)
)BODY EXPAND(@@)
%--@-@--;Mail Directory%--@-@--
'Command ==>_ZCMD +SCROLL ==>_AMT %
"Command?P"to Print,?U USERID"Search for Userid,?N NAME"Search for Name,
?D DOMAIN"Search for Domain,?F STRING"Find string,?X STRING"Find all
strings.
?SORT:"Sort on USERID, DOMAIN or NAME ascending A or descending D.
;S'Userid "Domain 'Name "C
%
)MODEL ROWS(&SCAN)
Z!USERID $DOMAIN `NAME #C
)INIT
ZVARS = '( SEL )'
&AMT = PAGE
)REINIT
IF (.MSG = ' ')
&SEL = ' '
REFRESH(SEL)
&ZCMD = ' '
REFRESH(ZCMD)
)PROC
IF (.RESP = ENTER)
&PFKEY = .PFKEY
```

```
VPUT (PFKEY)
)END
```

MAILDIRS

The CLIST MAILDIRS is used to display the mail directory and perform directory actions like copy receivers, find, sort, and print:

```
/*                                                                    */
/* MAILDIRS                                                            */
/* Display and Search SMTP Mail Directory                             */
/* Called from MAILSENS, ZCC and Panel INST81                       */
/*                                                                    */
/* Parameters:                                                         */
/* CALLER : identification of calling function.                      */
/*                                                                    */
/* Subroutines/Edit Macros:                                           */
/* %ZCC                                                                */
/* Isp table used: MAILDIR                                           */
/* Skeleton used : MAILDIR                                           */
/* Panel used      : INST81C                                          */
/*                                                                    */
/* Utilities used:                                                     */
/* SLEEP                                                       */
/* TSOLINE1                                                    */
/*                                                                    */
PROC Ø CALLER(DUMMY) DEBUG(nEBUG)
CONTROL NOMSG NOFLUSH NOLIST NOCONLIST NOSYMLIST NOCAPS
ATTN DO
  SET &FLUSH = FLUSH          /* NEXT STATEMENT MUST BE NULL LINE */
END
ERROR DO
  SET &RET = &LASTCC
  RETURN
END
SET &RET = Ø
IF &SYSCAPS(&STR(&DEBUG)) = DEBUG THEN DO
  CONTROL MSG NOFLUSH LIST CONLIST SYMLIST
  WRITE =====> Entering &SYSICMD <=====
END
IF &FLUSH = FLUSH THEN DO
  ISPEXEC TBEND &TABLEID
  IF &STR(&ZPFØ5) = &STR(F) THEN DO
    SET &ZPFØ5 = &STR(&SZPFØ5)
    SET &ZPF17 = &STR(&SZPF17)
    ISPEXEC VPUT (ZPFØ5,ZPF17) PROFILE
  END
  SET &DEBUG = NEBUG
  ISPEXEC VPUT (DEBUG) SHARED
  SET &ZEDSMMSG = &str(Function interrupted)
```

```

ISPEXEC SETMSG MSG(ISRZ001)
EXIT CODE(4)
END
IF &SYSISPF = &STR(NOT ACTIVE) THEN DO
  WRITE =====> Sorry only executable under ISPF (&SYSICMD).
  EXIT CODE(16)
END
ISPEXEC CONTROL ERRORS RETURN
ISPEXEC VGET (ZSCREEN)
SET COPIED = NO
SET &TABLEID = &STR(MAILDIR)
SET &TLIB = INST.ISPTLIB
SET &PANEL = INST81C
ISPEXEC VGET (ZPF05,ZPF17) PROFILE
SET &SZPF05 = &STR(&ZPF05)
SET &SZPF17 = &STR(&ZPF17)
SET &ZPF05 = &STR(F)          /* FIND          */
SET &ZPF17 = &STR(F)          /* FIND          */
ISPEXEC VPUT (ZPF05,ZPF17) PROFILE
ISPEXEC VGET (ZSCREEN)
ISPEXEC TBCLOSE &TABLEID
IF (&SYSDSN(&STR('&TLIB(&TABLEID)')) NE OK THEN DO
  ISPEXEC CONTROL DISPLAY LINE START(14)
  SLEEP 1
  WRITE =====> +
  &STR(Table &TLIB(&TABLEID) &SYSDSN(&STR('&TLIB(&TABLEID)'))))
  IF &STR(&ZPF05) = &STR(F) THEN DO
    SET &ZPF05 = &STR(&SZPF05)
    SET &ZPF17 = &STR(&SZPF17)
    ISPEXEC VPUT (ZPF05,ZPF17) PROFILE
  END
  EXIT CODE(16)
END
ISPEXEC TBOPEN &TABLEID NOWRITE SHARE
SET &SORTCRIT = &STR(NAME,C,A,USERID,C,A,DOMAIN,C,A)
ISPEXEC TBSORT &TABLEID FIELDS(&SORTCRIT)
ISPEXEC TBTOP &TABLEID
SET &ZTDTOP = &STR(0)
SET &LASTHIT = &STR(0)
SET &FINDNUM = 0
ISPEXEC CONTROL DISPLAY REFRESH
SET &TBDISRET = 0
DO WHILE &ZTDSELS > 0 OR &TBDISRET < 8
  IF &SYSENV NE BACK THEN DO
    SET &SEL = &STR()
    IF &SAVEI = YES THEN DO
      ISPEXEC CONTROL DISPLAY RESTORE
      SET &SAVEI = &STR()
    END
  IF &ZTDSELS > 0 AND &TBDISRET NE 20 AND &TBDISRET NE 8 THEN DO
    ISPEXEC TBTOP &TABLEID

```

```

ISPEXEC TBSKIP &TABLEID NUMBER(&LINENUM) NOREAD
SET &RET = 0
IF &ZTDSLS > 1 THEN DO
  ISPEXEC TDISPL &TABLEID POSITION(LINENUM)
END
ELSE DO
  ISPEXEC TDISPL &TABLEID POSITION(LINENUM) PANEL(&PANEL)
END
SET &TBDISRET = &RET
ISPEXEC CONTROL DISPLAY SAVE
SET &SAVEI = YES
END
ELSE DO
  ISPEXEC TBTOP &TABLEID
  ISPEXEC VGET (ZVERB ZSCROLLN)
  SET &SCROLL = &ZDTOP + &FINDNUM
  IF &ZVERB = UP THEN DO
    SET &SCROLL = &STR(-&ZSCROLLN+&ZDTOP)
  END
  IF &ZVERB = DOWN THEN DO
    SET &SCROLL = &STR(&ZSCROLLN+&ZDTOP)
  END
  IF &SCAN NE &STR(SCAN) THEN DO
    ISPEXEC TBSKIP &TABLEID NUMBER(&SCROLL) NOREAD
    ISPEXEC TBQUERY &TABLEID ROWNUM(ROWNUM) POSITION(CRP)
  END
  SET &RET = 0
  IF &ZDTOP > &ROWNUM AND &SCAN NE &STR(SCAN) THEN DO
    ISPEXEC TDISPL &TABLEID POSITION(LINENUM)
  END
  ELSE DO
    IF &SCROLL > 1 THEN DO
      ISPEXEC TDISPL &TABLEID POSITION(LINENUM) PANEL(&PANEL)
    END
    ELSE DO
      ISPEXEC TDISPL &TABLEID POSITION(LINENUM) PANEL(&PANEL)
    END
  END
  SET &TBDISRET = &RET
END
END
/* CONTINUE FROM PANEL */
IF &LENGTH(&STR(&SYSNSUB(1,&ZCMD))) > 0 THEN DO
  SET &STRING = &STR(&SYSNSUB(1,&ZCMD))
  SET &RSTRING = &STR() /* CLEAR FOR CALL TO SUBROUTINE RETURN */
  SET &RET = 0
  SYSCALL VARSTRNG RSTRING STRING(&STR('&SYSNSUB(1,&STRING)')) +
  DEBUG(&DEBUG)
  /* SYSCALL RETURN CODE DOES NOT TAKE ERROR ROUTINE */
  SET &RET = &LASTCC
  SET &ZCMD = &STR(&SYSNSUB(1,&RSTRING))

```

```

END
IF &SYSCAPS(&STR(&SYSNSUB(1,&ZCMD))) = NODEBUG THEN DO
  CONTROL NOMSG NOFLUSH NOLIST NOCONLIST NOSYMLIST
  SET &DEBUG = &SYSCAPS(&STR(&SYSNSUB(1,&ZCMD)))
  ISPEXEC VPUT (DEBUG) SHARED
END
IF &SYSCAPS(&STR(&SYSNSUB(1,&ZCMD))) = TIME THEN DO
  SET &DEBUG = &SYSCAPS(&STR(&SYSNSUB(1,&ZCMD)))
  ISPEXEC VPUT (DEBUG) SHARED
END
IF &SYSCAPS(&STR(&SYSNSUB(1,&ZCMD))) = NOTIME THEN DO
  SET &DEBUG = &SYSCAPS(&STR(&SYSNSUB(1,&ZCMD)))
  ISPEXEC VPUT (DEBUG) SHARED
END
/* PROCESS COMMAND REPEAT                                     */
IF &STR(&SYSNSUB(1,&P1)) NE X THEN DO
  SET &CONTINUE = &STR(YES)
END
DO WHILE (&STR(&SYSNSUB(1,&P1)) = X AND &PFKEY = PF05) OR +
&CONTINUE NE &STR()
  ISPEXEC VGET (PFKEY)
  IF &PFKEY = PF05 OR &PFKEY = PF17 OR +
(&STR(&SYSNSUB(1,&ZCMD)) = +
&STR(&SYSNSUB(1,&SAVEFIND)) AND &STR(&SYSNSUB(1,&ZCMD)) NE &STR()) +
  THEN DO
    IF &ZTDTOP < &LASTHIT AND &ZTDTOP <= &ROWNUM AND &LASTHIT > 0 +
    THEN DO
      SET &ZTDTOP = &ZTDTOP - 1
    END
    IF &STR(&SYSNSUB(1,&SAVEFIND)) NE &STR() THEN DO
      SET &ZCMD = &STR(&SYSNSUB(1,&SAVEFIND))
    END
    SET &SCROLL = &ZTDTOP
    IF &STR(&SYSNSUB(1,&P1)) = X THEN DO
      SET &SCROLL = &STR(+0)
    END
    ISPEXEC TBSKIP &TABLEID NUMBER(&SCROLL) NOREAD
  END
  SET &SYSDVAL = &STR(&SYSNSUB(1,&ZCMD))
  SET &SYSDVAL = &STR(&SYSNSUB(1,&SYSDVAL))
  READDVAL &P1 &P2 &P3 &P4 &P5 &P6 &P7
  IF &SYSCAPS(&STR(&SYSNSUB(1,&ZCMD))) = DEBUG THEN DO
    CONTROL MSG NOFLUSH LIST CONLIST SYMLIST
    SET &DEBUG = &SYSCAPS(&STR(&SYSNSUB(1,&ZCMD)))
    ISPEXEC VPUT (DEBUG) SHARED
  END
  /* PREPARE FIND FUNCTIONS                                     */
  IF (&STR(&SYSNSUB(1,&P1)) = U OR &STR(&SYSNSUB(1,&P1)) = N OR +
&STR(&SYSNSUB(1,&P1)) = D OR &STR(&SYSNSUB(1,&P1)) = F OR +
&STR(&SYSNSUB(1,&P1)) = X) AND +
&STR(&SYSNSUB(1,&P2)) NE &STR() THEN DO

```



```

SET &FIND = YES
SET &SAVEFIND = &STR(&SYSNSUB(1,&ZCMD))
IF &STR(&SYSNSUB(1,&P1)) = X AND &PFKEY = &STR() THEN DO
  ISPEXEC TBTOP &TABLEID
END
ELSE DO
  SET &FIND = NO
END
/* PROCESS INPUT COMMAND */
IF &LENGTH(&STR(&SYSNSUB(1,&ZCMD))) > 5 THEN DO
  IF &STR(&SYSNSUB(1,&P1)) = SORT THEN DO
    IF &STR(&SYSNSUB(1,&P3)) = &STR() THEN DO
      SET &P3 = &STR(A)
    END
    IF &STR(&SYSNSUB(1,&P3)) = &STR(ASCENDING) THEN DO
      SET &P3 = &STR(A)
    END
    IF &STR(&SYSNSUB(1,&P3)) = &STR(D) THEN DO
      SET &P3 = &STR(D)
    END
    IF &STR(&SYSNSUB(1,&P3)) NE &STR(ASCENDING) AND +
    &STR(&SYSNSUB(1,&P3)) NE &STR(D) AND +
    &STR(&SYSNSUB(1,&P3)) NE &STR(A) AND +
    &STR(&SYSNSUB(1,&P3)) NE &STR(D) THEN DO
      SET &P7 = &SYSNSUB(1,&P6)
      SET &P6 = &SYSNSUB(1,&P5)
      SET &P5 = &SYSNSUB(1,&P4)
      SET &P4 = &SYSNSUB(1,&P3)
      SET &P3 = &STR(A)
    END
  SET &SORTCRIT = &STR(&SYSNSUB(1,&P2,C,&P3))
  IF &STR(&SYSNSUB(1,&P4)) NE &STR() THEN DO
    IF &STR(&SYSNSUB(1,&P5)) = &STR() THEN DO
      SET &P5 = &STR(A)
    END
    IF &STR(&SYSNSUB(1,&P5)) = &STR(ASCENDING) THEN DO
      SET &P5 = &STR(A)
    END
    IF &STR(&SYSNSUB(1,&P5)) = &STR(D) THEN DO
      SET &P5 = &STR(D)
    END
    IF &STR(&SYSNSUB(1,&P5)) NE &STR(ASCENDING) AND +
    &STR(&SYSNSUB(1,&P5)) NE &STR(D) AND +
    &STR(&SYSNSUB(1,&P5)) NE &STR(A) AND +
    &STR(&SYSNSUB(1,&P5)) NE &STR(D) THEN DO
      SET &P7 = &SYSNSUB(1,&P6)
      SET &P6 = &SYSNSUB(1,&P5)
      SET &P5 = &STR(A)
    END
  SET &SORTCRIT = &STR(&SYSNSUB(1,&P2,C,&P3,&P4,C,&P5))

```

