IBM Software Group
Enterprise Networking and Transformation Solutions (ENTS)

IBM Communication Controller for Linux (CCL) on System z

Implementing CCL

Alfred B Christensen, Raleigh,
North Carolina, USA
alfredch@us.ibm.com

© 2006 IBM Corporation

October 2006

IBM Systems

Trademarks and notices

The following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States or other countries or both:

- Advanced Peer-to-Peer Networking®
- AIX®
- alphaWorks®
- AnyNet®
- AS/400®
- BladeCenter®
- Candle®
- CICS®
- DB2 Connect
- DB2®
- DRDA®
- e-business on demand®
- e-business (logo)
- e business(logo)®
- ESCON®
- FICON®
- GDDM®
- HiperSockets
- HPR Channel Connectivity
- HyperSwap
- i5/OS (logo)
- i5/OS®
- IBM (logo)®
- IBM®
- IMS
- IP PrintWay
- IPDS
- iSeries
- LANDP®
- Language Environment®
- MQSeries®
- MVS
- NetView®
- OMEGAMON®
- Open Power
- OpenPower
- Operating System/2®
- Operating System/400®
- OS/2®
- OS/390®
- OS/400®
- Parallel Sysplex®
- PR/SM
- pSeries®
- RACF®
- Rational Suite®
- Rational®
- Redbooks
- Redbooks (logo)
- Sysplex Timer®
- System i5
- System p5
- System x
- System z
- System z9
- Tivoli (logo)®
- Tivoli®
- VTAM®
- WebSphere®
- xSeries®
- z9
- zSeries®
- z/Architecture
- z/OS®
- z/VM®
- z/VSE

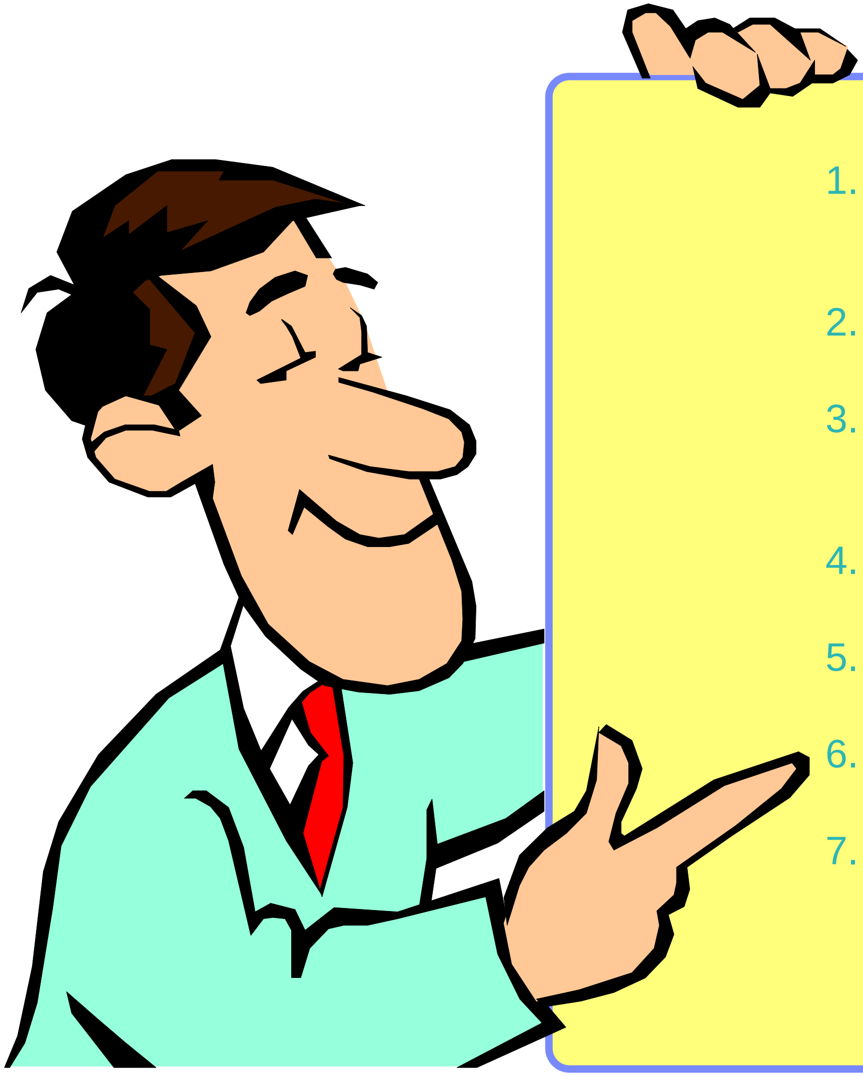
- Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.
- Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.
- Intel, Intel Inside (logos), MMX and Pentium are trademarks of Intel Corporation in the United States, other countries, or both.
- UNIX is a registered trademark of The Open Group in the United States and other countries.
- Linux is a trademark of Linus Torvalds in the United States, other countries, or both.
- Red Hat is a trademark of Red Hat, Inc.
- SUSE® LINUX Professional 9.2 from Novell®
- Other company, product, or service names may be trademarks or service marks of others.
- This information is for planning purposes only. The information herein is subject to change before the products described become generally available.
- All statements regarding IBM future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All performance data contained in this publication was obtained in the specific operating environment and under the conditions described and is presented as an illustration. Performance obtained in other operating environments may vary and customers should conduct their own testing.

Refer to www.ibm.com/legal/us for further legal information.

© 2006 IBM Corporation

Agenda



1. **CCL implementation and NCP migration planning**
2. **Sample SNA network before CCL**
3. **Migration Scenario 1: SNA LLC2 on an Ethernet LAN**
4. **Installation of CCL**
5. **Operational aspects**
6. **Migration Scenario 2: CCL DLSw**
7. **Appendixes:**
   - **A: CDLC channel connectivity to CCL**
   - **B: QDIO Layer 2 access**
   - **C: IPTG between two SNI NCPs**

© 2006 IBM Corporation

CCL implementation and NCP migration planning

© 2006 IBM Corporation

CCL project outline

- **Make a physical inventory of your current Communication Controller environment**
  - Communication Controller model, size, features, line interfaces, LAN interfaces, etc.

- **Make a logical and functional inventory**
  - NCP related functions
    - Boundary function lines, INN lines, SNI lines
    - Use of duplicate TIC MAC addressing for availability and scalability
    - XRF, NRF, NPSI
    - NTuneMON, NPA-LU
  - Functions that are not supported by CCL, such as NTO, XI, NSI, and NSF
  - Network Node Processor functions (3746-900 or 3746-950)

Refer to *IBM Communication Controller Migration Guide*, SG24-6298 appendix A and B for inventory worksheets.

- **Reconcile and optimize**
  - Identify hardware and software components that are no longer used
  - Remove hardware components that are no longer used (can reduce both maintenance cost and NCP Tier pricing)
  - Clean up NCP definitions accordingly

- **Controller consolidation and migration strategy planning**
  - Define high availability strategy - levels of redundancy and fail-over capabilities
  - Identify workloads that could be moved off SNA wide area networking via SNA/IP integration technologies
  - Which NCPs to move to CCL and in which order
  - Which NCPs to consolidate when moving to CCL NCPs and in which order
  - Which NNP or MAE functions to migrate to CSL (Communications Server for Linux on zSeries)
  - Which remaining functions to consolidate into fewer Communication Controller footprints

© 2006 IBM Corporation

CCL project outline (continued)

CCL provides improved options for redundancy design: you don't need to buy an extra IBM 3745 to deploy a stand-by NCP.

- **Overall CCL environment design for high availability**
  - Number of Linux images and number of CCL NCPs required to support strategy
  - Linux and CCL NCP deployment from a data center and CEC perspective - LPARs or z/VM, which CCL NCP goes where
  - Linux and CCL availability design - management and recovery procedures and tools
  - Network availability and load balancing through duplicate MAC support for token-ring or Ethernet LAN connectivity to CCL NCPs (Ethernet LAN requires additional design of DLSw components)
  - Wide area network connectivity through aggregation layer routers - how many, what type of WAN interfaces, how to provide redundancy for WAN termination if required
    - Consider optimization opportunities by terminating WAN lines in remote locations that today are already connected through an IP backbone to the data center - using DLSw technology over the IP backbone
  - LAN infrastructure changes - token-ring and/or Ethernet, how to interconnect and/or migrate
  - System z hardware requirements: IFLs, memory, DASD, OSA ports
  - Physical LAN cabling between OSA ports, switches, and aggregation layer routers
  - Define any changes business partners may have to implement (depends on migration strategy)
  - Define any changes to peripheral SNA link stations (depends on migration strategy)

© 2006 IBM Corporation

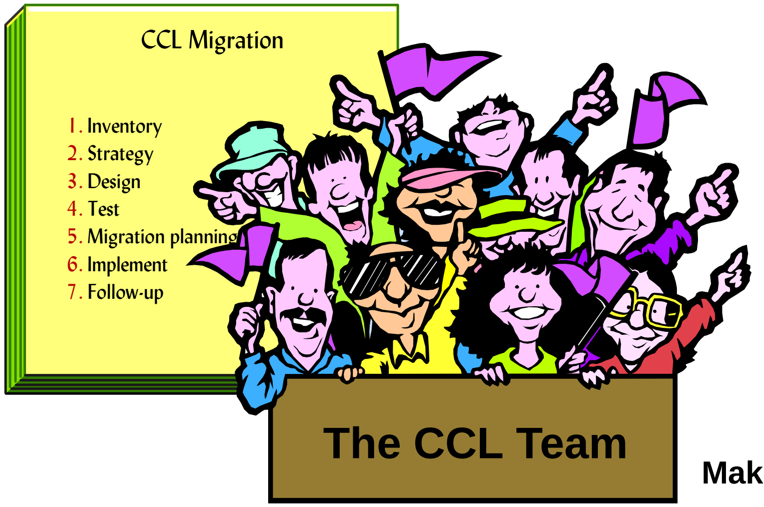
CCL project outline (continued)

- **Establish a test environment**
  - One or two Linux on System z with CCL and test NCPs
  - Experiment, test, learn - make sure to include recovery scenarios in the test activities
- **CCL detailed migration planning per CCL NCP**
  - NCP migration strategy
    - Deactivate old NCP subarea and activate new NCP with same subarea in CCL (this requires the least amount of NCP changes and allows reuse of existing TIC MAC addresses by OSA ports)
    - Keep old NCP subarea active, activate new NCP with new subarea in CCL, and migrate resources over time to new NCP (requires changes to SNA subarea path definitions and may prevent you from reusing existing TIC MAC addresses in the new environment)
  - WAN connectivity migration strategy
    - If the existing IBM 3745/46 has TIC interfaces, a migration of WAN lines to aggregation layer routers could be considered before moving the NCP to CCL (simplifies the move to CCL)
    - Otherwise the move of WAN lines to aggregation layer routers cannot start until the NCP has been activated in the CCL
  - Fall back planning for each planned step

© 2006 IBM Corporation

CCL project outline (continued)