```

END
IF &STR(&SYSNSUB(1,&P6)) NE &STR() THEN DO
  IF &STR(&SYSNSUB(1,&P7)) = &STR() THEN DO
    SET &P7 = &STR(A)
  END
  IF &STR(&SYSNSUB(1,&P7)) = &STR(ASCENDING) THEN DO
    SET &P7 = &STR(A)
  END
  IF &STR(&SYSNSUB(1,&P7)) = &STR(DESCENDING) THEN DO
    SET &P7 = &STR(D)
  END
  IF &STR(&SYSNSUB(1,&P7)) NE &STR(ASCENDING) AND +
&STR(&SYSNSUB(1,&P7)) NE &STR(DESCENDING) AND +
&STR(&SYSNSUB(1,&P7)) NE &STR(A) AND +
&STR(&SYSNSUB(1,&P7)) NE &STR(D) THEN DO
    SET &P7 = &STR(A)
  END
  SET &SORTCRIT = &STR(&SYSNSUB(1,&P2,C,&P3,&P4,C,&P5,&P6,C,&P7))
END
IF &STR(&SYSNSUB(1,&P2)) NE &STR(NAME) AND +
&STR(&SYSNSUB(1,&P4)) NE &STR(NAME) AND +
&STR(&SYSNSUB(1,&P6)) NE &STR(NAME) THEN DO
  SET &SORTCRIT = &SYSNSUB(1,&SORTCRIT)&STR(,NAME,C,A)
END
IF &STR(&SYSNSUB(1,&P2)) NE &STR(USERID) AND +
&STR(&SYSNSUB(1,&P4)) NE &STR(USERID) AND +
&STR(&SYSNSUB(1,&P6)) NE &STR(USERID) THEN DO
  SET &SORTCRIT = &SYSNSUB(1,&SORTCRIT)&STR(,USERID,C,A)
END
IF &STR(&SYSNSUB(1,&P2)) NE &STR(DOMAIN) AND +
&STR(&SYSNSUB(1,&P4)) NE &STR(DOMAIN) AND +
&STR(&SYSNSUB(1,&P6)) NE &STR(DOMAIN) THEN DO
  SET &SORTCRIT = &SYSNSUB(1,&SORTCRIT)&STR(,DOMAIN,C,A)
END
ISPEXEC TBSORT &TABLEID FIELDS(&SYSNSUB(1,&SORTCRIT))
ISPEXEC TBTOP &TABLEID
END
END
IF &LENGTH(&STR(&SYSNSUB(1,&ZCMD))) > 0 THEN DO
  SET &SAVEZCMD = &STR(&SYSNSUB(1,&ZCMD))
END
IF &SYSCAPS(&STR(&SYSNSUB(1,&ZCMD))) = P THEN DO
  SET &SKEL = &TABLEID
  ISPEXEC TBTOP &TABLEID
  ISPEXEC FTOPEM TEMP
  ISPEXEC FTINCL &SKEL
  ISPEXEC FTCLOSE
  ISPEXEC VGET ZTEMPF
  SUB * END(ZZ) JOBCHAR(Y)
//STEP1 EXEC PGM=IEBGENER,REGION=6M
//SYSPRINT DD SYSOUT=&DUMMYCL

```

```

//SYSIN DD DUMMY
//SYSUT2 DD SYSOUT=*
//SYSUT1 DD DSN=&ZTEMPF,DISP=SHR
ZZ
FREE DA('&ZTEMPF')
SET &ZEDSMMSG = &str(Print job submitted)
ISPEXEC SETMSG MSG(ISRZ001)
END
/* SET UP FOR FIND FUNCTIONS */
IF &FIND = YES THEN DO
SET &FINDNUM = 0
SET &TBSKRET = 0
SET &SYSDVAL = &STR(&SYSNSUB(1,&SAVEFIND))
SET &SYSDVAL = &STR(&SYSNSUB(1,&SYSDVAL))
READDVAL &A1 &A2 &A3 &A4 &A5 &A6 &A7 &A8
IF &STR(&SYSNSUB(1,&P1)) = U THEN DO
SET &ARG = USERID
END
IF &STR(&SYSNSUB(1,&P1)) = N THEN DO
SET &ARG = NAME
END
IF &STR(&SYSNSUB(1,&P1)) = D THEN DO
SET &ARG = DOMAIN
END
SET &SEEK = &STR(&SYSNSUB(1,&A2))
IF &STR(&SYSNSUB(1,&P1)) = U OR &STR(&SYSNSUB(1,&P1)) = N OR +
&STR(&SYSNSUB(1,&P1)) = D OR &STR(&SYSNSUB(1,&P1)) = F OR +
&STR(&SYSNSUB(1,&P1)) = X THEN DO
IF &STR(&SYSNSUB(1,&A3)) NE &STR() THEN DO
SET &A2 = &STR(&SYSNSUB(1,&A2))&STR( )&STR(&SYSNSUB(1,&A3))
END
IF &STR(&SYSNSUB(1,&A4)) NE &STR() THEN DO
SET &A2 = &STR(&SYSNSUB(1,&A2))&STR( )&STR(&SYSNSUB(1,&A4))
END
IF &STR(&SYSNSUB(1,&A5)) NE &STR() THEN DO
SET &A2 = &STR(&SYSNSUB(1,&A2))&STR( )&STR(&SYSNSUB(1,&A5))
END
IF &STR(&SYSNSUB(1,&A6)) NE &STR() THEN DO
SET &A2 = &STR(&SYSNSUB(1,&A2))&STR( )&STR(&SYSNSUB(1,&A6))
END
IF &STR(&SYSNSUB(1,&A7)) NE &STR() THEN DO
SET &A2 = &STR(&SYSNSUB(1,&A2))&STR( )&STR(&SYSNSUB(1,&A7))
END
IF &STR(&SYSNSUB(1,&A8)) NE &STR() THEN DO
SET &A2 = &STR(&SYSNSUB(1,&A2))&STR( )&STR(&SYSNSUB(1,&A8))
END
SET &SEEK = &SYSLC(&STR(&SYSNSUB(1,&A2)))
END
/* FIND FUNCTIONS */
DO WHILE &TBSKRET = 0 AND &FIND = YES
SET &SCAN = &STR()

```

```

SET &RET = 0
ISPEXEC TBSKIP &TABLEID POSITION(FINDNUM)
SET &TBSKRET = &RET
SET &FIELD = &SYSLC(&STR(&SYSNSUB(4,&&&ARG))) /* field search */
IF &STR(&SYSNSUB(1,&P1)) = F THEN DO
  SET &FIELD = &SYSLC(&STR(&SYSNSUB(1,&USERID))+
    &STR(&SYSNSUB(1,&NAME))&STR(&SYSNSUB(1,&DOMAIN)))
END
IF &STR(&SYSNSUB(1,&P1)) = X THEN DO
  SET &FIELD = &SYSLC(&STR(&SYSNSUB(1,&USERID))+
    &STR(&SYSNSUB(1,&NAME))&STR(&SYSNSUB(1,&DOMAIN)))
END
IF &TBSKRET = 0 THEN DO
  IF &LENGTH(&SYSNSUB(1,&SEEK)) <= +
    &LENGTH(&SYSNSUB(1,&FIELD)) +
  THEN DO
    SET &NAMESTR = +
      &SYSINDEX(&SYSNSUB(1,&SEEK),&SYSNSUB(1,&FIELD))
    SET &NAMESTR = +
      &SYSINDEX(&SYSNSUB(2,&SEEK),&SYSNSUB(2,&FIELD))
    IF &NAMESTR > 0 THEN DO
      SET &TBSKRET = 9999
      IF &LASTHIT > 0 THEN DO
        SET &ZTDTOP = &STR(0)
      END
    ELSE DO
      SET &ZTDTOP = &ZTDTOP - 1
    END
    SET &ZTDTOP = &STR(0)
    IF &STR(&SYSNSUB(1,&P1)) = X THEN DO
      ISPEXEC CONTROL DISPLAY LINE START(8)
      WRITE &STR(&SYSNSUB(1,&USERID))&STR( )+
        &STR(&SYSNSUB(1,&DOMAIN))&STR( )&STR(&SYSNSUB(1,&NAME))
      SET &XHIT = YES
      SET &PFKEY = &STR(PF05)
      ISPEXEC VPUT (PFKEY)
    END
    SET &LASTHIT = &FINDNUM
    SET &ARG = &STR(&SYSNSUB(1,&A2))
    ISPEXEC VPUT (ARG)
    SET &ZEDSMMSG = &STR(&SYSNSUB(1,&ARG) found)
    ISPEXEC SETMSG MSG(ISRZ001)
    SET &ARG = &STR( )
    ISPEXEC VPUT (ARG)
  END
END
IF &TBSKRET = 8 THEN DO
  SET &ARG = &STR(&SYSNSUB(1,&A2))
  ISPEXEC VPUT (ARG)
  SET &ZEDSMMSG = &STR(&SYSNSUB(1,&ARG) not found)

```

```

ISPEXEC SETMSG MSG(ISRZ001)
IF &STR(&SYSNSUB(1,&P1)) = X THEN DO
  SET &PFKEY = &STR()
  ISPEXEC VPUT (PFKEY)
  IF &XHIT = YES THEN DO
    SET &ZEDSMMSG = &STR(&SYSNSUB(1,&ARG) found)
    ISPEXEC SETMSG MSG(ISRZ001)
  END
  SET &XHIT = NO
END
SET &ARG = &STR()
ISPEXEC VPUT (ARG)
END
END
SET &ZCMD = &STR()
END
ELSE DO
  SET &FINDNUM = 0
END
IF &STR(&SYSNSUB(1,&P1)) NE X THEN DO
  SET &CONTINUE = &STR()
END
END
/* SELECT FUNCTIONS */
IF &LENGTH(&SYSNSUB(1,&SEL)) > 0 THEN DO
  SET &STRING = &STR(&SYSNSUB(1,&SEL))
  SET &RSTRING = &STR() /* CLEAR FOR CALL TO SUBROUTINE RETURN */
  SET &RET = 0
  SYSCALL VARSTRNG RSTRING STRING(&STR('&SYSNSUB(1,&STRING)')) +
  DEBUG(&DEBUG)
  /* SYSCALL RETURN CODE DOES NOT TAKE ERROR ROUTINE */
  SET &RET = &LASTCC
  SET &SEL = &STR(&SYSNSUB(1,&RSTRING))
END
IF &STR(&SEL) NE &STR() THEN DO
  IF &STR(&CALLER) NE DUMMY THEN DO
    ISPEXEC CONTROL DISPLAY LINE START(14)
    SLEEP 1
  END
  SET &STRING = &STR(&SYSNSUB(1,&USERID))
  SET &RSTRING = &STR() /* CLEAR FOR CALL TO SUBROUTINE RETURN */
  SET &RET = 0
  SYSCALL VARSTRNG RSTRING STRING(&STR('&SYSNSUB(1,&STRING)')) +
  DEBUG(&DEBUG)
  /* SYSCALL RETURN CODE DOES NOT TAKE ERROR ROUTINE */
  SET &RET = &LASTCC
  SET &DIRRC = &STR(&SYSNSUB(1,&RSTRING))
  SET &MEMOCC = &STR(&SYSNSUB(1,&RSTRING))
  /* */
  SET &STRING = &STR(&SYSNSUB(1,&DOMAIN))
  SET &RSTRING = &STR() /* CLEAR FOR CALL TO SUBROUTINE RETURN */

```

```

SET &RET = Ø
SYSCALL VARSTRNG RSTRING STRING(&STR('&SYSNSUB(1,&STRING)')) +
DEBUG(&DEBUG)
/* SYSCALL RETURN CODE DOES NOT TAKE ERROR ROUTINE */
SET &RET = &LASTCC
SET &DIRDM = &STR(&SYSNSUB(1,&RSTRING))
SET &CCDM = &STR(&SYSNSUB(1,&RSTRING))
/*
SET &STRING = &STR(&SYSNSUB(1,&NAME))
SET &RSTRING = &STR() /* CLEAR FOR CALL TO SUBROUTINE RETURN */
SET &RET = Ø
SYSCALL VARSTRNG RSTRING STRING(&STR('&SYSNSUB(1,&STRING)')) +
DEBUG(&DEBUG)
/* SYSCALL RETURN CODE DOES NOT TAKE ERROR ROUTINE */
SET &RET = &LASTCC
SET &SNAME = &STR(&SYSNSUB(1,&RSTRING))
/*
IF &STR(&CALLER) NE ZCC THEN DO
  IF &STR(&CALLER) NE DUMMY THEN DO
    WRITE &STR(&SYSNSUB(1,&DIRRC) AT &SYSNSUB(1,&DIRDM) +
    &SYSNSUB(1,&SNAME)) copied.
    IF &ZTDSELS > 1 THEN DO
      WRITE =====> Only one receiver can be selected as primary; use ZCC +
      to add CC-receivers
    END
    SET &COPIED = YES
    ISPEXEC VPUT (DIRRC,DIRDM)
    SET &C = &STR(X)
    ISPEXEC TBEND &TABLEID
    IF &STR(&ZPFØ5) = &STR(F) THEN DO
      SET &ZPFØ5 = &STR(&SZPFØ5)
      SET &ZPF17 = &STR(&SZPF17)
      ISPEXEC VPUT (ZPFØ5,ZPF17) PROFILE
    END
    EXIT CODE(Ø) /* RECEIVER COPIED FROM DIRECTORY */
  END
END
ELSE DO
  IF &STR(&SYSNSUB(1,&MEMOCC)) NE &STR(?) THEN DO
    IF &STR(&C) NE &STR() THEN DO
      WRITE =====> &STR(&SYSNSUB(1,&MEMOCC) AT &SYSNSUB(1,&CCDM)) +
      ignored, was already selected as CC.
      SLEEP 1
    END
  ELSE DO
    WRITE &STR(&SYSNSUB(1,&MEMOCC) AT &SYSNSUB(1,&CCDM) +
    &SYSNSUB(1,&SNAME)) copied.
    SET &COPIED = YES
    ISPEXEC VPUT (MEMOCC,CCDM)
    SET &C = &STR(X)
    SET &RET = Ø
  END
END

```