- **CCL implementation**
  - Establish planned infrastructure (Linux images, CCLs, OSA ports, cabling, switches, etc.)
  - Migrate one NCP at a time according to detailed plan
- **Consolidation of remaining IBM 3745/46 resources**
  - Functions not supported by CCL should be consolidated into fewer and smaller IBM 3745/46s
  - Clean up NCPs and associated licenses for old environment



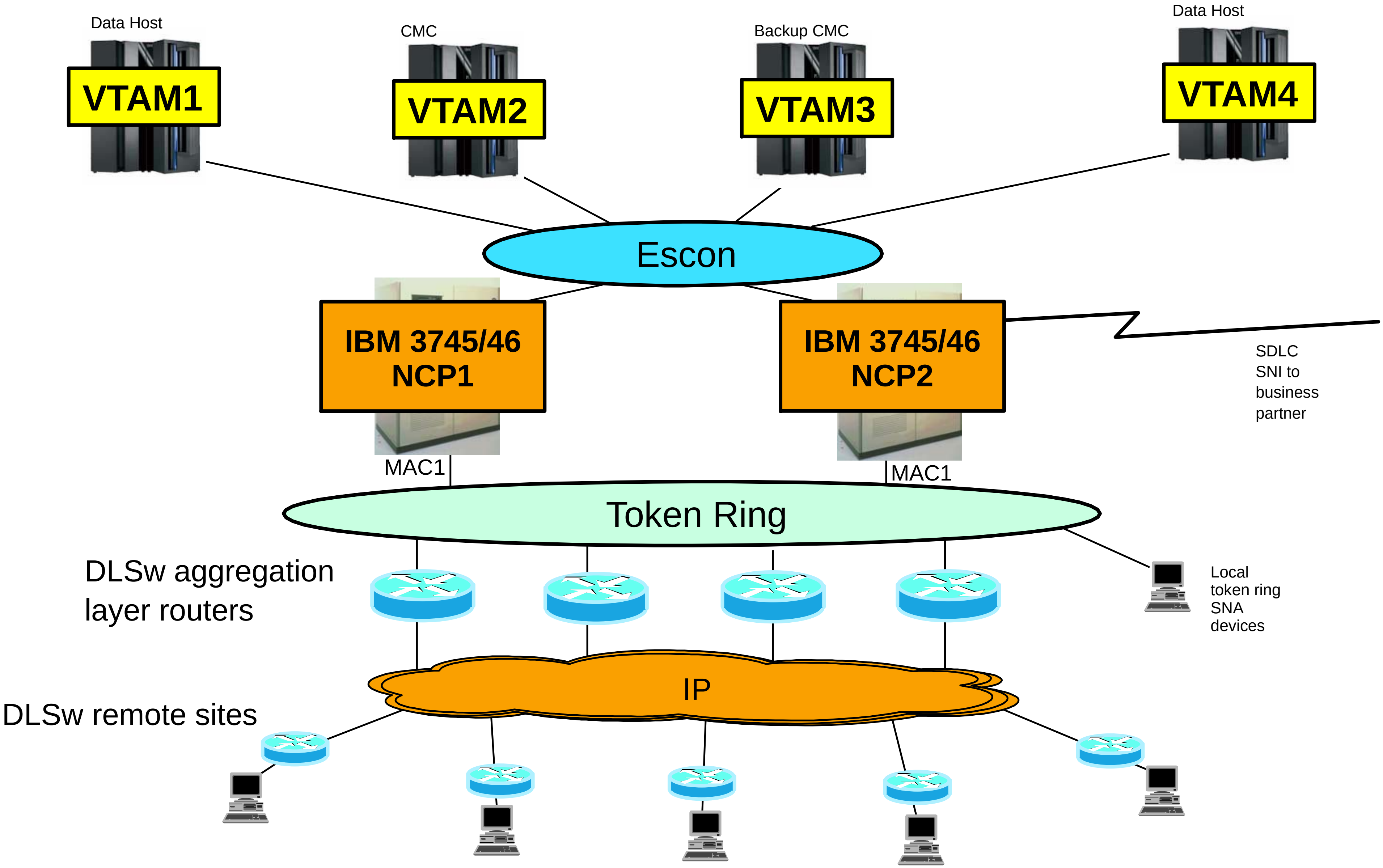
Make sure you have a plan before you start!

© 2006 IBM Corporation

Sample SNA network before CCL

© 2006 IBM Corporation

Sample SNA network (diagram)



Data Host

VTAM1

CMC

VTAM2

Backup CMC

VTAM3

Data Host

VTAM4

Escon

IBM 3745/46
NCP1

IBM 3745/46
NCP2

SDLC
SNI to
business
partner

MAC1

MAC1

Token Ring

DLSw aggregation
layer routers

Local
token ring
SNA
devices

IP

DLSw remote sites

© 2006 IBM Corporation

Sample SNA network (description)

- **Multiple VTAMs,each with an ESCON connection to the NCPs**
  - One CMC, one backup CMC, and two data hosts

- **Two IBM 3745/46s**
  - For redundancy

- **Some SNI connections to business partners**
  - Perhaps over SDLC leased lines

- **SNA BNN devices connect to NCPs over token ring**
  - Duplicate TIC configurationb: both NCPs use MAC1 as their source MAC
  - All BNN devices configured to connect to MAC1

- **Some SNA BNN devices are local**
  - Attached to the data center token ring infrastructure
  - IBM 3174, CS/AIX, CS/Linux, CS/Windows, etc.

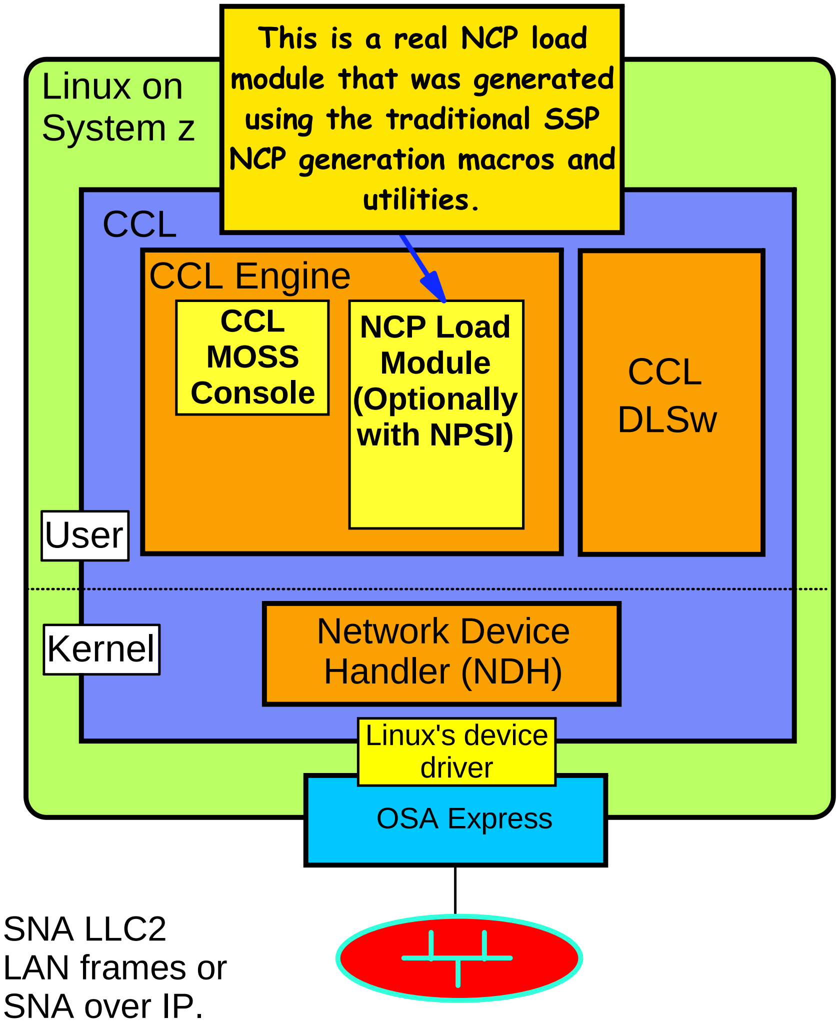
- **Most are remote**
  - DLSw router at the remote site terminates the SNA LLC2 from the BNN device
  - SNA data is transported over IP to DLSw peer routers in the data center
    - Non-SNA IP traffic may also flow over that IP network
    - Remote DLSw router may load balance over a number of DLSw peers
  - Aggregation layer DLSw routers in the data center put SNA data onto the token ring
    - LLC2 connection between the IBM 3745 NCP and the aggregation layer router is fast and highly reliable
    - DLSw router can pick either of the two NCPs with the duplicate MAC

© 2006 IBM Corporation

Migration Scenario 1:
SNA LLC2 on an Ethernet LAN

© 2006 IBM Corporation

CCL overall structure and components - recap



This is a real NCP load module that was generated using the traditional SSP NCP generation macros and utilities.

Linux on System z

CCL

CCL Engine

CCL MOSS Console

NCP Load Module (Optionally with NPSI)

CCL DLSw

User

Kernel

Network Device Handler (NDH)

Linux's device driver

OSA Express

SNA LLC2
LAN frames or
SNA over IP.

Note: You will continue to use ACF/SSP to generate, load, and dump an NCP load module.

- **CCL** supports an **NCP** performing Boundary Functions, INN, and SNI link connectivity, as well as **NPSI** .
- **CCL** consists of both user-space and kernel-space functions:
  - **CCL engine** emulates an IBM 3745-31A with 16 MB memory supporting an NCP load module and a MOSS console interface.
    - The **MOSS console** is accessed through a standard Web browser.
  - **Network Device Handler (NDH)** is a kernel extension that acts as the interface between a real network interface (such as an OSA port) and the CCL adapter emulation support.
    - The only supported LAN interface from an NCP perspective is a token-ring.
      - CCL V1R1 and V1R2: TIC2
      - CCL V1.2.1 TIC2 and TIC3
    - The actual LAN to which the OSA port is connected may be either token-ring or IEEE802.3 Ethernet (NDH will transform between the frame formats).
    - Serial lines are terminated in an aggregation layer router that connects to the CCL NCP via:
      - SNA LLC2 over a LAN
      - DLSw over an IP network
      - XOT over an IP network for non-SNA X.25 access to NPSI
    - CDLC connectivity via OSA for NCP (OSN)
  - **CCL DLSw** is a separate user-space application
    - communicates with CCL NCP through NDH using LLC2 flows
    - communicates with other DLSw peers through the Linux TCP sockets layer, using DLSw protocols