```

    ISPEXEC TBPUR &TABLEID
    %ZCC CALLER(&SYSICMD) DEBUG(&STR(&DEBUG)) /* recursive call */
    IF &SYSCAPS(&STR(&DEBUG)) = DEBUG THEN DO
        WRITE =====> Reentering &SYSICMD <=====
    END
    SET &MEMOCC = &STR()
    SET &CCDM = &STR()
    ISPEXEC VPUT (MEMOCC,CCDM)
END
END
IF &ZTDSLS = 1 THEN DO
    TSOLINE1 /* clear screen when last select handled */
END
END
END
END
ISPEXEC TBEND &TABLEID
IF &STR(&ZPF05) = &STR(F) THEN DO
    SET &ZPF05 = &STR(&SZPF05)
    SET &ZPF17 = &STR(&SZPF17)
    ISPEXEC VPUT (ZPF05,ZPF17) PROFILE
END
/* */
IF &COPIED = YES THEN DO
    IF &STR(&CALLER) NE ZCC THEN DO
        ISPEXEC VPUT (DIRRC,DIRDM)
    END
ELSE DO
    SET &CCDM = &STR()
    IF &COPIED = YES THEN DO
        SET &MEMOCC = &STR()
    END
ELSE DO
    SET &MEMOCC = &STR(?)
END
END
ISPEXEC VPUT (MEMOCC,CCDM)
EXIT CODE(0) /* RECEIVER COPIED FROM DIRECTORY */
END
EXIT CODE(4) /* RECEIVER NOT COPIED FROM DIRECTORY */
/* INLINE SUBROUTINES */
VARSTRNG: +
PROC 1 RSTRING STRING() DEBUG(NEBUG)
/* INLINE VARSTRNG ROUTINE; TRUNCATE TRAILING BLANKS */
CONTROL NOMSG NOFLUSH NOLIST NOCONLIST NOSYMLIST
ATTN DO
    SET &FLUSH = FLUSH /* NEXT STATEMENT MUST BE NULL LINE */
END
ERROR DO
    SET &RET = &LASTCC
RETURN

```

```

END
IF &SYSCAPS(&STR(&DEBUG)) NE DEBUG THEN DO
  ISPEXEC VGET (DEBUG)
END
IF &STR(&DEBUG) = &STR() THEN DO
  SET &DEBUG = NEBUG
END
IF &SYSCAPS(&STR(&DEBUG)) = DEBUG THEN DO
  CONTROL MSG NOFLUSH LIST CONLIST SYMLIST
END
IF &FLUSH = FLUSH THEN DO
  SET &DEBUG = NEBUG
  ISPEXEC VPUT (DEBUG) SHARED
  SET &ZEDSMMSG = &str(Function interrupted)
  ISPEXEC SETMSG MSG(ISRZ001)
  RETURN CODE(0)
END
ISPEXEC CONTROL ERRORS RETURN
SYSREF &RSTRING
SET &RSTRING = &STR(&SYSNSUB(1,&STRING))
IF &STR(&SYSNSUB(1,&STRING)) = &STR() THEN DO
  RETURN CODE(0)
END
SET &RET = 0
SET &ALLBLANKS = +
&STR(
-
-
-
)

IF &LENGTH(&SYSNSUB(1,&STRING)) > 0 THEN DO
  SET &SRCHSPC = &STR( ) /* look for last blank */
  SET &STARTSPC = 1
  SET &LOCSPC = 1
  SET &LASTSPC = 0
  SET &LENRCP = &LENGTH(&STR(&SYSNSUB(1,&STRING)))
  SET &MAXT = &LENRCP
  SET &R = 0
  DO WHILE &LOCSPC > 0 AND &STARTSPC <= &LENRCP AND &R < &MAXT
    SET &R = &R + 1
    SET &LOCSPC = +
    &SYSINDEX(&STR(&SRCHSPC),&STR(&SYSNSUB(1,&STRING)),&STARTSPC)
    IF &LOCSPC > 0 THEN DO
      SET &STARTSPC = &LOCSPC + 1
      SET &LASTSPC = &LOCSPC
      IF &LASTSPC > 0 AND &LENRCP > &EVAL(&LASTSPC + 1) +
      THEN DO /* REST BLANKS */
        IF &SUBSTR(&LASTSPC:&LENRCP,&STR(&SYSNSUB(1,&STRING))) = +
        &SUBSTR(1:&LENRCP-&LASTSPC-1,&STR(&ALLBLANKS)) THEN DO
          SET &STRING = +
          &SUBSTR(1:&LASTSPC-1,&STR(&SYSNSUB(1,&STRING)))
          SET &R = &MAXT

```



```

        END
    END
END
END
END
SET &RSTRING = &STR(&SYSNSUB(1,&STRING))
RETURN CODE(Ø)
END
/*

```

```
*/
```

MAIN MAILBOX PANEL INST81

This panel is called from an ISPF panel as selection 81 as follows:

```

        '81', 'PANEL(INST81) NEWAPPL(U81) PASSLIB'

)ATTR
! TYPE(OUTPUT) INTENS(HIGH) CAPS(ON) JUST(LEFT) PAD(' ')
\ TYPE(TEXT) INTENS(LOW) CAPS(ON) JUST(LEFT) PAD(' ') SKIP(ON)
¬ TYPE(INPUT) COLOR(RED) CAPS(OFF) JUST(LEFT) PAD('_')
TYPE(INPUT) INTENS(LOW) CAPS(ON) JUST(RIGHT) PAD(' ')
< TYPE(INPUT) INTENS(NON)
" TYPE(TEXT) COLOR(GREEN) INTENS(HIGH)
¬ TYPE(TEXT) COLOR(TURQ) INTENS(HIGH)
' TYPE(TEXT) COLOR(YELLOW) INTENS(HIGH)
; TYPE(TEXT) COLOR(RED) INTENS(HIGH)
? TYPE(TEXT) COLOR(PINK) INTENS(HIGH)
3 TYPE(TEXT) COLOR(RED) INTENS(HIGH) HILITE(REVERSE) /* hex fa */
w TYPE(TEXT) COLOR(RED) INTENS(HIGH) HILITE(USCORE) /* hex fb */
p TYPE(TEXT) COLOR(RED) INTENS(HIGH) HILITE(BLINK) /* hex fc */
)BODY EXPAND(@@) window(8Ø,24)
%-@-@-;E-Mail Interface%-@-@-
%Execute Function ==>_ZCMD 'Member =>_MEMOMB
'Optnl
'Tso Dataset Name ==>_MEMODS '
'Mail Title ==>¬MEMOID
'Receivers Name ==>¬MEMORC
'Question Mark =>

Directory
'Domain ==>¬MEMODG
'Optional
+ "Mail can be sent directly from TSO to any receiver in CC:Mail,
Internet,
+ "or to a TSO-user. Mail can also be received.
+ "Mail can be sent directly from ISPF edit using macros ZMAIL,
ZREMAIL, ZREPLY
'Via SMTP:?CC:Mail Europe:"enter receivers name. Domain defaults
to;sc.ccm.com
' ?CC:Mail USA:"enter receivers name. Enter Domain
as;usa"or;us.ccm.com
' ?TSO user in MULP:"enter receivers TSO userid, Domain defaults

```

```

to;MULP
'      ?TSO user in other MVS:"enter receivers TSO userid and
Domain=NJE-addr
'      ?Internet:"enter users identification and domain name.
,
+
+ % 2+` Edit Mail in Tso Dataset      %|% 10+` Edit and Send Mail to
SMTP
+ %11+` Send Mail from Dataset to Smtp|% 12+` Receive Mail from Smtp
+ %13+` Show Mail Directory          %|%
+
+_DEBUG+
+
+
)INIT
VGET (MEMODS,MEMOMB,MEMOID,MEMORC,MEMODG)
.CURSOR = ZCMD
&ZPRIM = NO
&DEBUG = ' '
&MEMORC = ' '
&MEMODG = ' '
)REINIT
IF (&MEMODS = ' ')
    &MEMODS = '&ZUSER..SMTP'
REFRESH(*)
.ATTR (.CURSOR) = 'COLOR(RED) HILITE(REVERSE)'
)PROC
REFRESH(*)
IF (&ZCMD = ' ')
    IF (&MEMORC = '?')
        &ZCMD = '13'
VER (&ZCMD, NONBLANK, MSG=INST341)
&JOB = &ZCMD
IF (&ZCMD = 2)
    VER (&MEMODS, NONBLANK, MSG=INST344)
    VER (&MEMODS, DSNAME, MSG=INST344)
    IF (&MEMOMB = ' ')
        VER (&MEMOMB, NAME, MSG=INST345)
IF (&ZCMD = 10)
    VER (&MEMOID, NONBLANK, MSG=INST346)
    IF (&MEMOID = ' ')
        .CURSOR = MEMOID
    VER (&MEMORC, NONBLANK, MSG=INST347)
    IF (&MEMORC = ' ')
        .CURSOR = MEMORC
    &DM = ' '
    IF (&MEMODG = ' ')
        &DM = 'DOMAIN(''&MEMODG'')'
IF (&ZCMD = 11)
    VER (&MEMODS, NONBLANK, MSG=INST344)
    VER (&MEMODS, DSNAME, MSG=INST344)

```

```

IF (&MEMODS = ' ')
  .CURSOR = MEMODS
&SMTPDS = &MEMODS
IF (&MEMOMB ≠ ' ')
  VER (&MEMOMB,NAME,MSG=INST345)
  &SMTPDS = '&MEMODS(&MEMOMB) '
VER (&MEMOID,NONBLANK,MSG=INST346)
IF (&MEMOID = ' ')
  .CURSOR = MEMOID
VER (&MEMORC,NONBLANK,MSG=INST347)
IF (&MEMORC = ' ')
  .CURSOR = MEMORC
&DM = ' '
IF (&MEMODG ≠ ' ')
  &DM = 'DOMAIN('&MEMODG'' )'
&ZSEL = TRANS( TRUNC (&ZCMD, '.')
  2, 'CMD(%MAILEDIT) '
  10, 'CMD(%MAILSENS RC('&MEMORC'' ) ID('&MEMOID'' ) +
  MAILPNL(YES) &DM) '
  11, 'CMD(%MAILSENS RC('&MEMORC'' ) ID('&MEMOID'' ) +
  DS(&SMTPDS) MAILPNL(YES) &DM) '
  12, 'CMD(%MAILRECV) '
  13, 'CMD(%MAILDIRS CALLER(DUMMY)) '
  A, 'PGM(ISFISP) PARM(H &ZUSER.Y) NOCHECK NEWAPPL(ISF) '
  H, 'PGM(ISFISP) PARM(H) NOCHECK NEWAPPL(ISF) '
  I, 'PGM(ISFISP) PARM(I) NOCHECK NEWAPPL(ISF) '
  O, 'PGM(ISFISP) PARM(O) NOCHECK NEWAPPL(ISF) '
  'DA', 'PGM(ISFISP) PARM(DA OJOB) NOCHECK NEWAPPL(ISF) '
  'LO', 'PGM(ISFISP) PARM(LOG) NOCHECK NEWAPPL(ISF) '
  ' ', 'PANEL(ISR@PRIM) '
  *, '?')
VPUT (MEMODS, MEMOMB, MEMOID, MEMORC, MEMODG)
VPUT (MEMODS, MEMOID) PROFILE
)END

```

PANEL INST81A

It is called from CLIST (edit macro) ZMAIL to get mail characteristics, ie receiver, receiver's domain, and subject:

```

)ATTR
! TYPE(OUTPUT) INTENS(HIGH) CAPS(ON) JUST(LEFT) PAD(' ')
\ TYPE(TEXT) INTENS(LOW) CAPS(ON) JUST(LEFT) PAD(' ') SKIP(ON)
↵ TYPE(INPUT) COLOR(RED) CAPS(OFF) JUST(LEFT) PAD('_ ')
TYPE(INPUT) INTENS(LOW) CAPS(ON) JUST(RIGHT) PAD(' ')
< TYPE(INPUT) INTENS(NON)
" TYPE(TEXT) COLOR(GREEN) INTENS(HIGH)
^ TYPE(TEXT) COLOR(TURQ) INTENS(HIGH)
' TYPE(TEXT) COLOR(YELLOW) INTENS(HIGH)
; TYPE(TEXT) COLOR(RED) INTENS(HIGH)

```

```

? TYPE(TEXT) COLOR(PINK) INTENS(HIGH)
3 TYPE(TEXT) COLOR(RED) INTENS(HIGH) HILITE(REVERSE) /* hex fa */
w TYPE(TEXT) COLOR(RED) INTENS(HIGH) HILITE(USCORE) /* hex fb */
p TYPE(TEXT) COLOR(RED) INTENS(HIGH) HILITE(BLINK) /* hex fc */
)BODY EXPAND(@@) window(80,24)
%-@-@-;E-Mail Interface%-@-@-
%Command ==>_ZCMD
+
+
'Mail Title ==>-MEMOID
+
'Receivers Name ==>-MEMORC
'Question Mark =>

Directory
'Domain (optional) ==>-MEMODG
,
+
+
?Receiver and Domain:
? CC:Mail Europe:"enter receivers name. Domain defaults to;sc.ccm.com
? CC:Mail USA:"enter receivers name. Enter Domain as;usa"or;us.ccm.com
? TSO user in Mulp:"enter receivers TSO userid, Domain defaults to;mulp
? TSO user in other MVS:"enter receivers TSO userid and Domain=NJE-addr
? Internet:"enter users identification and domain name.
+
+
_DEBUG+
+
+ ?Enter:;Continue ?PF3:;Terminate
+
+
)INIT
&ZPRIM = NO
.CURSOR = MEMOID
&DEBUG = ' '
&RESP = ' '
&MEMORC = ' '
&MEMODG = ' '
)REINIT
REFRESH(*)
.ATTR (.CURSOR) = 'COLOR(RED) HILITE(REVERSE)'
&RESP = ' '
)PROC
REFRESH(*)
VER (&MEMOID, NONBLANK, MSG=INST346)
IF (&MEMOID = ' ')
.CURSOR = MEMOID
VER (&MEMORC, NONBLANK, MSG=INST347)
IF (&MEMORC = ' ')
.CURSOR = MEMORC
&DM = ' '

```

```

IF (&MEMODG ^= ' ')
    &DM = 'DOMAIN('&MEMODG'' )'
IF (.RESP ^= ENTER)
    &RESP = CANCEL
VPUT (RESP)
VPUT (MEMOID, MEMORC, MEMODG, DM)
VPUT (MEMOID) PROFILE
)END

```

PANEL INST81B

Called from CLIST (edit macro) ZCC to get cc-receiver(s) and corresponding domain(s):

```

)ATTR
! TYPE(OUTPUT) INTENS(HIGH) CAPS(ON) JUST(LEFT) PAD(' ')
\ TYPE(TEXT) INTENS(LOW) CAPS(ON) JUST(LEFT) PAD(' ') SKIP(ON)
~ TYPE(INPUT) COLOR(RED) CAPS(OFF) JUST(LEFT) PAD('_')
TYPE(INPUT) INTENS(LOW) CAPS(ON) JUST(RIGHT) PAD(' ')
< TYPE(INPUT) INTENS(NON)
" TYPE(TEXT) COLOR(GREEN) INTENS(HIGH)
^ TYPE(TEXT) COLOR(TURQ) INTENS(HIGH)
' TYPE(TEXT) COLOR(YELLOW) INTENS(HIGH)
; TYPE(TEXT) COLOR(RED) INTENS(HIGH)
? TYPE(TEXT) COLOR(PINK) INTENS(HIGH)
3 TYPE(TEXT) COLOR(RED) INTENS(HIGH) HILITE(REVERSE) /* hex fa */
w TYPE(TEXT) COLOR(RED) INTENS(HIGH) HILITE(USCORE) /* hex fb */
p TYPE(TEXT) COLOR(RED) INTENS(HIGH) HILITE(BLINK) /* hex fc */
)BODY EXPAND(@@) window(80,24)
%-@-@-;E-Mail Interface%-@-@-
%Command ==>_ZCMD
+
+
+
+
'Receivers Name ==>~MEMOCC
'Question Mark =>
Directory
'Domain (optional) ==>~CCMEMODG
,
+
+
?Receiver and Domain:
? CC:Mail Europe:"enter receivers name. Domain defaults to;sc.ccm.com
? CC:Mail USA:"enter receivers name. Enter Domain as;usa"or;us.ccm.com
? TSO user in MULP:"enter receivers TSO userid, Domain defaults to;mulp
? TSO user in other MVS:"enter receivers TSO userid and Domain=NJE-addr
? Internet:"enter users identification and domain name.
+
+

```

```

_DEBUG+
+
+ ?Enter;;Continue ?PF3;;Terminate
+
+
)INIT
  &ZPRIM = NO
  .CURSOR = MEMOCC
  &DEBUG = ' '
  &RESP = ' '
  IF (&MEMOCC = ' ')
    &CCMEMODG = ' '
  IF (&MEMOCC = '?')
    .RESP = 'ENTER'
)REINIT
  REFRESH(*)
  .ATTR (.CURSOR) = 'COLOR(RED) HILITE(REVERSE)'
  &RESP = ' '
)PROC
  REFRESH(*)
  VER (&MEMOCC, NONBLANK, MSG=INST347)
  IF (&MEMOCC = ' ')
    .CURSOR = MEMOCC
  &CCDM = &CCMEMODG
  IF (.RESP ≠ ENTER)
    &RESP = CANCEL
  VPUT (RESP)
  VPUT (MEMOCC,CCDM)
)END

```

ISPF MESSAGE MEMBER INST34

It contains the relevant messages used in the mailbox routines:

```

INST340 'FUNCTION INTERRUPTED ' .ALARM=YES
'FUNCTION &SUBFUNC IN PROGRESS WAS INTERRUPTED.'

INST341 'NO SELECTION MADE ' .TYPE=ACTION .WINDOW=NORESP
'PLEASE ENTER ONE OF THE SELECTABLE FUNCTIONS FROM MENU.'

INST342 'DATASET ↯ PARTITIONED' .TYPE=ACTION .WINDOW=NORESP
'SMTP MAILBOX DATASET MUST BE PARTITIONED.'

INST343 'DATASET NOT AVAILABLE' .TYPE=ACTION .WINDOW=NORESP
'YOU ARE PROBABLY LOOKING FOR AN UNDEFINED DATASET.'

INST344 'ENTER DATASET NAME ' .TYPE=ACTION .WINDOW=NORESP
'ENTER NAME OF TSO DATASET TO CONTAIN MAIL.'

INST345 'ENTER MEMBER NAME ' .TYPE=ACTION .WINDOW=NORESP

```

'ENTER MEMBER NAME IN TSO DATASET TO CONTAIN MAIL.'

INST346 'ENTER MAIL ID ' .TYPE=ACTION .WINDOW=NORESP
'ENTER NAME OR SUBJECT OF MAIL YOU WANT TO PROCESS.'

INST347 'ENTER RECEIVER (USER)' .TYPE=ACTION
'SPECIFY THE USERID WHO SHALL RECEIVE THIS MAIL.'

INST348 'RECORD LENGTH \rightarrow 255. ' .TYPE=ACTION .WINDOW=NORESP
'RECORD LENGTH OF SMTP MAILBOX DATASET MUST BE 255.'

INST349 'NO MAIL TO RECEIVE ' .TYPE=WARNING .WINDOW=NORESP
'SORRY THERE WAS NO MAIL TO RECEIVE'

Editor's note: this article will be continued in the next issue.

Nils Plum
Systems Programmer (Denmark)

© Xephon 1998

Implementing a Wide Area Network

The RAC is a UK-based motoring organization, which, apart from its breakdown service, also offers legal, technical, and other motoring-related services. As a round-the-clock business, it is heavily dependent on information technology. Thanks to the replacement of three Wide Area Networks (WANs) by one routed SMDS network, the RAC is better equipped than ever before. And with a return on investment of around a year, the financial savings are substantial.

Operating throughout the UK, the RAC has a number of regional Command Centres and other offices. Although physical separation between offices is common for many businesses, using different IT platforms across locations does present problems. Relying on three different WANs can make life difficult indeed. The RAC's three networks have recently become a single, scalable infrastructure.

THREE WAY SERVICE

Developed over several years, the three WANs had entirely different purposes. An SNA network served an IBM 3090 mainframe, used for

the membership system. Problems included Front End Processor (FEP) capacity constraints, line utilization mis-matches, and spiralling costs for more bandwidth.

A second RAC network, running DECNET protocols, supported frontline rescue services. Based on a fully-meshed design, it provided staff in five RAC Command Centres with the CARS (Computer Aided Rescue Services) application. Although it is highly resilient, congestion problems were beginning to be apparent.

Thirdly, the fast-growing 'Enterprise' network supported pensions and payroll, Lotus ccMail, personnel, and document image processing users. Built around Cisco 2500 routers, a single point of failure and capacity constraints were the main concerns.

STRATEGIC REVIEW

Issues included difficult information sharing, high cost of ownership, and duplicated network support. Lack of bandwidth management, poor resilience, and under-utilization also figured. LANs were based on Token Ring and Ethernet with IP, IPX, SNA, and DECNET protocols, all adding to the support workload.

In developing the RAC's new WAN strategy, cost-savings were top-of-the-list. Installing more circuits was impractical and expensive. The RAC wanted a single location-independent network, allowing anywhere-to-anywhere user connections.

To find a solution, the project was effectively split into two areas. Three WANs would become one. Existing Kilostream and FBS (Flexible Bandwidth Services) bearer circuits would be replaced with a single technology. The core issue was swiftly identified – the key to the whole project was the IBM mainframe and how it was integrated into the new network.

By using a Cisco router with a Channel Interface Processor card (CIP), the mainframe's Front End Processor (FEP) could be replaced. Through this direct connection to the mainframe channel, services would be delivered using TCP/IP protocols to routers on remote sites. Use of multiple routers and ISDN standby links would provide fully automatic resilience.

SMDS, ATM, and Frame Relay services were then reviewed. Switched Multimegabit Data Services (SMDS) offered a connectionless approach and reduced network and administration costs, as well as an easy migration path. It would also provide scalable bandwidth for an any-to-any client/server, multiprotocol network for all current and future applications.

PROJECT

By November 1996, the project was underway. All the equipment was tested and pre-configured by Chernikeeff before being installed. The Enterprise Network was the first to be migrated. SMDS was deployed and connected to new routers, and the old routes were closed down. Next on the list was the SNA network. The mainframe was connected to a Cisco 7500 router via a CIP card with other SMDS-linked routers deployed on remote sites.

Attention then focused on the mission-critical CARS network. Migration was over a cautious seven-month period to avoid disruption to members.

HIGH AVAILABILITY

Using IBM NetView 6000 and CiscoWorks, network management is now straightforward. Should any SMDS circuits go down, users are transparently switched to ISDN. Router failures are accommodated too.

With the new network delivering “extremely high levels of flexibility” for 2,500 users, the RAC’s savings will top £600,000 in just four years. The return on investment is around a year. This has helped along other projects including new switched Ethernet LANs, a Cisco AS5200 remote access router for dial-up users, server migrations, and disaster recovery planning.

Charles Fountain
Marketing Manager
Chernikeeff Networks Ltd

© Chernikeeff Networks Ltd 1998

Link between NCPs – part 2

This month we continue the article describing a method that allows ten computers to be linked using a switched link.

```
L690000L LINE ADDRESS=(5,HALF), LIGNE SDLC (3174) *
SERVLIM=17, *
ISTATUS=ACTIVE, *
SPEED=19200 (CLOCKNG=EXT)
* STATOPT='SDLC*LIGNE 17 HALF DUPLEX 19200'
C690000L PU ADDR=01, *
DISCNT=NO, PU STAYS ACTIVE WITHOUT LU CONNECTION *
MAXDATA=521, MAX DATA PER PIU *
MAXOUT=4, MAX. NO. OF PIU SENT WITHOUT RESPONSE *
PASSLIM=7, *
PUDR=YES, *
LOGAPPL=A695CICS, *
PUTYPE=2, *
VPACING=4 ENTER HOST AND 3745
* STATOPT='PU GESTION L '
T690000EQ LU LOCADDR=02, *
LUDR=YES
* STATOPT='PU GESTION E '
T690000ER LU LOCADDR=03, *
LUDR=YES
* STATOPT='PU GESTION E '
T690000ES LU LOCADDR=04, *
LUDR=YES
* STATOPT='PU GESTION E '
T690000CW LU LOCADDR=05, *
LUDR=YES
* STATOPT='PU GESTION C '
T690000DQ LU LOCADDR=06, *
LUDR=YES
* STATOPT='PU GESTION D '
T690000FF LU LOCADDR=07, *
LUDR=YES
* STATOPT='PU LIBRE '
T690000DR LU LOCADDR=08, *
LUDR=YES
* STATOPT='PU GESTION D '
T690000DS LU LOCADDR=09, *
LUDR=YES
* STATOPT='PU GESTION E '
T690000FI LU LOCADDR=10, *
LUDR=YES
* STATOPT='PU AUDIT '
T690000IJ LU LOCADDR=11, *
LUDR=YES
```

```

*          STATOPT='PU LO          '
T690000IK LU LOCADDR=12,          *
          LUDR=YES
*          STATOPT='PU L*LO          '
T690000GL LU LOCADDR=13,          *
          LUDR=YES
*          STATOPT='PU L*GESTION D'
T690000GM LU LOCADDR=14,          *
          LUDR=YES
*          STATOPT='PU L*GUICHET  '
T690000GN LU LOCADDR=15,          *
          LUDR=YES
*          STATOPT='PU L*STANDARD  '
T690000KO LU LOCADDR=16,          *
          LUDR=YES
*          STATOPT='PU L*KAISSE    '
T690000KP LU LOCADDR=17,          *
          LUDR=YES
*          STATOPT='PU *GESTION    '
*
*****
**          LIGNES /SDLC 3270 / FORMATION C. I. C. S.          **
**          15 SCREENS + ONE PRINTER E. S. S. A. I.          **
*****
*
L69000000 LINE ADDRESS=(27,HALF),   LINE SDLC (3174)          *
          SERVLIM=17,              *
          ISTATUS=ACTIVE,          *
          SPEED=9600                (CLOCKNG=EXT)              *
*          STATOPT='SDLC*LIGNE 27 '
C69000000 PU ADDR=01,              *
          DISCNT=NO,                PU STAYS ACTIVE WITHOUT LU CONNECTION *
          MAXDATA=265,              MAX DATA PER PIU          *
          MAXOUT=7,                 MAX. NO. OF PIU SENT WITHOUT RESPONSE *
          PASSLIM=7,                *
          PUDR=YES,                 *
          USSTAB=USSTAB3C,          *
          PUTYPE=2,                  *
          VPACING=8                  ENTER HOST AND 3745
*          STATOPT=' *FORMATION'
T6900000A LU LOCADDR=02,           LU ECRAN                  *
          LUDR=YES
*          STATOPT=' *FORMATION'
T6900000B LU LOCADDR=03,           LU ECRAN                  *
          LUDR=YES
*          STATOPT=' *FORMATION'
T6900000C LU LOCADDR=04,           LU ECRAN                  *
          LUDR=YES
*          STATOPT=' *FORMATION'
T6900000D LU LOCADDR=05,           LU ECRAN                  *
          LUDR=YES

```