© 2006 IBM Corporation

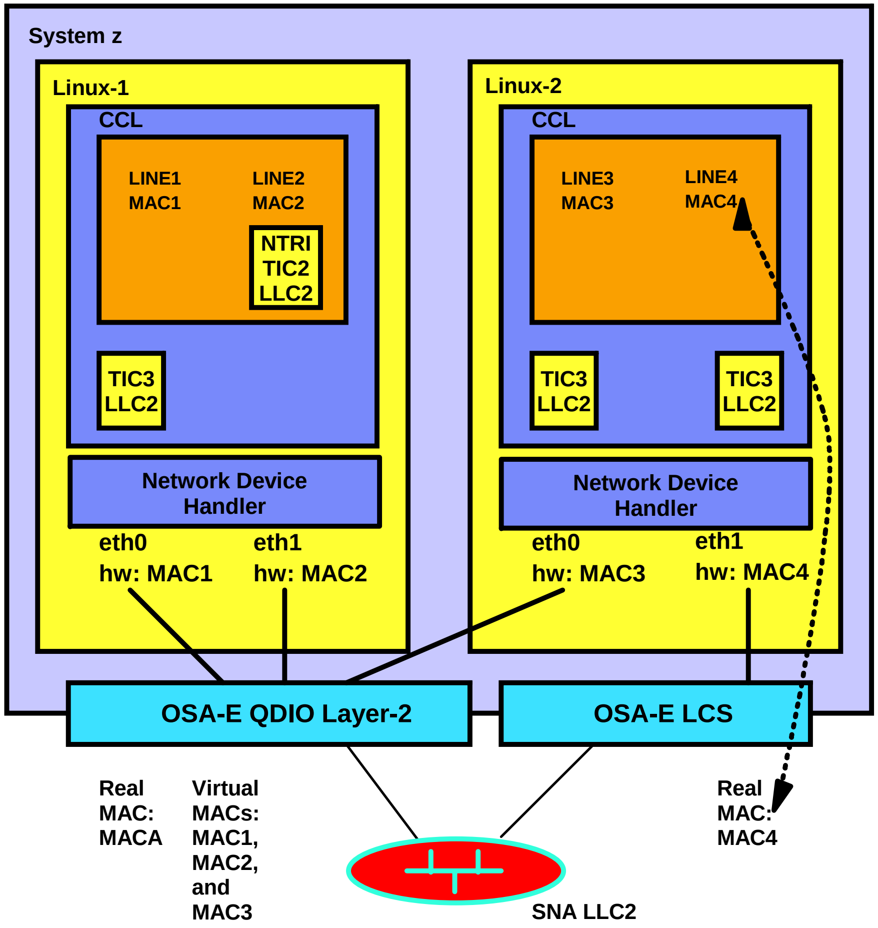
CCL SNA LLC2 LAN connectivity - recap

- **NCP definitions for two types of LAN adapters:**
  - **TIC2** - NCP Token-ring Interface (NTRI) support
    - NCP instructions used for LLC2 logic (executed by the IBM 3745 emulator)
  - **TIC3** - IBM 3746 Token-ring Processor (TRP)
    - Native System z instructions used for LLC2 logic
    - Also referred to as "native LAN" support

  TIC3 uses much less CPU than TIC2 !!!

- **Linux has two different LAN device drivers for LLC2:**
  - **LAN Channel Station (LCS)**
    - Copper cabling, token ring or Ethernet
    - NCP physical LINE local MAC address specification must match OSA port's configured real MAC address (OSA/SF)
    - One NCP physical line per OSA LCS port
    - Limited sharing via configured SAP numbers
  - **Queued Direct I/O (QDIO) operating in layer 2 mode**
    - Copper and fiber cabling, Ethernet only
    - NCP physical LINE local MAC address specification must match Linux interface hardware address (virtual MAC), but not OSA port's real MAC address
      - Up to 2048 virtual MAC addresses per OSA port
      - Eases migration for NCPs with many MAC addresses
    - Requires a Linux 2.6 kernel
    - Works for Linux LPARs or z/VM guests
    - If used by Linux as z/VM guests it can be used in combination with z/VM's virtual switch

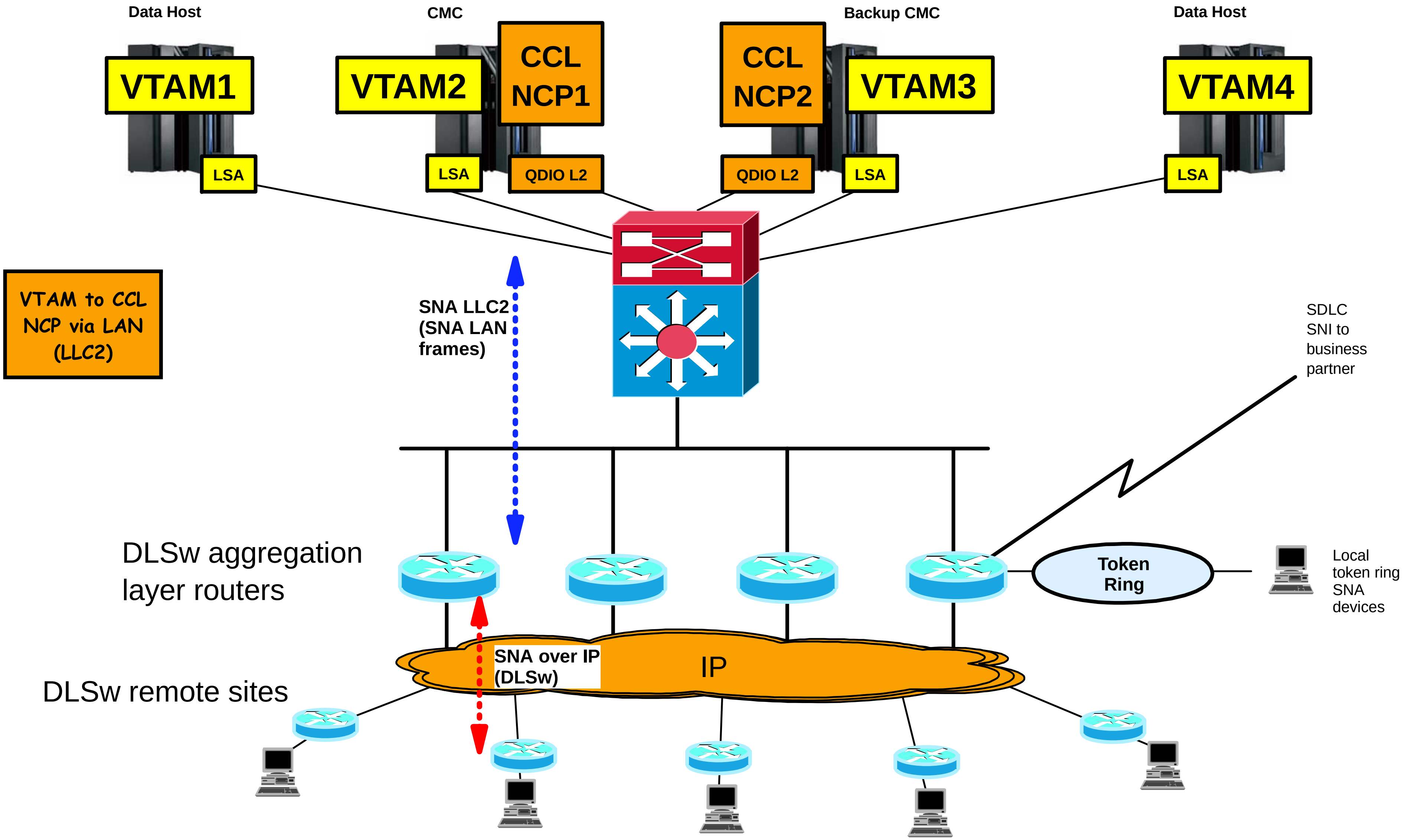
- **NDH ties NCP LAN definitions to Linux devices**
  - CCL registers NCPs token ring MAC address
  - NDH finds the **eth** or **tr** device with a matching MAC
    - does TR-to-ETH frame conversion if necessary



Layer-2 mode is supported by Fast Ethernet, 1000BASE-T Ethernet, Gigabit Ethernet, and 10 Gigabit Ethernet features on OSA-Express and OSA-Express2 on z890, z990, and System z9

© 2006 IBM Corporation

Migration to CCL using LLC2 over Ethernet (diagram)


Data Host
CMC
Backup CMC
Data Host
VTAM1
VTAM2
CCL NCP1
CCL NCP2
VTAM3
VTAM4
LSA
LSA
QDIO L2
QDIO L2
LSA
LSA
VTAM to CCL NCP via LAN (LLC2)
SNA LLC2 (SNA LAN frames)
SDLC SNI to business partner
DLSw aggregation layer routers
Token Ring
Local token ring SNA devices
SNA over IP (DLSw)
IP
DLSw remote sites

Migration to CCL using LLC2 over Ethernet (description)

- **Install two CCLs, one in each of two CECs (for redundancy)**
  - Same logical subarea configuration (four VTAMs, two NCPs), but running NCP inside CCL instead of inside IBM 3745
- **All SNA traffic now flows over a high speed ethernet core**
  - Removed the IBM 3745/46s, the ESCON that was used by the them, and as much of the data center token ring infrastructure as possible
- **Ethernet LAN (gigabit) connectivity between the VTAMs and NCPs, instead of ESCON**
  - LSA (copper 1000baseT) on the VTAM side, XCA major node definitions
  - QDIO Layer2 (copper or fiber) on the CCL side, NCP token ring physical/logicals
- **Ethernet LAN (GigE or 1000Mb) connectivity between the NCPs and the aggregation layer routers, instead of 16Mb token ring**
  - NCP definitions for token ring devices stay the same as before
    - Use same MAC address (MAC1) on token ring physical LINE
    - QDIO Layer2 device defined with canonical version of NCP's non-canonical MAC
  - Two separate VLANs are required
    - One for each instance of the duplicate MAC
- **SDLC (for example for SNI) connections migrated to DLSw**
  - NCP SDLC Line definitions changed to token ring
  - Serial ports and SDLC definitions added to a DLSw router
    - SDLC partner does not need to change
- **Local SNA token ring devices bridged (SR/TB) onto ethernet**
  - No changes to token ring attached devices, local or remote

© 2006 IBM Corporation

MAC address formats - token-ring (non-canonical) and Ethernet (canonical)

- **The NCP sees all LAN interfaces as being token-ring**
  - A token-ring MAC address is in the non-canonical form and this form is what must be coded in the NCP generation deck.
  - The NCP requires locally administered MAC addresses
    - MAC addresses starting with B'01xx xxxx'
  - If the OSA port is token-ring, then the MAC address in the NCP and in OSA/SF for the OSA port match
  - If the OSA port is Ethernet, then the MAC address in the NCP must be the non-canonical form of the Ethernet canonical MAC address as specified in OSA/SF
  - Canonical is little-endian, while non-canonical is big-endian
  - A utility is provided with CCL to assist in the conversion
    - Canonical
    - Canonical.cmd (REXX version)

Canonical address (Ethernet)	**08**	**00**	**3f**	**e1**	**4d**	**a8**
Binary	**00001000**	**00000000**	**00111111**	**11100001**	**01001101**	**10101000**
Reverse bits in each byte	**00010000**	**00000000**	**11111100**	**10000111**	**10110010**	**00010101**
Non-canonical version (token-ring)	**10**	**00**	**fc**	**87**	**b2**	**15**

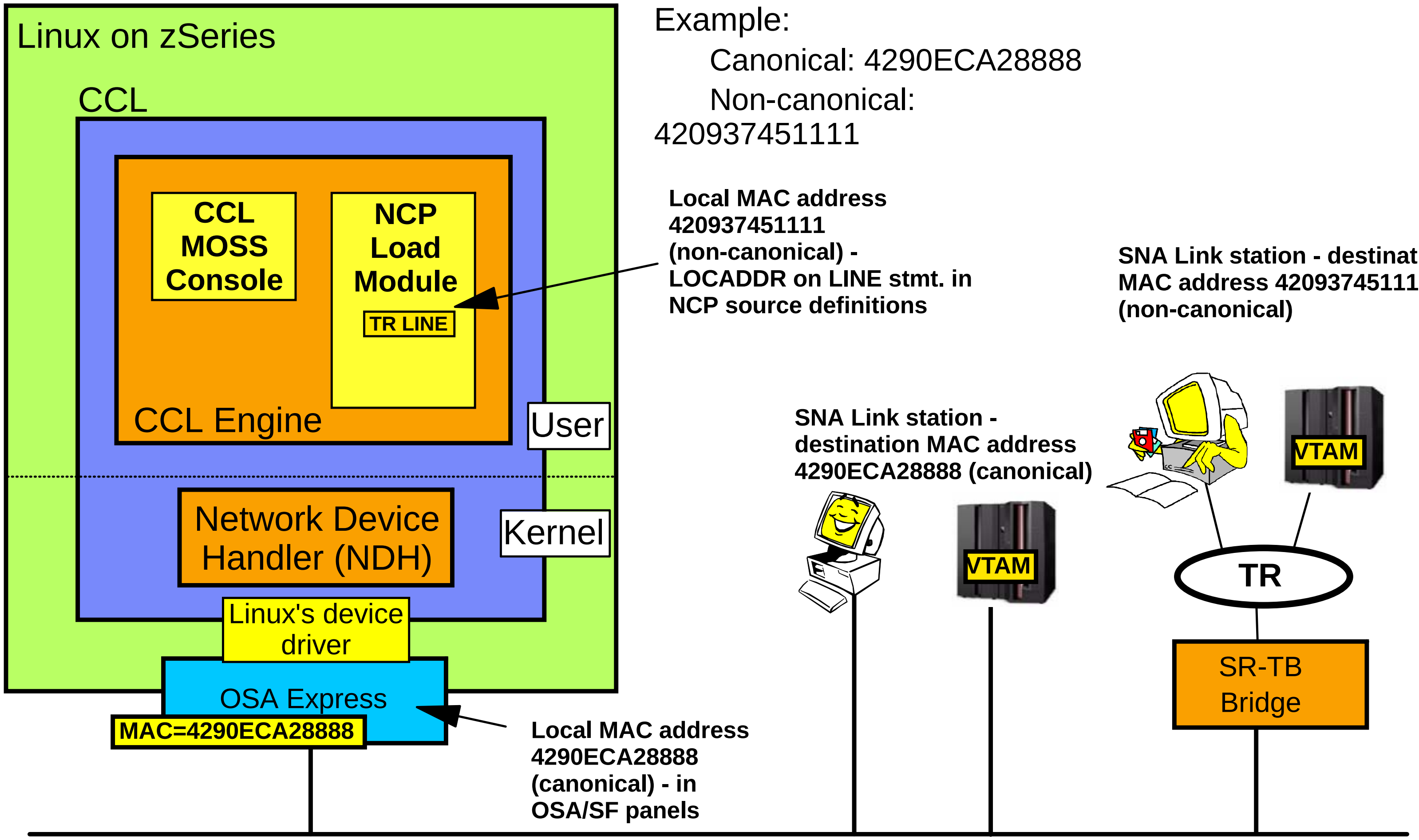
© 2006 IBM Corporation

Ethernet considerations - canonical or non-canonical MAC address

Example:

Canonical: 4290ECA28888

Non-canonical: 420937451111



Local MAC address 420937451111 (non-canonical) - LOCADDR on LINE stmt. in NCP source definitions

SNA Link station - destination MAC address 420937451111 (non-canonical)

SNA Link station - destination MAC address 4290ECA28888 (canonical)

Local MAC address 4290ECA28888 (canonical) - in OSA/SF panels

```
linux127:/opt/ibm/Communication_Controller_for_Linux/samples/mac_addr_converters # ./canonical 4290ECA28888
420937451111
```

© 2006 IBM Corporation

Sample NCP definitions for Ethernet connectivity

```
C72PTRG1 GROUP ECLTYPE=(PHY,ANY),ADAPTER=TIC2,ANS=CONT,MAXTSL=16732,   X
               RCVBUFC=32000,USSTAB=AUSSTAB,ISTATUS=ACTIVE,XID=NO,     X
               RETRIES=(20,5,5)
*-----------------------------------------------------------------
* Physical Ethernet LINE - BNN and INN
*-----------------------------------------------------------------
C72TR88  LINE  ADDRESS=(1088,FULL),TRSPEED=16,PORTADD=88,              X
               LOCADD=420937451111
C72PU88A PU
**********************************************************************
*     NTRI BNN LOGICAL LINES FOR TOKEN RING PORT 1088                *
**********************************************************************
C72BNNG1 GROUP ECLTYPE=LOGICAL,ANS=CONTINUE,AUTOGEN=200,CALL=INOUT,    X
               ISTATUS=ACTIVE,PHYSRSC=NONE,                            X
               USSTAB=AUSSTAB,RETRIES=(10,10,10,20),XMITDLY=NONE,      X
               MODETAB=AMODETAB
**********************************************************************
*     NTRI INN LOGICAL LINES FOR TOKEN RING PORT 1088                *
**********************************************************************
C72INNG1 GROUP ECLTYPE=(LOGICAL,SUBAREA),ANS=CONT,PHYSRSC=C72PU88A,    X
               LOCALTO=13.5,REMOTTO=18.2,T2TIMER=(0.2,0.2,3),          X
               ISTATUS=ACTIVE,SDLCST=(C72PRI,C72SEC),MONLINK=CONT
*---------------------------------------------------------------------
* Linkstation to VTAM NETC.C02N
*---------------------------------------------------------------------
C72LG2A  LINE  TGN=1,TGCONF=SINGLE
C72PG2A  PU    ADDR=18420937450220,SSAP=(08,H)
```

NCP name in this example is C72DUPE

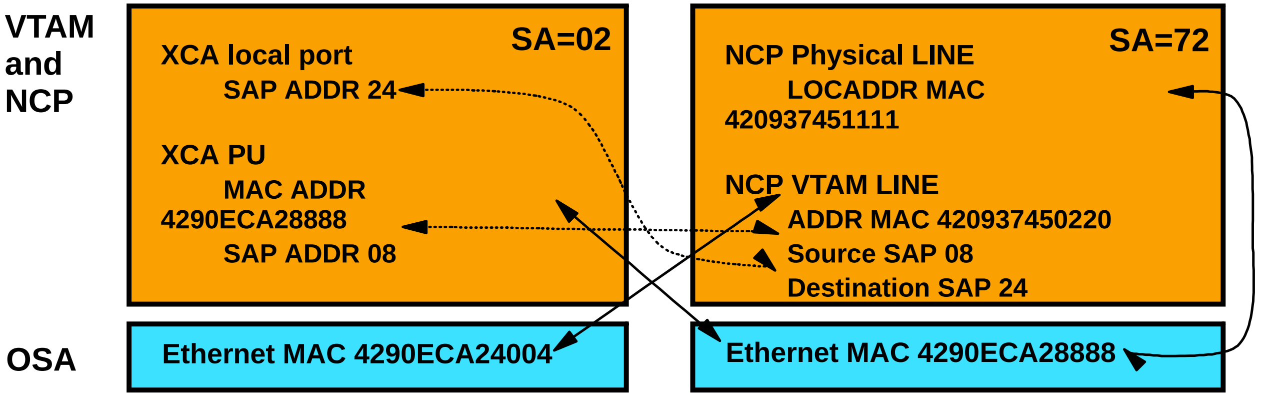
- **This is the NCP MAC address in non-canonical form**
  - Canonical: 4290ECA28888
- **The first two digits is VTAM's SAP number in hexadecimal**
  - X'18' = SAP 24
- **The remaining digits is the non-canonical form of VTAM's MAC address**
  - Canonical: 4290ECA24004
- **For the subarea link to VTAM, we will use local SAP 08**

© 2006 IBM Corporation

Sample VTAM XCA major node for Ethernet connectivity

```
C02XCADT VBUILD  TYPE=XCA
*
C02PRTA  PORT  MEDIUM=CSMACD,ADAPNO=0,SAPADDR=24,CUADDR=2EEA,          X
               TIMER=100
C02GRPA  GROUP DIAL=NO,ISTATUS=ACTIVE
*
C02ETHLA LINE  USER=SNA,ISTATUS=ACTIVE
C02ETHPA PU    MACADDR=4290ECA28888,PUTYPE=5,SUBAREA=72,TGN=1,         *
               SAPADDR=08,ALLOWACT=YES
```

- **This is the NCP's MAC address in canonical form**
  - Non-canonical: 420937451111



From a hardware point of view:
Ethernet MAC 4290ECA24004
SAP 24
From an NCP point of view:
TR MAC 420937450220
SAP 24

From a hardware point of view:
Ethernet MAC 4290ECA28888
SAP 08
From an NCP point of view:
TR MAC 420937451111
SAP 08

© 2006 IBM Corporation

How to load an NCP load module into CCL over a LAN (OSA LSA port from VTAM)

- **The very first NCP load module must be manually transferred to Linux and loaded into the CCL via a shell command interface**
  - ./cclengine -m<NCP load mod name> -p<MOSS port> <ccl engine name>
    - This process can be automated to be performed during IPL of Linux
- **VTAM's XCA definitions need to be activated:**
  - **VARY net,ACT,ID=XCA_pu**
- **The NCP can then be activated from VTAM using a normal V NET,ACT,ID=<NCP name> command**
  - **VARY net,ACT,ID=NCPname**
- **The LOADFROM=HOST option is not supported by CCL over a LAN, but is by CCL V1R2 when connecting to a CCL NCP over an OSA for NCP (OSN) CHPID**
- **The LOADFROM=EXTERNAL option is not supported for a CCL that is directly adjacent to VTAM**
- **NCP load modules on the MOSS disk can from then on be refreshed using the existing VTAM MODIFY LOAD commands to save a new NCP load module to the MOSS disk (a Linux file), and to schedule a timed IPL of the newly transferred NCP load module:**
  - MODIFY net,LOAD,ID=NCPname,ACTION=ADD/REPLACE,LOADMOD=loadmod,IPLTIME=
  - MODIFY net,LOAD,ID=NCPname,ACTION=SETTIME,LOADMOD=loadmod,IPLTIME=