```

*          STATOPT=' *FORMATION'
T690000E LU LOCADDR=06,      LU ECRAN          *
          LUDR=YES
*          STATOPT=' *FORMATION'
T690000F LU LOCADDR=07,      LU ECRAN          *
          LUDR=YES
*          STATOPT=' *FORMATION'
T690000G LU LOCADDR=08,      LU ECRAN          *
          LUDR=YES
*          STATOPT=' *FORMATION'
T690000H LU LOCADDR=09,      LU ECRAN          *
          LUDR=YES
*          STATOPT=' *FORMATION'
T690000I LU LOCADDR=10,      LU ECRAN          *
          LUDR=YES
*          STATOPT=' *FORMATION'
T690000J LU LOCADDR=11,      LU ECRAN          *
          LUDR=YES
*          STATOPT=' *FORMATION'
T690000K LU LOCADDR=12,      LU ECRAN          *
          LUDR=YES
*          STATOPT=' *FORMATION'
T690000L LU LOCADDR=13,      LU ECRAN          *
          LUDR=YES
*          STATOPT=' *FORMATION'
T690000M LU LOCADDR=14,      LU ECRAN          *
          LUDR=YES
*          STATOPT=' *FORMATION'
T690000N LU LOCADDR=15,      LU ECRAN          *
          LUDR=YES
*          STATOPT=' *FORMATION'
T690000O LU LOCADDR=16,      LU ECRAN          *
          LUDR=YES
*          STATOPT=' *FORMATION'
T690000P LU LOCADDR=17,      LU IMPRIMANTE      *
          DLOGMOD=SCS3287,MODETAB=BIND3287,    *
          PACING=7,          *
          LUDR=YES          *
*          STATOPT='FORM*LIBRE      '
*
*-----*
*   LIGNE S N A      C. I. C. S.   R. E. E. L.      *
*   CONTROLEUR S : 16 SCREENS          *
*-----*
L690000S LINE ADDRESS=(3,FULL), LIGNE SDLC (3174)   *
          SERVLIM=17,          *
          ISTATUS=ACTIVE,      *
          SPEED=19200          (CLOCKNG=EXT)
*          STATOPT='SDLC*LIGNE 3 FULL DUPLEX'
C690000S PU ADDR=01,          *
          DISCNT=NO,          PU STAYS ACTIVE WITHOUT LU CONNECTION *

```

```

MAXDATA=521,      MAX DATA PER PIU      *
MAXOUT=4,         MAX. NO. OF PIU SENT WITHOUT RESPONSE *
PASSLIM=7,
PUDR=YES,
LOGAPPL=A695CICS,
PUTYPE=2,
VPACING=4        ENTRE HOST ET 3745
*
T69000SA LU      STATOPT='PU S*FORMATION'
LOCADDR=02,
LUDR=YES
*
T69000SB LU      STATOPT='PU S*SCR      '
LOCADDR=03,
LUDR=YES
*
T69000KC LU      STATOPT='PU S*SCR      '
LOCADDR=04,
LUDR=YES
*
T69000OQ LU      STATOPT='PU S*SAISIE  '
LOCADDR=05,
LUDR=YES
*
T69000OR LU      STATOPT='PU S*OC      '
LOCADDR=06,
LUDR=YES
*
T69000OS LU      STATOPT='PU S*OC      '
LOCADDR=07,
LUDR=YES
*
T69000SH LU      STATOPT='PU S*OC      '
LOCADDR=08,
LUDR=YES
*
T69000RG LU      STATOPT='PU S*LIBRE  '
LOCADDR=09,
LUDR=YES
*
T69000NQ LU      STATOPT='PU S*SCR      '
LOCADDR=10,
LUDR=YES
*
T69000NR LU      STATOPT='PU S*COMPTA. '
LOCADDR=11,
DLOGMOD=M32782M,      MICRO COMPTA
LUDR=YES,
LOGAPPL=A691TS
*
T69000NS LU      STATOPT='S MIC*COMPTA. '
LOCADDR=12,
LUDR=YES
*
T69000NT LU      STATOPT='PU S*COMPTA. '
LOCADDR=13,
LUDR=YES
*
T69000NU LU      STATOPT='PU S*COMPTA. '
LOCADDR=14,
LUDR=YES
*
T69000ET LU      STATOPT='PU S*COMPTA. '
LOCADDR=15,
LUDR=YES

```

```

*          STATOPT='PU S*GESTION E'
T690000EU LU LOCADDR=16,
LUDR=YES
*          STATOPT='PU S*GESTION E'
T690000EV LU LOCADDR=17,
LUDR=YES
*          STATOPT='PU S*GESTION E'
*
*-----*
*   LIGNE S N A      C. I. C. S.   R. E. E. L.
*-----*
L690000F LINE ADDRESS=(1,FULL),  LIGNE SDLC (3174)
SERVLIM=17,
ISTATUS=ACTIVE,
SPEED=19200      (CLOCKNG=EXT)
*          STATOPT='SDLC*LIGNE 11 '
C690000F PU  ADDR=01,
DISCNT=NO,      PU STAYS ACTIVE WITHOUT LU CONNECTION
MAXDATA=265,    MAX DATA PER PIU
MAXOUT=7,      MAX. NO. OF PIU SENT WITHOUT RESPONSE
PASSLIM=7,
PUDR=YES,
LOGAPPL=A695CICS,
PUTYPE=2,
VPACING=8      ENTRE HOST ET 3745
*          STATOPT='PU F*GESTION A'
T690000AQ LU LOCADDR=02,
LUDR=YES
*          STATOPT='PU F*GESTION A'
T690000AR LU LOCADDR=03,
LUDR=YES
*          STATOPT='PU F*GESTION A'
T690000AS LU LOCADDR=04,
LUDR=YES
*          STATOPT='PU F*GESTION A'
T690000AT LU LOCADDR=05,
LUDR=YES
*          STATOPT='PU F*GESTION A'
T690000AU LU LOCADDR=06,
LUDR=YES
*          STATOPT='PU F*GESTION A'
*
T690000BQ LU LOCADDR=07,
LUDR=YES
*          STATOPT='PU F*GESTION B'
T690000BR LU LOCADDR=08,
LUDR=YES
*          STATOPT='PU F*GESTION B'
T690000BS LU LOCADDR=09,
LUDR=YES
*          STATOPT='PU F*GESTION B'

```

```

T69000BT LU      LOCADDR=10,
                  LUDR=YES
*
* STATOPT='PU F*GESTION B'
T69000BU LU      LOCADDR=11,
                  LUDR=YES
*
* STATOPT='PU F*GESTION B'
*
T69000CQ LU      LOCADDR=12,
                  LUDR=YES
*
* STATOPT='PU F*GESTION C'
T69000CR LU      LOCADDR=13,
                  LUDR=YES
*
* STATOPT='PU F*GESTION C'
T69000CS LU      LOCADDR=14,
                  LUDR=YES
*
* STATOPT='PU F*GESTION C'
T69000CT LU      LOCADDR=15,
                  LUDR=YES
*
* STATOPT='PU F*GESTION C'
T69000CU LU      LOCADDR=16,
                  LUDR=YES
*
* STATOPT='PU F*GESTION C'
T69000CV LU      LOCADDR=17,
                  LUDR=YES
*
* STATOPT='PU F*GESTION C'
*
*-----*
*   LIGNE S N A M   S D L C SAISIE KEYMASTER
*-----*
L690000M LINE    ADDRESS=(2,FULL),   LIGNE SDLC (3174)
                  SERVLIM=17,
                  ISTATUS=ACTIVE,
                  SPEED=9600          (CLOCKNG=EXT)
*
* STATOPT='SDLC*LIGNE 2 FULL DUPLEX'
C690000M PU      ADDR=01,
                  DISCNT=NO,          PU STAYS ACTIVE WITHOUT LU CONNECTION
                  MAXDATA=265,        MAX DATA PER PIU
                  MAXOUT=7,           MAX. NO. OF PIU SENT WITHOUT RESPONSE
                  PASSLIM=7,
                  PUDR=YES,
                  LOGAPPL=A699CICS,
                  PUTYPE=2,
                  VPACING=8           ENTRE HOST ET 3745
*
* STATOPT='      *SAISIE   '
T69000MA LU      LOCADDR=02,
                  LUDR=YES
*
* STATOPT='      *SAISIE   '
T69000MB LU      LOCADDR=03,
                  LUDR=YES
*
* STATOPT='      *SAISIE   '
T69000MC LU      LOCADDR=04,

```

```

*          LUDR=YES
T69000MD LU STATOPT=' *SAISIE ' *
          LOCADDR=05,
          LUDR=YES
*          STATOPT=' *SAISIE ' *
T69000ME LU LOCADDR=06,
          LUDR=YES
*          STATOPT=' *SAISIE ' *
T69000MF LU LOCADDR=07,
          LUDR=YES
*          STATOPT=' *SAISIE ' *
T69000MG LU LOCADDR=08,
          LUDR=YES
*          STATOPT=' *SAISIE ' *
T69000MH LU LOCADDR=09,
          LUDR=YES
*          STATOPT=' *SAISIE ' *
T69000MI LU LOCADDR=10,
          LUDR=YES
*          STATOPT=' *SAISIE ' *
T69000MJ LU LOCADDR=11,
          LUDR=YES
*          STATOPT=' *SAISIE ' *
T69000MK LU LOCADDR=12,
          LUDR=YES
*          STATOPT=' *SAISIE ' *
T69000ML LU LOCADDR=13,
          LUDR=YES
*          STATOPT=' *SAISIE ' *
T69000MM LU LOCADDR=14,
          LUDR=YES
*          STATOPT=' *SAISIE ' *
T69000MN LU LOCADDR=15,
          LUDR=YES
*          STATOPT=' *SAISIE ' *
T69000MO LU LOCADDR=16,
          LUDR=YES
*          STATOPT=' *SAISIE ' *
T69000MP LU LOCADDR=17,          LU IMPRIMANTE *
          DLOGMOD=SCS3287,MODETAB=BIND3287, *
          PACING=7, *
          LUDR=YES
*          STATOPT='IMP *SAISIE '
*
*-----*
*   LIGNE S N A *   C. I. C. S.   R. E. E. L.   **   S D L C   *
*-----*
*   OC : 16 SCREENS 2 PRINTERS *
*-----*
L6900000 LINE ADDRESS=(10,HALF), LINE SDLC (3174) *
          SERVLIM=17, *

```



```

          ISTATUS=ACTIVE,
          SPEED=9600      (CLOCKNG=EXT)
*
C6900000 PU  STATOPT='SDLC*LIGNE 10 '
          ADDR=01,
          DISCNT=NO,      PU STAYS ACTIVE WITHOUT LU CONNECTION
          MAXDATA=521,    MAX DATA PER PIU
          MAXOUT=4,      MAX. NO. OF PIU SENT WITHOUT RESPONSE
          PASSLIM=7,
          PUDR=YES,
          LOGAPPL=A695CICS,
          PUTYPE=2,
          VPACING=4      ENTER HOST AND 3745
*
T6900000A LU STATOPT=' *OC *ARCH'
          LOCADDR=02,
          LUDR=YES
*
T6900000B LU STATOPT=' *OC '
          LOCADDR=03,
          LUDR=YES
*
T6900000C LU STATOPT=' *OC '
          LOCADDR=04,
          LUDR=YES
*
T6900000D LU STATOPT=' *OC '
          LOCADDR=05,
          LUDR=YES
*
T6900000E LU STATOPT=' *OC '
          LOCADDR=06,
          LUDR=YES
*
T6900000F LU STATOPT=' *OC '
          LOCADDR=07,
          LUDR=YES
*
T6900000G LU STATOPT=' *OC '
          LOCADDR=08,
          LUDR=YES
*
T6900000H LU STATOPT=' *OC '
          LOCADDR=09,
          LUDR=YES
*
T6900000I LU STATOPT=' *OC '
          LOCADDR=10,
          LUDR=YES
*
T6900000J LU STATOPT=' *OC '
          LOCADDR=11,
          LUDR=YES
*
T6900000K LU STATOPT=' *OC '
          LOCADDR=12,
          DLOGMOD=SCS3287,MODETAB=BIND3287,
          PACING=7,
          LUDR=YES
*
T6900000L LU STATOPT='IMP *OC *ARCH'
          LOCADDR=13,
          LUDR=YES

```

```

*          STATOPT=' *RM      '
T690000M LU LOCADDR=14,
LUDR=YES
*          STATOPT=' *RM      '
T690000N LU LOCADDR=15,
LUDR=YES
*          STATOPT=' *RM      '
T690000P LU LOCADDR=16,
LUDR=YES
*          STATOPT=' *OC      '
T690000P LU LOCADDR=17,
DLOGMOD=SCS3287,MODETAB=BIND3287,
PACING=7,
LUDR=YES
*          STATOPT='IMP *OC *ARCH'
*-----*
*   L I G N E   S   N   A   Q   S   D   L   C   L   I   B   R   E
*-----*
L690000Q LINE ADDRESS=(24,HALF), LIGNE SDLC (3174)
SERVLIM=17,
ISTATUS=INACTIVE,
SPEED=9600 (CLOCKNG=EXT)
*          STATOPT='SDLC*LIGNE 24 '
C690000Q PU ADDR=01,
DISCNT=NO, PU STAYS ACTIVE WITHOUT LU CONNECTION
MAXDATA=265, MAX DATA PER PIU
MAXOUT=7, MAX. NO. OF PIU SENT WITHOUT RESPONSE
PASSLIM=7,
PUDR=YES,
PUTYPE=2,
USSTAB=USSTAB3C,
VPACING=8 ENTRE HOST ET 3745
*          STATOPT='PU Q*LIBRE '
T690000A LU LOCADDR=02,
LUDR=YES
*          STATOPT='PU Q*LIBRE '
T690000B LU LOCADDR=03,
LUDR=YES
*          STATOPT='PU Q*LIBRE '
T690000C LU LOCADDR=04,
LUDR=YES
*          STATOPT='PU Q*LIBRE '
T690000D LU LOCADDR=05,
LUDR=YES
*          STATOPT='PU Q*LIBRE '
T690000E LU LOCADDR=06,
LUDR=YES
*          STATOPT='PU Q*LIBRE '
T690000F LU LOCADDR=07,
LUDR=YES

```