© 2006 IBM Corporation

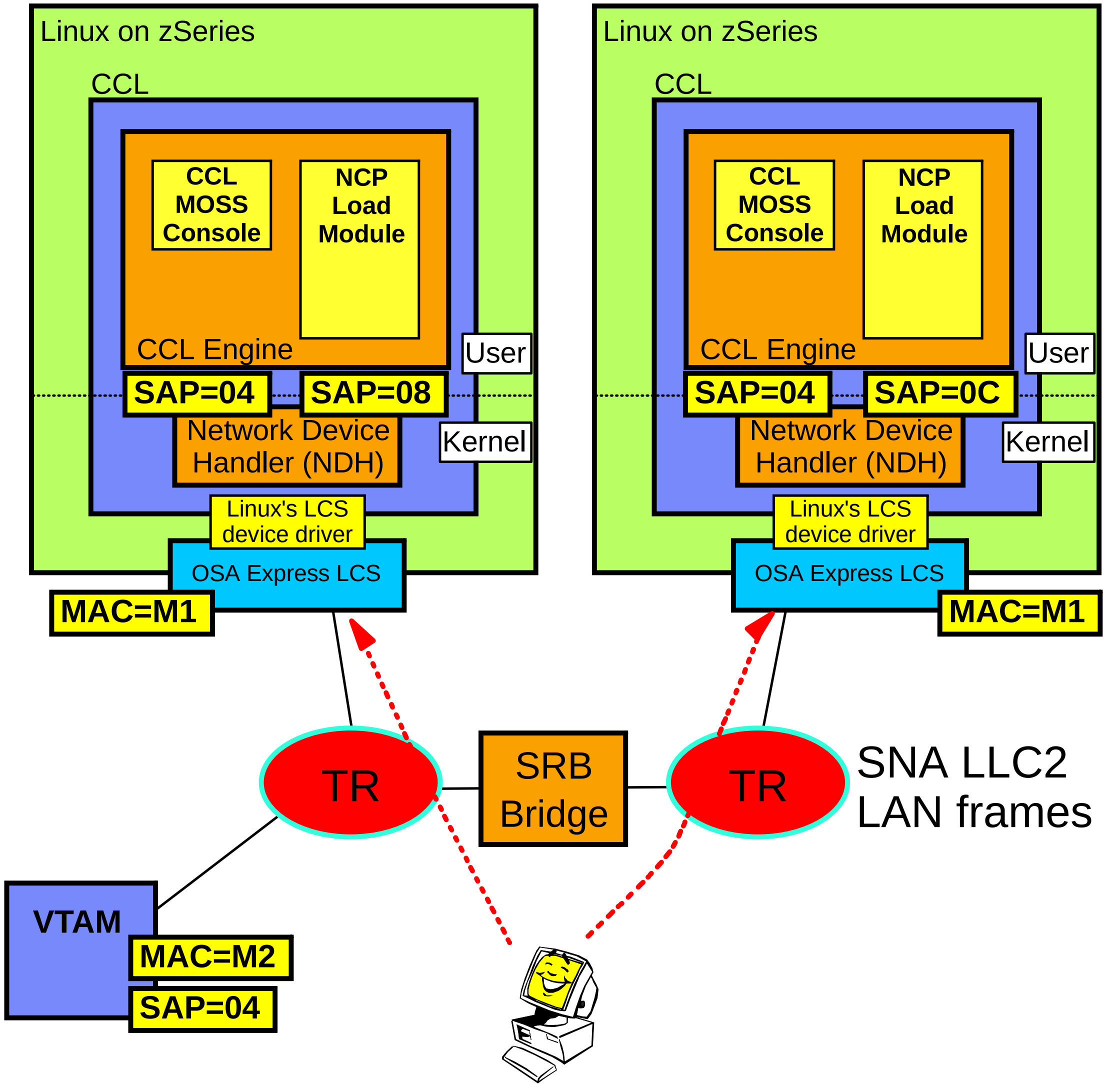
Sample VTAM activation and display commands

```
v net,act,id=c02xcadt,all
IST093I C02XCADT ACTIVE
IST464I LINK STATION C02ETHPA HAS CONTACTED C72DUPE SA 72
IST093I C02ETHPA ACTIVE


v net,act,id=c72dupe,all
IST093I C72DUPE ACTIVE
IST093I C72PU88A ACTIVE
IST464I LINK STATION C72PG2A HAS CONTACTED C02NPU SA 2
IST093I C72PG2A ACTIVE


D NET,ID=C02ETHLA,E
IST097I DISPLAY ACCEPTED
IST075I NAME = C02ETHLA, TYPE = LINE 592
IST486I STATUS= ACTIV----E, DESIRED STATE= ACTIV
IST087I TYPE = LEASED             , CONTROL = SDLC, HPDT = *NA*
IST134I GROUP = C02GRPA, MAJOR NODE = C02XCADT
IST1500I STATE TRACE = OFF
IST1656I VTAMTOPO = REPORT, NODE REPORTED - YES
IST1657I MAJOR NODE VTAMTOPO = REPORT
IST396I LNKSTA   STATUS     CTG GTG  ADJNODE ADJSA    NETID    ADJLS
IST397I C02ETHPA ACTIV--W-E   1   1  C72DUPE    72     NETC
IST314I END
```

The 'W' flag indicates that ALLOWACT=YES was coded on this PU in VTAM's XCA major node.

© 2006 IBM Corporation

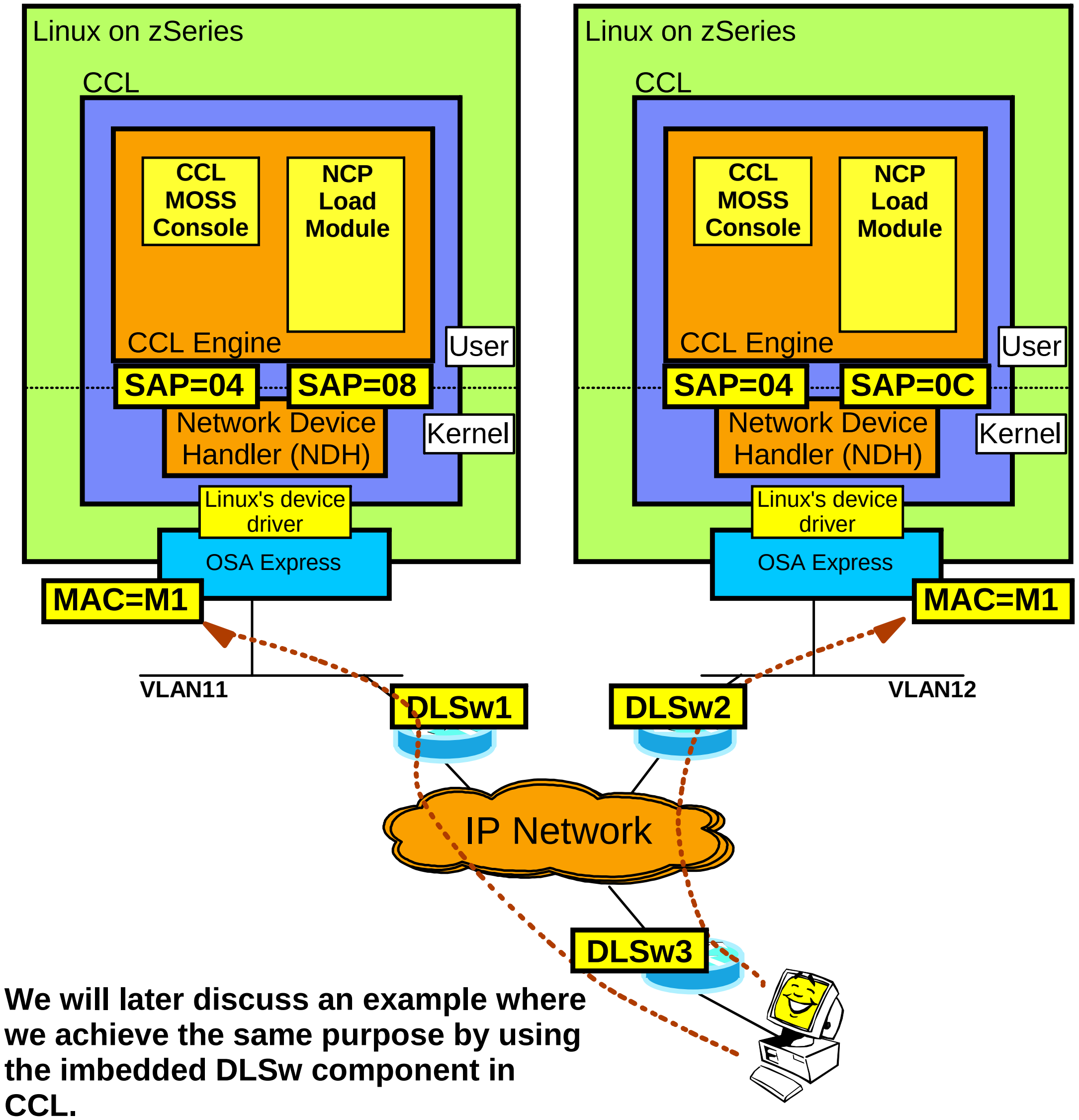
Duplicate token-ring MAC addresses in two CCL NCPs for NCP availability and load balancing



- **Load balancing of remote SNA link station access when both CCL instances are up and running**
  - Each NCP needs unique SAP for VTAM links - VTAM needs to know them as separate NCPs (SAP 08 and 0C in this setup)
  - The two NCPs need the same SAP for downstream links (SAP 04 in this setup)
- **Availability**
  - If an LCS port, a Linux image, an NCP, or a CCL engine goes down, remote link stations can recover over the other CCL instance
    - As usual in an SNA network, such a switch is disruptive to SNA sessions (subarea and APPN)
    - SNA sessions over HPR will survive such a switch
  - Traditional availability aspects would direct one towards two Linux images on two different zSeries CECs in two different data centers for maximum availability

© 2006 IBM Corporation

Duplicate Ethernet MAC addresses in two CCL NCPs for NCP availability and load balancing - external DLSw routers



We will later discuss an example where we achieve the same purpose by using the imbedded DLSw component in CCL.

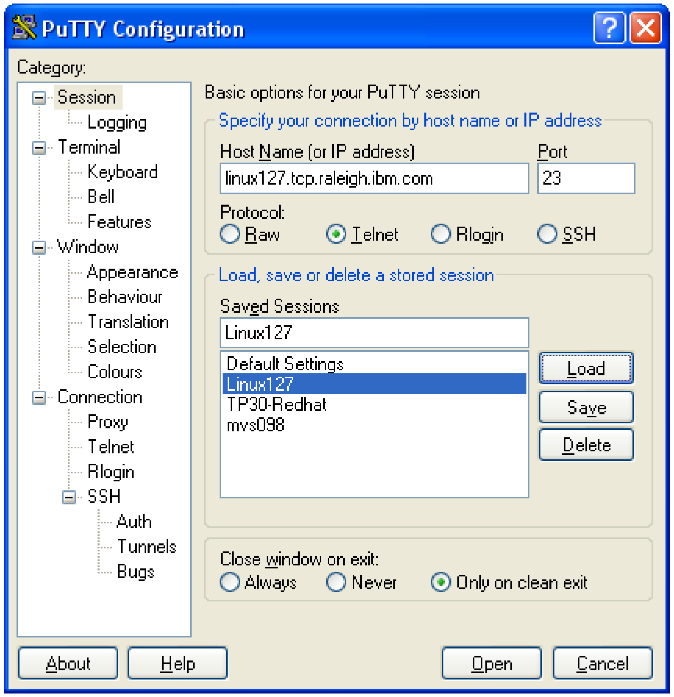
- **Ethernet segments with duplicate MAC addresses cannot be bridged together**
  - Ethernet bridging technology does not support two devices on one VLAN with the same MAC address
- **Elements of the DLSw technology can instead be used to connect the Ethernet segments with duplicate MAC addresses together and with other LAN segments**
  - MAC=M1 appears on both VLAN11 and VLAN12, so they must be kept separate
  - DLSw1 and DLSw2 both report reachability to MAC address M1 and SAP 04
  - DLSw3 can be configured to use load balancing (for example, round-robin) towards the two DLSw peers for all connections from downstream SNA devices to destination MAC address M1
    - Allows an even spread of SNA link station connections when both are available

© 2006 IBM Corporation

Installation of CCL

© 2006 IBM Corporation

A useful Windows tool: PuTTY - a telnet and SSH client



Basic options for your PuTTY session

Specify your connection by host name or IP address

Host Name (or IP address): linux127.tcp.raleigh.ibm.com

Port: 23

Protocol: ○ Raw ● Telnet ○ Rlogin ○ SSH

Load, save or delete a stored session

Saved Sessions: Linux127

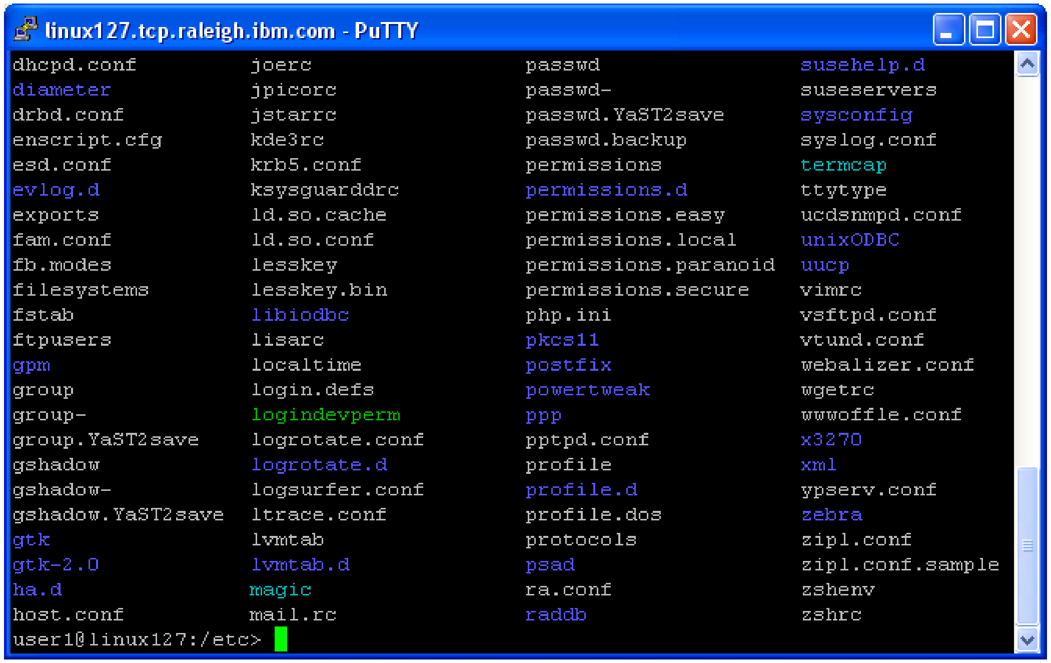
- Default Settings
- Linux127
- TP30-Redhat
- mvs098

Close window on exit: ○ Always ○ Never ● Only on clean exit

http://www.chiark.greenend.org.uk/~sgtatham/putty/

PuTTY can be used both as a telnet client and as an SSH client into your Linux system.

- **You may also find an X-Windows server solution for Windows to be useful when working with Linux on System z.**
- **The main SUSE Linux setup and configuration tool (YAST) is best accessed via X-Windows.**
- **X-Windows solutions for Windows come for free, for a small fee, and expensive.**
  - The small fee (USD 25 or so) solutions usually work fine.
- **No endorsement - just an example:**
  - MI/X 4.2
  - www.microimage.com



```
dhcpd.conf         joerc                 passwd                susehelp.d
diameter           jpicorc               passwd-               suseservers
drbd.conf          jstarrc               passwd.YaST2save      sysconfig
enscript.cfg       kde3rc                passwd.backup         syslog.conf
esd.conf           krb5.conf             permissions           termcap
evlog.d            ksysguarddrc          permissions.d         ttytype
exports            ld.so.cache           permissions.easy      ucdsnmpd.conf
fam.conf           ld.so.conf            permissions.local     unixODBC
fb.modes           lesskey               permissions.paranoid  uucp
filesystems        lesskey.bin           permissions.secure    vimrc
fstab              libiodbc              php.ini               vsftpd.conf
ftpusers           lisarc                pkcs11                vtund.conf
gpm                localtime             postfix               webalizer.conf
group              login.defs            powertweak            wgetrc
group-             logindevperm          ppp                   wwwoffle.conf
group.YaST2save    logrotate.conf        pptpd.conf            x3270
gshadow            logrotate.d           profile               xml
gshadow-           logsurfer.conf        profile.d             ypserv.conf
gshadow.YaST2save  ltrace.conf           profile.dos           zebra
gtk                lvmtab                protocols             zipl.conf
gtk-2.0            lvmtab.d              psad                  zipl.conf.sample
ha.d               magic                 ra.conf               zshenv
host.conf          mail.rc               raddb                 zshrc
user1@linux127:/etc> 
```

© 2006 IBM Corporation

Installation steps

1. **Transfer the install image to the Linux system where it is to be installed**
2. **Unpack the install image using the tar command**
3. **Run the install program that starts the InstallShield**
4. **Answer the questions from the InstallShield and let InstallShield perform the install**
5. **Install the NDH rpm package**
6. **Build and load the NDH kernel modules**
7. **Generate and transfer an NCP load module to Linux**
8. **Start the CCL engine**
9. **Automate the startup process so it is done during Linux boot**

© 2006 IBM Corporation

Starting the install process

- **Copy the following file to a temporary directory on the machine where CCL will be installed:**
  - cclv1.2.1.tar.gz
- **Use the following command to untar the file:**
  - tar -zxvf cclv1.2.1.tar.gz
- **The following files will be extracted:**
  - cclv1.2.1/README
  - cclv1.2.1/setuplinux390.bin
  - cclv1.2.1/ndh/ndh-1.2.1-x.s390.rpm
    - where x denotes the rpm version.
- **Change to the cclv1.2.1 directory and type the following to run the CCL installation program:**
  - ./setuplinux390.bin &

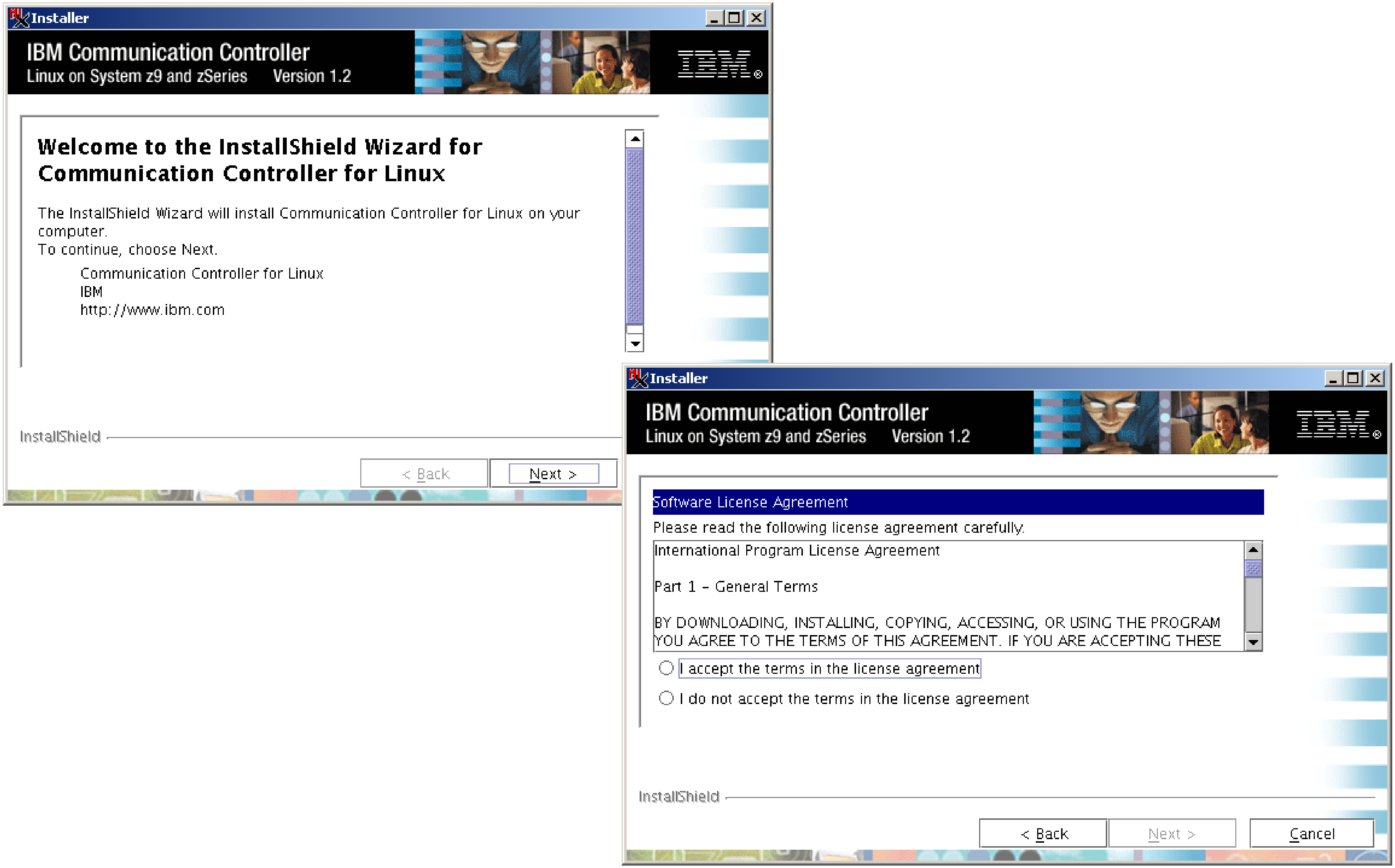
```
linux167:/tmp # tar -xzvf cclv1.2.1.tar.gz
cclv1.2.1/
cclv1.2.1/ndh/
cclv1.2.1/ndh/ndh-1.2.1-1.s390.rpm
cclv1.2.1/README
cclv1.2.1/setuplinux390.bin
linux167:/tmp # cd cclv1.2.1
linux167:/tmp/cclv1.2.1 # export DISPLAY=9.65.224.37:0
linux167:/tmp/cclv1.2.1 # ./setuplinux390.bin &
[1] 25597
linux167:/tmp/cclv1.2.1 #

           Initializing InstallShield Wizard........
           Launching InstallShield Wizard........
```

This example uses an X-Windows server. Graphical installation can also be done using VNC.