```

*          STATOPT='PU Q*LIBRE      '
T690000QG LU LOCADDR=08,
              LUDR=YES
*          STATOPT='PU Q*LIBRE      '
T690000QH LU LOCADDR=09,
              LUDR=YES
*          STATOPT='PU Q*LIBRE      '
T690000QI LU LOCADDR=10,
              LUDR=YES
*          STATOPT='PU Q*LIBRE      '
T690000QJ LU LOCADDR=11,
              LUDR=YES
*          STATOPT='PU Q*LIBRE      '
T690000QK LU LOCADDR=12,
              LUDR=YES
*          STATOPT='PU Q*LIBRE      '
T690000QL LU LOCADDR=13,
              LUDR=YES
*          STATOPT='PU Q*LIBRE      '
T690000QM LU LOCADDR=14,
              LUDR=YES
*          STATOPT='PU Q*LIBRE      '
T690000QN LU LOCADDR=15,
              LUDR=YES
*          STATOPT='PU Q*LIBRE      '
T690000QO LU LOCADDR=16,
              LUDR=YES
*          STATOPT='PU Q*LIBRE      '
T690000QP LU LOCADDR=17,
              DLOGMOD=SCS3287,MODETAB=BIND3287,
              PACING=7,
              LUDR=YES
*          STATOPT='IMP *OQ  *REUN'
*-----*
*  LIGNE S N A      POUR SAS GRAPHIQUE      LIGNE 18      *
*  LIGNE S N A      POINT A POINT POUR MICRO  LIGNE 19      *
*  LIGNE S N A      (SAS GRAPHIQUE)          *
*  LIGNE S N A      *
*  LIGNE S N A      MICRO SAS                *
*  LIGNE S N A      *
*-----*
*-----*
*  LIGNE S N A      POUR SAS GRAPHIQUE      LIGNE 17      *
*-----*
*-----*
L69000017 LINE ADDRESS=(17,HALF), LIGNE SDLC (3174)      *
              SERVLIM=17,                               *
              ISTATUS=INACTIVE,                         *
              SPEED=9600      (CLOCKNG=EXT)
*          STATOPT='SDLC*LIGNE 17  '

```

```

C6900017 PU   ADDR=01,                               *
              DISCNT=NO,      PU STAYS ACTIVE WITHOUT LU CONNECTION *
              MAXDATA=521,    MAX DATA PER PIU                    *
              MAXOUT=4,       MAX. NO. OF PIU SENT WITHOUT RESPONSE *
              PASSLIM=7,                                           *
              PUDR=YES,                                           *
              PUTYPE=2,                                           *
              USSTAB=USSTAB3C,                                     *
              VPACING=4      ENTRE HOST ET 3745
*
M69PC17A LU   STATOPT='PU17*LIBRE      '
              LOCADDR=02,                                         *
              LUDR=YES,                                           *
              MODETAB=SLC32XX,DLOGMOD=M3192G3
*
I69PC17A LU   STATOPT='PU 17*MICRO SAS GRAPHIQUE '
              LOCADDR=03,MODETAB=SLC328X,DLOGMOD=G328XP2
*-----*
*-----*
*   LIGNE S N A      POUR SAS GRAPHIQUE      LIGNE 18      *
*-----*
*-----*
L6900018 LINE ADDRESS=(18,HALF),  LIGNE SDLC (DYNR)          *
              SERVLIM=50,                                         *
              CLOCKNG=EXT,      MODEM DONNE LA VITESSE          *
              SPEED=9600      (CLOCKNG=EXT)
*
C6900018 PU   STATOPT='SDLC*LIGNE 18 '
              ADDR=01,                               *
              DISCNT=NO,      PU STAYS ACTIVE WITHOUT LU CONNECTION *
              MAXDATA=521,    MAX DATA PER PIU                    *
              MAXOUT=4,       MAX. NO. OF PIU SENT WITHOUT RESPONSE *
              PASSLIM=7,                                           *
              PUDR=YES,                                           *
              PUTYPE=2,                                           *
              USSTAB=USSTAB3D,                                     *
              VPACING=4      ENTER HOST AND 3745
*
M69P1802 LU   STATOPT='PU18*LIBRE      '
              LOCADDR=02,                                         *
              LUDR=YES,                                           *
              MODETAB=SLC32XX,DLOGMOD=M3192G3
*
              STATOPT='PU 19*MICRO SAS GRAPHIQUE PORT 02'

```

Editor's note: in the original code, the four lines above are repeated for each of the numbers in the range 03 to 99.

```

M6908100 LU   LOCADDR=100,                                     *
              LUDR=YES,                                           *
              MODETAB=SLC32XX,DLOGMOD=M3192G3
*
              STATOPT='PU 19*MICRO SAS GRAPHIQUE PORT 100'

```

Editor's note: in the original code, the four lines above are repeated for each of the numbers in the range 101 to 127.

```

M6908128 LU    LOCADDR=128,
                LUDR=YES,
                MODETAB=SLC32XX,DLOGMOD=M3192G3
*              STATOPT='PU 19*MICRO SAS GRAPHIQUE PORT 128'
*-----*
*-----*
*   LIGNE S N A      POUR SAS GRAPHIQUE      LIGNE 19      *
*               POINT A POINT                *
*-----*
*-----*
L6900019 LINE  ADDRESS=(19,HALF),  LIGNE SDLC (DYNR)      *
                SERVLIM=17,                *
                CLOCKNG=DIRECT,            *
                ISTATUS=ACTIVE,           *
                SPEED=9600                NE SERT A RIEN (CLOCKNG=EXT)
*              STATOPT='SDLC*LIGNE 19 '
M69PU19  PU    ADDR=01,                ADRESSE  SDLC      *
                VPACING=3,                PACING  VTAM      *
                MAXOUT=7,                MAX DE PIU POUR UN MESSAGE
                PACING=0,                *
                PASSLIM=7,                NOMBRE DE PIU CONTIGUES EMISES
                PUTYPE=2                  *
M69P1902 LU    LOCADDR=02,                *
                LUDR=YES,                *
                MODETAB=SLC32XX,DLOGMOD=M3192G3
*              STATOPT='PU 19*MICRO SAS GRAPHIQUE PORT 02'

```

Editor's note: in the original code, the four lines above are repeated for each of the numbers in the range 03 to 99.

```

M6909100 LU    LOCADDR=100,                *
                LUDR=YES,                *
                MODETAB=SLC32XX,DLOGMOD=M3192G3
*              STATOPT='PU 19*MICRO SAS GRAPHIQUE PORT 100'
M6909101 LU    LOCADDR=101,                *
                LUDR=YES,                *
                MODETAB=SLC32XX,DLOGMOD=M3192G3
*              STATOPT='PU 19*MICRO SAS GRAPHIQUE PORT 101'

```

Editor's note: in the original code, the four lines above are repeated for each of the numbers in the range 102 to 127.

```

M6909128 LU    LOCADDR=128,                *
                LUDR=YES,                *
                MODETAB=SLC32XX,DLOGMOD=M3192G3
*              STATOPT='PU 19*MICRO SAS GRAPHIQUE PORT 128'
*-----*
*-----*
*---          NEW LINE NOV 1996 FOR RUE DESAIX 25 LIGNE          *
*-----*

```

```

*--      LIGNE 22 SERVICE COMPTE MULTIPLE-----*
*-----*
*-----*
*-----*
*-----*
*-----*
*-----*
L690000H LINE ADDRESS=(22,HALF), LIGNE SDLC (25 RUSE DESAIX CT MUL) *
SERVLIM=17, *
ISTATUS=ACTIVE, *
SPEED=9600 (CLOCKNG=EXT)
* STATOPT='SDLC*LIGNE 22 HALFDUPLEX'
C690000H PU ADDR=01, *
DISCNT=NO, PU STAYS ACTIVE WITHOUT LU CONNECTION *
MAXDATA=521, MAX DATA PER PIU *
MAXOUT=4, MAX. NO. OF PIU SENT WITHOUT RESPONSE *
PASSLIM=7, *
PUDR=YES, *
LOGAPPL=A695CICS, *
PUTYPE=2, *
VPACING=4 ENTER HOST AND 3745
* STATOPT='PU GESTION H '
T690000HA LU LOCADDR=02, *
LUDR=YES
* STATOPT='PU GESTION H '
T690000HB LU LOCADDR=03, *
LUDR=YES
* STATOPT='PU GESTION H '
T690000HC LU LOCADDR=04, *
LUDR=YES
* STATOPT='PU GESTION H '
T690000HD LU LOCADDR=05, *
LUDR=YES
* STATOPT='PU GESTION H '
T690000HE LU LOCADDR=06, *
LUDR=YES
* STATOPT='PU GESTION H '
T690000HF LU LOCADDR=07, *
LUDR=YES
* STATOPT='PU GESTION H '
T690000HG LU LOCADDR=08, *
LUDR=YES X
* STATOPT='PU GESTION H '
T690000HH LU LOCADDR=09, *
LUDR=YES
* STATOPT='PU GESTION H '
T690000HI LU LOCADDR=10, *
LUDR=YES
* STATOPT='PU GESTION H '
T690000HJ LU LOCADDR=11, *
LUDR=YES

```

```

*          STATOPT='PU GESTION H '
T69000HK LU LOCADDR=12,
LUDR=YES
*          STATOPT='PU S*GESTION H '
T69000HL LU LOCADDR=13,
LUDR=YES
*          STATOPT='PU S*GESTION H '
T69000HM LU LOCADDR=14,
LUDR=YES
*          STATOPT='PU S*GESTION H '
T69000HN LU LOCADDR=15,
LUDR=YES
*          STATOPT='PU S*GESTION H '
T69000HO LU LOCADDR=16,
LUDR=YES
*          STATOPT='PU S*GESTION H '
T69000HP LU LOCADDR=17,
DLOGMOD=SCS3287,MODETAB=BIND3287,
PACING=7,
LUDR=YES
*          STATOPT='IMP *GESTION H '
***** X25 LYON *****
PRINT GEN
X25.PAD INDEX=1,PADPARAM=NULL
*
X25.NET NETTYPE=1,DM=YES,
CPHINDX=1,
OUHINDX=4
X25.VCCPT INDEX=1,
MAXPKTL=128,          PARAM. DE CONTROLE DE FLUX :
VWINDOW=3             LONGUEUR PAQUET.TAILLE FENETRE
*          INSLW=(25,0) DEF SLOWDOWN % (N EXITE PAS A CRETEIL)
X25.OUFT INDEX=1,
OPTFACL=0100          TAXATION AU DEMANDEUR
X25.OUFT INDEX=2,          FACILITE:
OPTFACL=0101          TAXATION AU DEMANDE
X25.OUFT INDEX=3,          FACILITE:
OPTFACL=0101          TAXATION AU DEMANDE
X25.OUFT INDEX=4          FACILITE: AUCUNE
MCH1 X25.MCH ADDRESS=0,          LIC (TYPE 1)
ANS=CONT,
LCGDEF=(0,15),          CANAUX LOGIQUES DE 0 A 17
FRMLGTH=131,          128+3 LONG. TRAME (MODULE 8)
MWINDOW=7,          TAILLE FENETRE AU NIVEAU TRAME
DSABLTO=3.0,          (DEF) DISABLE TIMEOUT
ENABLTO=3.0,          (DEF) ENABLE TIMEOUT
GATE=GENERAL,          POUR COMMUTE LLC4
LCN0=NOTUSED,          (DEF) SI NETTYPE = 1
LLCLIST=(LLC0,LLC2,LLC3,LLC4,LLC5), PROTOCOLE
PAD=INTEG,          SUPPORT PAD INTEGRE

```

```

TRAN=EVEN,          TRADUCTION ASCII EBCDIC PAD      *
LUNAME=MCH1LU,     LU MCH                          *
NCPGRP=G690X25M,   NOM MACRO GROUP QUI SERA GENEREE          *
NDRETRY=2,         DEFAULT                          *
NPRETRY=10,        DEFAULT                          *
PUNAME=MCH1PU,                                          *
SPEED=19200,                                             *
STATION=DTE,       DEFAULT                          *
TDTIMER=1,         DEFAULT                          *
ISTATUS=ACTIVE,                                        *
TPTIMER=1.6,       TEMPORISATEUR T1 TRANSPAC      *
SHM=YES,           SHORT HOLD MODE                 *
SVCINN=3,          NBRE DE LIENS INN                *
SDRTCNT=7,         NBRE DE RETRIES POUR SHM        *
SDRTIME=2          NBRE DE SECONDES ENTRE 2 RETRIES
*
X25.LCG LCGN=0
*
ZAI69001 X25.LINE TYPE=S,NCPGRP=G00INNA,            *
          DSTNODE=INN
PAI69001 X25.PU PUTYPE=4,ISTATUS=INACTIVE
*
ZAI69002 X25.LINE TYPE=S,NCPGRP=G00INNA,            *
          DSTNODE=INN
PAI69002 X25.PU PUTYPE=4,ISTATUS=INACTIVE
*
ZAI69003 X25.LINE TYPE=S,NCPGRP=G00INNA,            *
          DSTNODE=INN
PAI69003 X25.PU PUTYPE=4,ISTATUS=INACTIVE
*
*
LCVC      X25.VC   LCN=(1,15), 15 CVC CANAUX LOGIQUES 1 A 15 *
          TYPE=SWITCHED,  COMMUTE                    *
          NCPGRP=G00LX25,  NOM GROUPE MCH1 (ADD 00)   *
          OUFINDX=1,       UTILISE ENTREE 1 DANS OUFIT *
          VCCINDX=1,       UTILISE ENTREE 1 DANS VCCPT *
          CALL=INOUT
*****
*****      LIGNE 12 RNIS          64K      *****
*****
*****
*
*      DEFINITION DU RESEAU X25 DE LA LIGNE TRANSPAC 64 K *
*
*      ATTENTION IDBLKC 062 MCH3 ENQUETE INN *
*****
MCH3      X25.MCH ADDRESS=12,          SPEED 64 K *
          ANS=CONT,                    *
          DIRECT=NO,                   ATTACHEMENT EXTERNE *

```