CCL installation can also be performed in line mode:

- ./setuplinux390.bin -console

Installshield 1/4


Installer
IBM Communication Controller
Linux on System z9 and zSeries Version 1.2
IBM

Welcome to the InstallShield Wizard for Communication Controller for Linux

The InstallShield Wizard will install Communication Controller for Linux on your computer.
To continue, choose Next.

Communication Controller for Linux
IBM
http://www.ibm.com

InstallShield
< Back
Next >
Installer
IBM Communication Controller
Linux on System z9 and zSeries Version 1.2
IBM

Software License Agreement

Please read the following license agreement carefully.

International Program License Agreement

Part 1 - General Terms

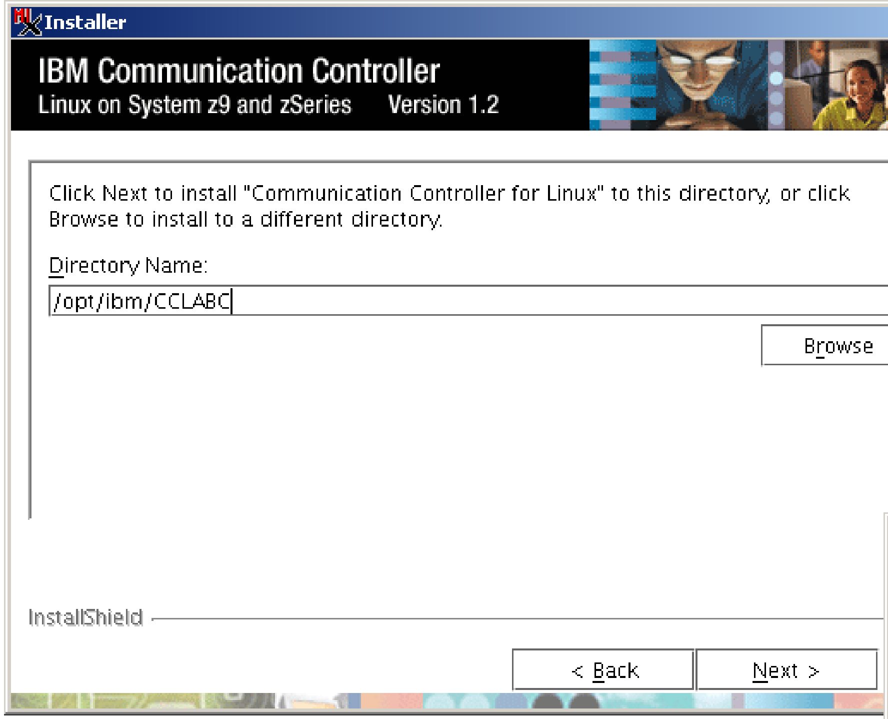
BY DOWNLOADING, INSTALLING, COPYING, ACCESSING, OR USING THE PROGRAM YOU AGREE TO THE TERMS OF THIS AGREEMENT. IF YOU ARE ACCEPTING THESE

○ I accept the terms in the license agreement

○ I do not accept the terms in the license agreement

© 2006 IBM Corporation

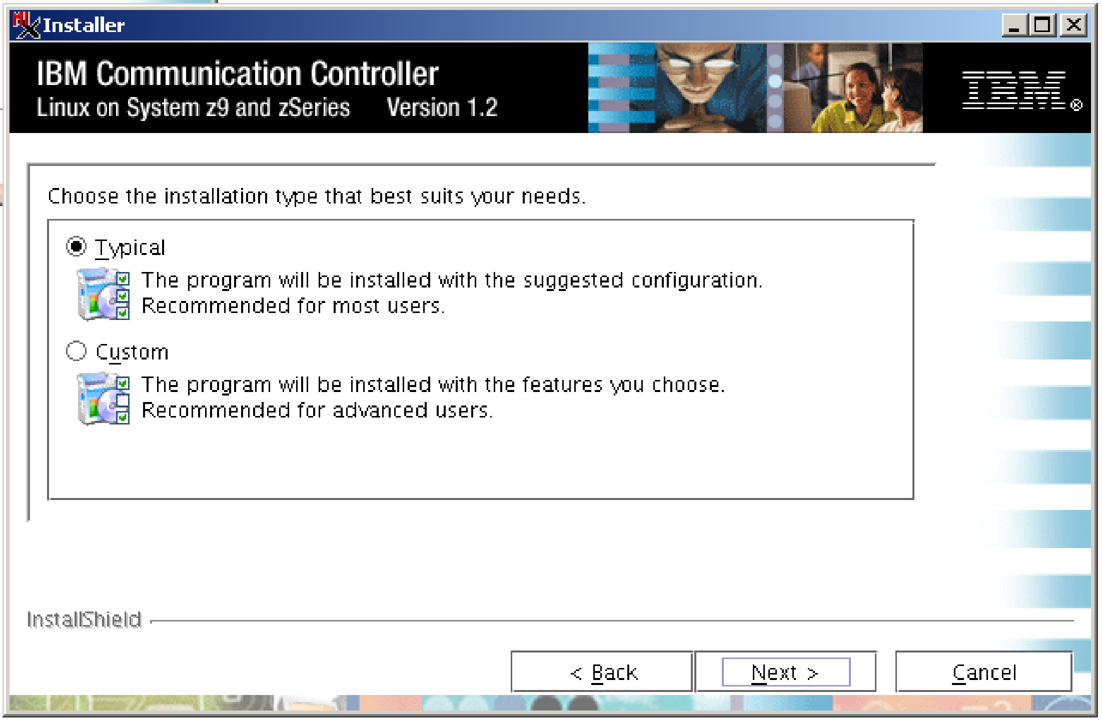
Installshield 2/4


Installer
IBM Communication Controller
Linux on System z9 and zSeries Version 1.2
InstallShield
Browse
< Back
Next >

Click Next to install "Communication Controller for Linux" to this directory, or click Browse to install to a different directory.

Directory Name: /opt/ibm/CCLABC

- **Default install directory is IBM_Communication_Controller_for_Linux**
  - You will have to type that in may times from here on
  - I prefer to change it up here front to something simpler (shorter)

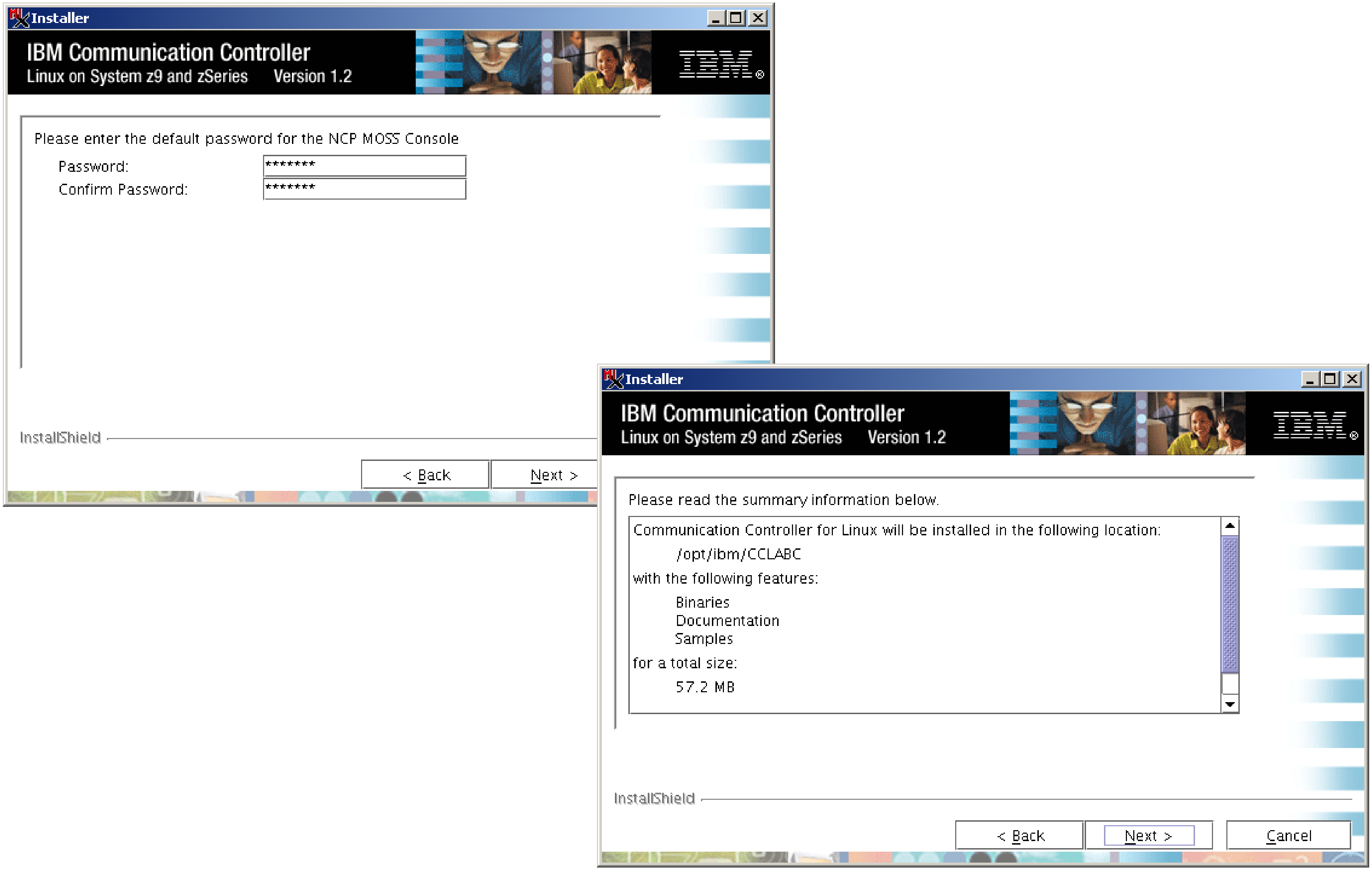

Installer
IBM Communication Controller
Linux on System z9 and zSeries Version 1.2
InstallShield
< Back
Next >
Cancel

Choose the installation type that best suits your needs.

● Typical
The program will be installed with the suggested configuration. Recommended for most users.

○ Custom
The program will be installed with the features you choose. Recommended for advanced users.

Installshield 3/4


Installer
IBM Communication Controller
Linux on System z9 and zSeries Version 1.2
InstallShield
< Back
Next >
Cancel

Please enter the default password for the NCP MOSS Console

Password: *******
Confirm Password: *******

Please read the summary information below.