```

DSABLTO=3.0,          (DEF) DISABLE TIMEOUT          *
ENABLTO=3.0,          (DEF) ENABLE TIMEOUT            *
FRMLGTH=131,          128+3 LONG. TRAME (MODULE 8)      *
GATE=GENERAL,          POUR COMMUTE LLC4                 *
ISTATUS=ACTIVE,       *
LCGDEF=(0,20),          CANAUX LOGIQUES DE 0 A 20        *
LCN0=NOTUSED,          (DEF) SI NETTYPE = 1          *
LLCLIST=(LLC0,LLC2,LLC3,LLC4,LLC5),  PROTOCOLE        *
LLC0=0,LLC2=2,LLC3=3,LLC4=4,LLC5=5,                  *
SUBADDR=YES,          *
IDBLKC=062,          *
LUNAME=MCH3LU,          LU MCH                          *
MWINDOW=7,            TAILLE FENETRE AU NIVEAU TRAME    *
NCPGRP=GX25M12B,      NOM MACRO GROUP QUI SERA GENERE   *
NDRETRY=2,            DEFAULT                          *
NPRETRY=10,           DEFAULT                          *
PAD=INTEG,            SUPPORT PAD INTEGRE                          *
PADINDX=1,            *
PUNAME=MCH3PU,        *
SPEED=65536,          *
STATION=DTE,          *
TPTIMER=1.6,          TEMPORISATEUR T1 TRANSPAC          *
TRAN=EVEN,            TRADUCTION ASCII EBCDIC PAD        *
SHM=YES,              SHORT HOLD MODE                          *
SVCINN=10,            NBRE DE LIENS INN                          *
SDRTCNT=7,            NBRE DE RETRIES POUR SHM            *
SDRTIME=2,            NBRE DE SECONDES ENTRE 2 RETRIES  *
X21NTWK=1984          RECOMM. 1984 DU CCITT                    *
LOGAPPL=A690TS,      *
*
*
*   GROUPE DE VOIES LOGIQUES
*
*       X25.LCG LCGN=0          NUMERO DE GROUPE DE CV
*
ZAI69121 X25.LINE TYPE=S,NCPGRP=G12INNX,          *
          DSTNODE=INN
PAI69121 X25.PU PUTYPE=4,ISTATUS=INACTIVE
*
ZAI69122 X25.LINE TYPE=S,NCPGRP=G12INNX,          *
          DSTNODE=INN
PAI69122 X25.PU PUTYPE=4,ISTATUS=INACTIVE
*
ZAI69123 X25.LINE TYPE=S,NCPGRP=G12INNX,          *
          DSTNODE=INN
PAI69123 X25.PU PUTYPE=4,ISTATUS=INACTIVE
*
ZAI69124 X25.LINE TYPE=S,NCPGRP=G12INNX,          *
          DSTNODE=INN
PAI69124 X25.PU PUTYPE=4,ISTATUS=INACTIVE
*

```

```

ZAI69125 X25.LINE TYPE=S,NCPGRP=G12INNX, *
          DSTNODE=INN
PAI69125 X25.PU PUTYPE=4,ISTATUS=INACTIVE
*
ZAI69126 X25.LINE TYPE=S,NCPGRP=G12INNX, *
          DSTNODE=INN
PAI69126 X25.PU PUTYPE=4,ISTATUS=INACTIVE
*
ZAI69127 X25.LINE TYPE=S,NCPGRP=G12INNX, *
          DSTNODE=INN
PAI69127 X25.PU PUTYPE=4,ISTATUS=INACTIVE
*
ZAI69128 X25.LINE TYPE=S,NCPGRP=G12INNX, *
          DSTNODE=INN
PAI69128 X25.PU PUTYPE=4,ISTATUS=INACTIVE
*
ZAI69129 X25.LINE TYPE=S,NCPGRP=G12INNX, *
          DSTNODE=INN
PAI69129 X25.PU PUTYPE=4,ISTATUS=INACTIVE
*
ZAI6912A X25.LINE TYPE=S,NCPGRP=G12INNX, *
          DSTNODE=INN
PAI6912A X25.PU PUTYPE=4,ISTATUS=INACTIVE
*
*       LIAISONS CVC
*
*       THE PU AND LU ARE DEFINED IN VTAMLST
*!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
*
          X25.VC LCN=(1,20),      20 CVC MIXED UTILISES *
          VCCINDX=1, *
          NCPGRP=G12LX25, *
          TYPE=SWITCHED, *
          OUFINDX=3, *
          CALL=INOUT
*!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
*
*****
*****
*****
*       PREMIERE LIAISON 9,6 X25 ADRESSE20 *
*****
MCH20    X25.MCH ADDRESS=20,      SPEED 9,6 HALF *
          ANS=CONT, *
          LCGDEF=(0,10),          CANAUX LOGIQUES DE 0 A 17 *
          FRMLGTH=131,           128+3 LONG. TRAME (MODULE 8) *
          MWINDOW=7,            TAILLE FENETRE AU NIVEAU TRAME *
          DSABLTO=3.0,          (DEF) DISABLE TIMEOUT *
          ENABLTO=3.0,          (DEF) ENABLE TIMEOUT *
          GATE=GENERAL,         POUR COMMUTE LLC4 *

```

```

LCNØ=NOTUSED,          (DEF) SI NETTYPE = 1          *
LLCLIST=(LLCØ,LLC2,LLC3,LLC4,LLC5),  PROTOCOLE          *
PAD=INTEG,             SUPPORT PAD INTEGRE          *
TRAN=EVEN,            TRADUCTION ASCII EBCDIC PAD    *
LUNAME=MCH2LU,        LU MCH                      *
NCPGRP=G692X25M,      NOM MACRO GROUP QUI SERA GENEREE *
NDRETRY=2,            DEFAULT                      *
NPRETRY=1Ø,           DEFAULT                      *
PUNAME=MCH2PU,
SPEED=96ØØ,
STATION=DTE,          DEFAULT                      *
TDTIMER=1,            DEFAULT                      *
ISTATUS=ACTIVE,
TPTIMER=1.6           TEMPORISATEUR T1 TRANSPAC          *
*
*   GROUPE DE VOIES LOGIQUES
*
*   X25.LCG LCGN=Ø           NUMBER OF GROUP  CV
*
*   LIAISONS CVC
*
*   X25.VC LCN=(1,1Ø),      1Ø CVC MIXTES UTILISES          *
*   VCCINDX=1,              *
*   NCPGRP=G2ØLX25,        *
*   TYPE=SWITCHED,         *
*   OUFINDX=1,             *
*   CALL=INOUT
*
*   LES PU ET LU SONT DEFINIES DANS VTAMLST
*
MCH21  X25.MCH ADDRESS=21,      SPEED 9,6 HALF          *
*   ANS=CONT,              *
*   IDBLKC=Ø62,           *
*   DIRECT=NO,            *
*   DBIT=NO,              *
*   LCGDEF=(Ø,2),         CANAUX LOGIQUES DE Ø A 2  *
*   FRMLGTH=131,          128+3 LONG. TRAME (MODULE 8) *
*   MWINDOW=7,           TAILLE FENETRE AU NIVEAU TRAME *
*   DSABLTO=3.Ø,         (DEF) DISABLE TIMEOUT      *
*   ENABLTO=3.Ø,         (DEF) ENABLE TIMEOUT      *
*   GATE=NO,              NO GATE                          *
*   LCNØ=NOTUSED,        (DEF) SI NETTYPE = 1          *
*   LLCLIST=(LLCØ,LLC3),  PROTOCOLE                  *
*   PAD=NO,               NOSUPPORT PAD INTEGRE          *
*   LUNAME=MC21LU,        LU MCH                      *
*   NCPGRP=G621X25M,      NOM MACRO GROUP QUI SERA GENEREE *
*   NDRETRY=1,            DEFAULT                      *
*   NPRETRY=1Ø,           DEFAULT                      *
*   PUNAME=MC21PU,
*   SPEED=96ØØ,
*   STATION=DCE,          ATTENTION HABITUDE DTE          *

```


Variable Bit Rate Voice over ATM

Although still in its infancy, Variable Bit Rate (VBR) Voice over ATM is attracting a great deal of interest. Corporates looking to maximize the efficiency and cost-effectiveness of their corporate communications are investigating the integration of both voice and data over the same ATM network. PTTs and service providers are also looking at the multi-service provisioning possibilities provided by the addition of voice circuits to ATM networks. There are a number of solutions in the market already, each with its own strengths and weaknesses, but only recently have standards-based solutions become available.

Many large enterprise users are already moving towards an ATM backbone network to fulfil their increasing data communications requirements, or to meet the increased bandwidth requirements from emerging multimedia applications. Combining voice traffic with these services makes financial sense, but how easy is it to combine the typically 'bursty' data traffic associated with LAN traffic and similar data applications with the time-sensitive bandwidth requirements of voice? Service providers are also looking to ATM backbone networks to provide new ways of delivering innovative and cost-effective services. Voice services are often regulated so that the only way of increasing service revenue in these areas is through more efficient infrastructures. ATM is seen as an increasingly viable option in these situations.

So, what are the different options for transporting voice across ATM networks, how do they compare, and how are they developing?

OPTIONS FOR TRANSPORTING VOICE IN ATM NETWORKS

Today, there are three standards-based methods that can be used for the transport of voice traffic over ATM networks. These are:

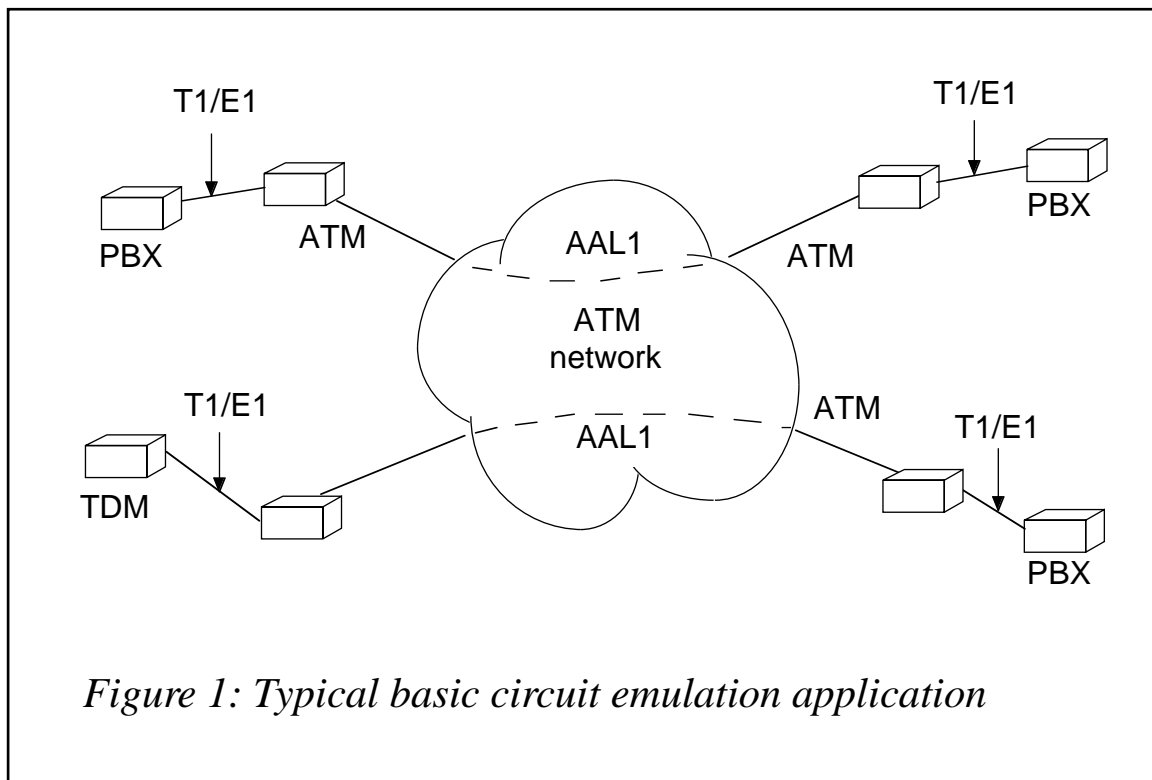
- 1 AAL1: Circuit Emulation
- 2 AAL1: Structured Circuit Emulation
- 3 AAL2: Variable Bit Rate Voice-Over-ATM.

There are, in addition, a number of proprietary systems designed to offer voice transport over ATM. In many cases, these provide reliable, high-quality services optimizing network efficiency, but may also negate interoperability – one of the key benefits of ATM. Standards-based solutions by their nature provide a more open approach and will guard against the network operator becoming locked into a single-vendor network.

CIRCUIT EMULATION

Unstructured Circuit Emulation is the ATM Forum-specified approach based on using an AAL1 CBR ATM connection to carry a full T1 or E1 circuit – or full T3 and E3 circuits – between two points in the network (Figure 1). This connection may typically be used to carry end-to-end transparent circuits between Time Division Multiplexers or Digital PBXs. Carrying voice over ATM in this manner is by far the simplest to implement and deploy, but is inefficient for two main reasons:

- The support of an emulated circuit inside an ATM network actually requires significantly more bandwidth than necessary. Because of the overhead in the ATM cell and in the AAL1 layer, the ATM connection always requires 12% more bandwidth than the circuit it is carrying. For example, a 1.536 Mbps DS1 circuit carried across an ATM backbone requires 1.73 Mbps of ATM bandwidth.
- The emulated T1/E1 circuit is typically configured as a point-to-point ATM Permanent Virtual Circuit and, as such, is always active, even if the service is not transmitting data or voice. For example, when the emulated T1/E1 carries a circuit between digital PBXs, the bandwidth is permanently allocated, even if there are no voice calls active. The bandwidth cannot simply change to meet new network or application topology requirements. This simple, but inflexible and inefficient, approach to transporting a full T1 or E1 over ATM does not enable the network to take advantage of the fact that bandwidth may become reusable when voice circuits are not being used. All voice channels are transmitted even when there are no active voice calls to the link. Other applications, such as ‘bursty’ data services, could utilize idle



voice bandwidth if it were available, but this is not the case with this approach. Therefore, not only is the bandwidth requirement excessive, but there is no opportunity to take advantage of statistical gain, which is a key factor for migrating to an ATM network.

STRUCTURED CIRCUIT EMULATION

Using Structured Circuit Emulation for Voice-Over-ATM offers some improvement over basic Circuit Emulation, because it allows AAL1 ATM connections to be established for $N \times 64$ Kbps circuits as opposed to a full T1 or E1. Figure 2 shows three digital PBXs connected in a mesh over the ATM backbone using Structured Circuit Emulation.

Individual groups, or bundles, of timeslots are mapped to ATM circuits that are sized for a portion of, rather than all, the timeslots within the E1 or T1 PBX trunks. Unlike Circuit Emulation, which is transparent to traffic, Structured Circuit Emulation must recognize each timeslot to allow routing of individual channels between PBX

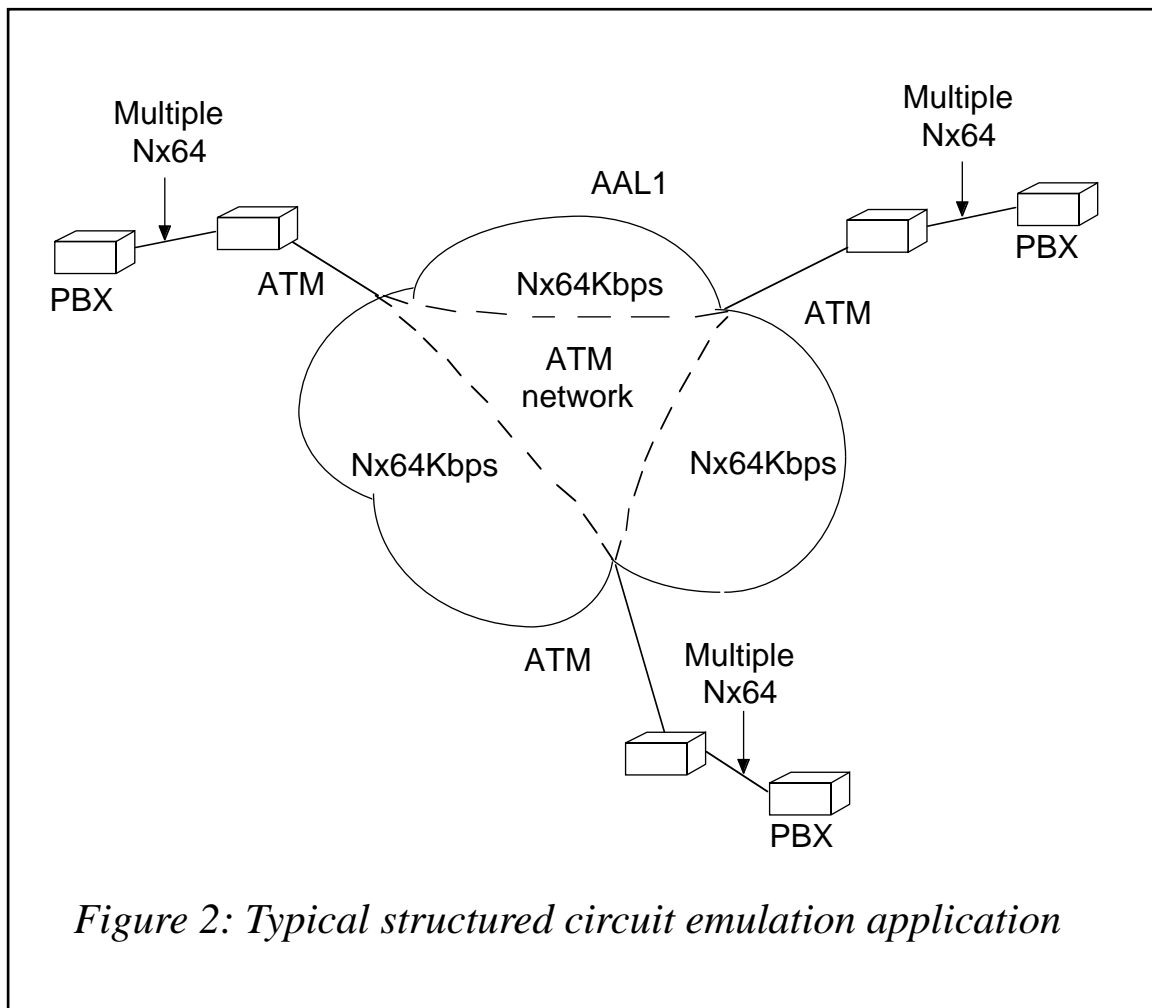


Figure 2: Typical structured circuit emulation application

trunks. Recognition of each timeslot allows the network to convey in-band Channel Associated Signalling (CAS) on a per-channel basis. However, Structured Circuit Emulation cannot accommodate out-of-band signalling protocols such as ISDN Q.931 or Q.SIG. Although the ability to establish more granular $N \times 64$ Kbps circuits between endpoints in the network seems to be an improvement over the full E1 or T1, the additional bandwidth requirements needed to establish an $N \times 64$ Kbps circuit over ATM are often overlooked. This can be disastrous because of the trade-offs involved in efficiency versus delay characteristics for voice services. Minimal delay is very important for voice networks, because delay in the network can lead to excessive echo on voice circuits – and to the need for echo cancellation. In basic Circuit Emulation, adaptation delay is minimal because the T1 or E1 bitstream is processed and segmented without regard to the framing structure within the signal. In Structured Circuit Emulation, care must be taken to operate only on those timeslots that are to be transported

across the ATM network. If an ATM cell is completely filled with information before being transmitted (this is the most efficient case), then each cell must wait for 47 sample times (47 octets of voice or data) until it is filled with data. In the most efficient case with a fill rate of 47 samples, an AAL1 circuit that transports six channels requires an ATM connection at a rate of 443 Kbps. Using Structured Circuit Emulation with partial cell fill (ie cells are partially filled with data and the remainder becomes overhead) would result in a lower delay, but requires higher bandwidth for the ATM connection. If partial fill is used with a fill rate of 24 samples, or octets, the result is an ATM bandwidth requirement of 866 Kbps for the very same six channels. A traditional T1 or E1 network requires 384 Kbps for six channels versus 443 Kbps (best case) or 866 Kbps (with 24 sample partial fill) using Structured Circuit Emulation. As such, the best case requires 15% more bandwidth, and the reduced delay/partial fill case requires more than double the bandwidth of traditional TDM!

Overall, Structured Circuit Emulation allows improved granularity for establishing fractional T1 or E1 circuits over ATM networks. However, the high penalty in bandwidth requirements and engineering challenges created by excessive delay may be overlooked. The cost of the additional bandwidth to support the service with low delays may outweigh the gains to be made by using Structured Circuit Emulation.

VARIABLE BIT RATE VOICE-OVER-ATM

To overcome the excessive bandwidth requirements needed for Structured Circuit Emulation, and provide the flexibility to allow the network operator to control delay on voice services, a new adaptation layer is required. AAL1 simply cannot be extended to meet these new ATM networking requirements. AAL2 is a new ATM Adaptation Layer, specified in ITU-T Recommendation I.363.2 (1997) with the specific mandate to provide efficient Voice-Over-ATM services.

Highlights of AAL2

AAL2 supports the following features in addition to the AAL1 adaptation protocol:

- Efficient bandwidth usage through Variable Bit Rate ATM traffic class
- ATM bandwidth reduction support for: voice compression, silence detection/suppression, idle voice channel deletion
- Multiple voice channels with varying bandwidth on a single ATM connection

AAL 2 provides bandwidth-efficient transmission of low-rate, short, and variable packets for delay-sensitive applications, and is designed to make use of the more statistically multiplexable Variable Bit Rate ATM Traffic Classes. Therefore, AAL2 is not limited to ATM connections using the CBR Traffic Class, and can enable voice applications using higher level features such as voice compression, silence detection/suppression, and idle channel removal. The structure of AAL2 allows network administrators to take traffic variations into account in the design of an ATM network and to optimize the network to match traffic conditions.

What do these enhancements mean in terms of service level and bandwidth requirements? Figure 3 illustrates the necessary bandwidth for six voice channels over ATM using each of the adaptation techniques described above.

Channels	Channel rate	Adaptation layer	Silence suppression	Bandwidth (Kbps)
6	64K PCM	AAL1 CES/CAS (24 samples per cell)	N/A	866
6	64K PCM	AAL2	No	495
6	64K PCM	AAL1 CES/CAS (47 samples per cell)	N/A	443
6	32K ADPCM	AAL2	No	283
6	64K PCM	AAL2	Yes	198
6	32K ADPCM	AAL2	Yes	113

Figure 3: Bandwidths

Standards-based, Variable Bit Rate (VBR) services defined in AAL2 therefore offer significant bandwidth savings over Structured Circuit Emulation. Using AAL2 with 32K ADPCM compression and expecting 50% silence and 20% idle channels on the PBX trunk requires a bandwidth of only 113K. This is a 74% bandwidth reduction over the best case when using Structured Circuit Emulation.

Obviously, VBR provides the most efficient solution for voice over ATM and can offer significant cost savings to users and service providers. Its combination with a standards-based service means that customers can be confident of interoperability of equipment from different vendors and connections with other networks.

Likely users include PTTs and large corporates for the reasons highlighted earlier. With voice revenues still accounting for up to 90% of service providers' revenues, there is a great deal of interest in technologies that can reduce costs and thus increase margins in this highly-regulated market. ATM services represent such a technology. Traditional carrier voice services usually travel over the TDM backbone. The trick is to maximize trunk utilization whilst minimizing cost. Migrating voice services to an ATM backbone traversing SONET or SDH infrastructure and using the VSM to deliver VBR voice traffic allows carriers to do this, whilst at the same time opening the door to opportunities for other ATM-based services to your customers. The ATM-based solution is also much less capital intensive than others which require tandem switching.

Variable Bit Rate services using standards based AAL2 adaptation therefore offer a bright future for voice over ATM. The combination of high reliability, low capital investment, and highly efficient bandwidth utilization makes the AAL2 layer a viable option for anyone wishing to use ATM to transport voice. AAL2 was developed specifically in response to demand for a more efficient method for multiplexing voice services over an ATM backbone network. Currently many leading companies and service providers are evaluating these technologies for voice over ATM with a view to roll-out services. With these pioneers effective and cost-efficient services based on voice over ATM will become a reality.

Keith Mumford
Director of Advanced Global Planning
General DataComm (UK)

© General DataComm 1998

TCP/SNA news

Cisco has added the Channel Port Adapter (CPA) product to its mainframe channel connectivity range. The CPA on a 7200 series router provides extended LAN/ WAN access to mainframes running IP or SNA applications.

It's aimed at companies that don't need the high-end attributes of Channel Interface Processor, but still want to replace IBM front-end processors, 3172 interconnect controllers, or low-performance mainframe gateways.

For further information contact:
Cisco, 5305 Gulf Drive, Suite 1, New Port Richey, FL 34652, USA.
Tel: (813) 817 0131.

* * *

Sterling Software has released Version 3.0 of Solve:Netmaster for TCP/IP, with a new Web-based interface, proactive monitoring and reporting functions, and support for Cisco's CIP.

A new alert monitor is for detecting significant IP network events such as listener availability, excessive connection idle time or byte counts, TCP/IP stack failure indicators, or SNMP information. Alerts can have automatic response, such as sending an e-mail, opening a trouble ticket, or issuing an SNMP trap.

Other features include a transaction path analyser which delivers real-time information for diagnosing recognized TN3270 response-time problems, and multi-system support, which includes a single log-on and single screen interface providing information for TCP/IP for MVS,

TCPaccess, and Cisco CIP TN3270 Server connections running over any number of systems.

For further information contact:
Sterling Software, 1800 Alexander Bell Drive, Reston, VA 22091, USA.
Tel: (703) 264 8000.
Sterling Software, 1 Longwalk Road, Stockley Park, Uxbridge, Middx, UB11 1DB, UK.
Tel: (0181) 867 8000.

* * *

OpenConnect has launched its OC:// WebConnect SNA Access Server ESCON Option, promising much improved connectivity and data transmission speed between the mainframe and TCP/IP networks.

It's a ready-to-run, rack-mount system that can be directly attached to the mainframe or indirectly attached via one or more ESCON Directors. The attachment provides the physical and electrical connection between the SCSI host and IBM ESCON optical fibre cables.

It supports high bandwidth connections and parallel data streams, and a gateway transfer option (GTO) interface allows an ESCON connection between a Unix workstation running OC/GTO and an IBM mainframe, thus allowing high-speed file transfers between the two hosts.

For further information contact:
OpenConnect Systems, 2711 LBJ Frwy, Suite 800, Dallas, TX 75234-7324, USA.
Tel: (972) 484 5200.



xephon