Communication Controller for Linux will be installed in the following location:

/opt/ibm/CCLABC

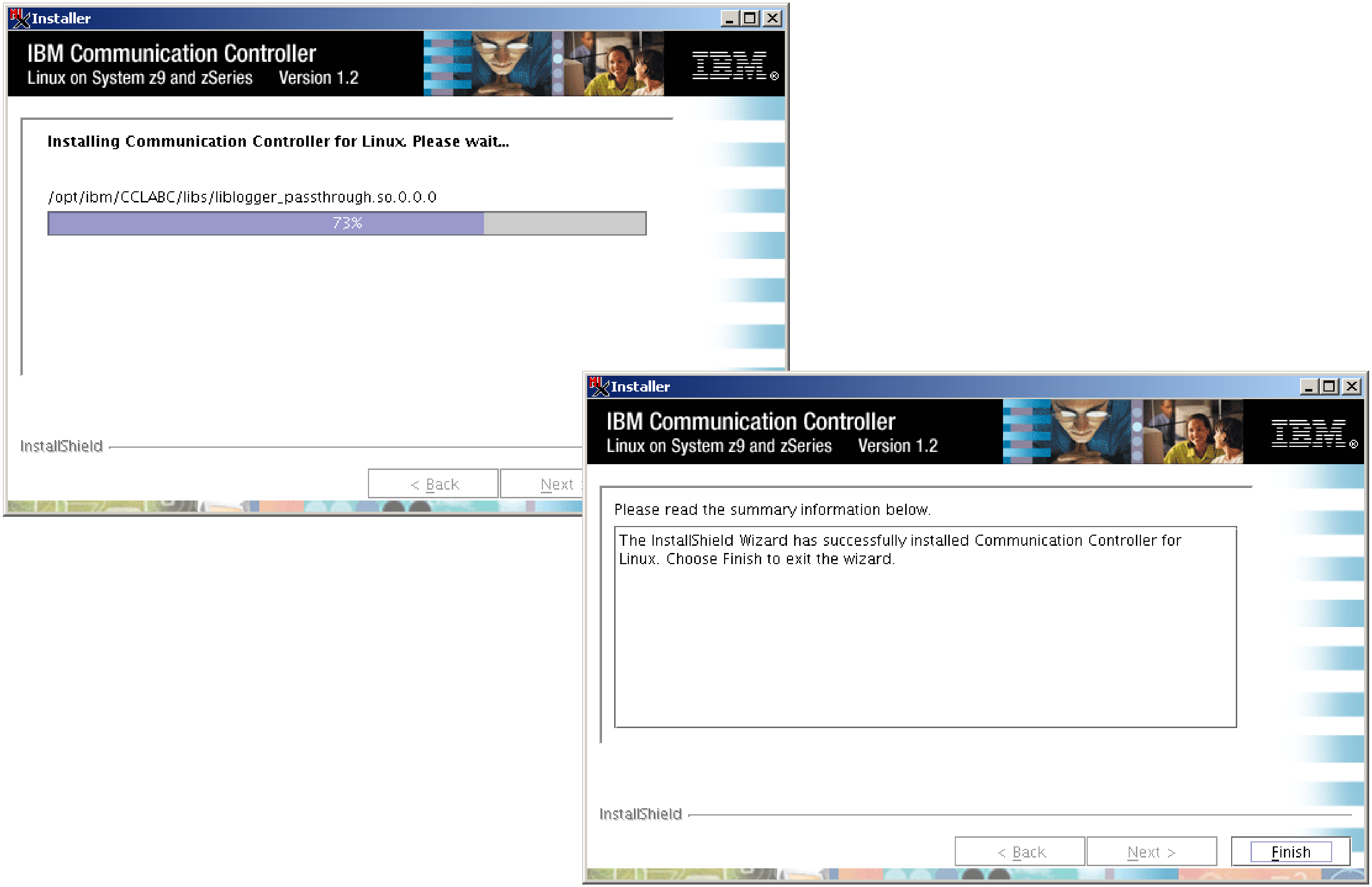
with the following features:

Binaries
Documentation
Samples

for a total size:

57.2 MB

Installshield 4/4


Installer
IBM Communication Controller
Linux on System z9 and zSeries Version 1.2
IBM
InstallShield
< Back
Next >
Finish

Installing Communication Controller for Linux. Please wait...

/opt/ibm/CCLABC/libs/liblogger_passthrough.so.0.0.0

73%

Please read the summary information below.

The InstallShield Wizard has successfully installed Communication Controller for Linux. Choose Finish to exit the wizard.

© 2006 IBM Corporation

Install the NDH rpm

- **NDH is in CCL V1.2.1 shipped as an rpm, and is simply installed using the rpm command of Linux:**

```
linux167:/tmp/cclv1.2.1 # cd ndh
linux167:/tmp/cclv1.2.1/ndh # dir
total 56
drwxr-xr-x  2 501 users    88 Jul 19 14:42 .
drwxr-xr-x  3 501 users   136 Jul 19 14:42 ..
-rw-r--r--  1 501 users 54712 Apr 19 23:27 ndh-1.2.1-1.s390.rpm
linux167:/tmp/cclv1.2.1/ndh # rpm -i --ignorearch ndh-1.2.1-1.s390.rpm
```

- **After having installed the ndh rpm, go over into the /opt/ibm/ndh directory and issue the following command to load the ndh kernel module:**

```
linux167:/tmp/cclv1.2.1/ndh # cd /opt/ibm/ndh/
linux167:/opt/ibm/ndh # ./load_ndh.sh
NDH kernel modules loaded.
linux167:/tmp/cclv1.2.1/ndh # lsmod | grep ndh
ndh                   118088  0
qeth                  243904  1 ndh
linux167:/tmp/cclv1.2.1/ndh #
```

© 2006 IBM Corporation

Directory structure for CCL engines

- **CCL install directory**
  - By default, /opt/ibm/Communication_Controller_for_Linux
    - You can override that to something shorter during the InstallShield dialog - such as /opt/ibm/CCL
  - This is the directory where the ***cclengine*** and ***ccldls*** executables reside
    - You **must** position to this directory before issuing the ***cclengine*** command.
  - Other subdirectories of the install directory are created by the install process
    - **dls-config** contains the DLSw configuration XML file (and its DTD)
    - **docs**, **license** contain the README, User's Guide PDF, license text, man pages, etc.
    - **samples** contains utilities (e.g., canonical), silent install sample config file, etc.
    - **_uninst** contains the uninstaller (and **_jvm** contains its Java Virtual Machine)
    - **libs**, **MOSS_Console** contain files required by the executables
  - Operational directories are created when you run cclengine
    - **logs** contain log files for each combination of CCLEngineName and NCP load module
    - **dumps** contain NCP and CCL engine dumps
    - **traces** contain trace data collected by SIT (F net,TRACE,TYPE=SIT,ID=LINEname)

- **CCL Engine directories**
  - Each CCL engine must be assigned a name, referred to as the **CCL Engine Name**
    - Required parameter of the ***cclengine*** command.
  - The user must create a CCLEngineName subdirectory under the CCL install directory for each engine
    - Assuming we want to start two CCL engines named CCL1 and CCL2:
      - /opt/ibm/Communication_Controller_for_Linux/CCL1
      - /opt/ibm/Communication_Controller_for_Linux/CCL2
  - The binary NCP load module to be run in a given engine must reside in the CCLEngineName directory for that engine
    - Use uppercase letters for the load module file name ***(Linux is case-sensitive!)***

© 2006 IBM Corporation

Starting and Stopping the CCL engine

Starting CCL

- Navigate to CCL Install directory
- Decide on a CCL MOSS console IP port number (default: 2000)
- Specify CCL NCP load module name and CCLEngineName on command
- Use 'nohup' to protect against inadvertently stopping CCL
- Use '&' suffix to run cclengine as a background task
- Can start multiple cclengines, one command for each

```
[root@linux166 ~]# cd /opt/ibm/CCL/
[root@linux166 CCL]# nohup ./cclengine -p20001 -mNCP1 CCL1 &
[1] 2662
nohup: appending output to `nohup.out'
[root@linux166 CCL]# nohup ./cclengine -p20002 -mNCP2 CCL2 &
[2] 2672
nohup: appending output to `nohup.out'
[root@linux166 CCL]# pgrep -fl ccl
2662 ./cclengine -p20001 -mNCP1 CCL1
2672 ./cclengine -p20002 -mNCP2 CCL2
[root@linux166 CCL]#
```

Stop a given CCL engine using cclstop.sh

```
[root@linux166 CCL]# ./cclstop.sh CCL2

CCL engine stopped: CCL2
[2]- Killed                  nohup ./cclengine -p20002 -mNCP2 CCL2
[root@linux166 CCL]# pgrep -fl ccl
2662 ./cclengine -p20001 -mNCP1 CCL1
[root@linux166 CCL]#
```

Operational aspects

© 2006 IBM Corporation

Operating CCL NCPs

- **Operational procedures for the CCL NCP are almost identical to those for an NCP running in the IBM 3745/46 Communication Controller**
  - The VTAM operator console commands and messages are generally unchanged

- **NCP Load Modules managed by VTAM as if on MOSS disk**
  - Use VTAM DISPLAY DISK command to display NCP load modules
  - Use VTAM MODIFY LOAD command to add, replace, purge or rename an NCP load module in the Linux file system
  - Use VTAM MODIFY LOAD to set CCL to automatically reload the designated NCP at a scheduled time without any operator action (Timed IPL)

- **CCL MOSS Console**
  - Provides a set of IBM 3745/46 MOSS-like functions that are accessed via a web browser
  - Functions provided include:
    - Starting/stopping the CCL Engine
    - Reloading the CCL Engine with the active NCP
    - Dumping the NCP or CCL Engine
    - Managing the NCP/CCL Engine Dumps
    - Diagnostic traces
    - Displaying and altering NCP storage/general registers/local registers

© 2006 IBM Corporation

Monitoring CCL NCP

- **NTuneMON**
  - Supported by CCL at the release level supported by the corresponding NCP without changes to NTuneMON
  - ATUSS panel displays a unique character string when it is used to monitor CCL NCPs
    - Under “3745 HARDWARE INFO”, the MICROCODE EC field will show the CCL version and release, such as CCL1.2.1
    - Under “3745 HARDWARE INFO”, the FIX field will show the CCL package build date
  - CCL Engine will not provide CCU utilization so this will be reported as zero

- **NPM**
  - CCL Engine will not report CCU or TIC utilization

- **System Automation for z/OS V2R3**
  - SA's "Processor Operations" automation feature can automate any Linux on System z LPAR or guest under z/VM.
  - Startup, shutdown, and monitor Linux on System z itself
  - Startup, shutdown, and monitor CCL instances
  - The SA automation can handle any messages coming from Linux on System z and/or CCL as well as proactively monitor CCL itself by issuing CCL commands and having SA parse the results.

© 2006 IBM Corporation

Problem determination - NDH display commands

NDH socket list

```
[root@linux166]# cat /proc/net/ndh/socklist
NDH9700I SOCKLIST - Revision:1.78.1.8
ReadSock-Inode   WriteSock-Inode  UID      PROTO STATE          MAC-SAP Pairs
14926            14927            0        NDH-TR CONNECTED
                                                                41000d14dddd-04
                                                                41000d14dddd-14
14691            14691            0        NDH-TR CONNECTED
9651             9652             0        NDH-OSN CONNECTED
                                                             000000000000-03050004
9647             9648             0        NDH-TR CONNECTED
                                                                020080668181-04
9644             9644             0        NDH-X25 NOT CONNECTED
                                                             D14MCH
9632             9632             0        NDH-? NOT CONNECTED
NDH9700I SOCKLIST END
[root@linux166]#
```

T/R

DLSw

CDLC

eth

X.25

CCL

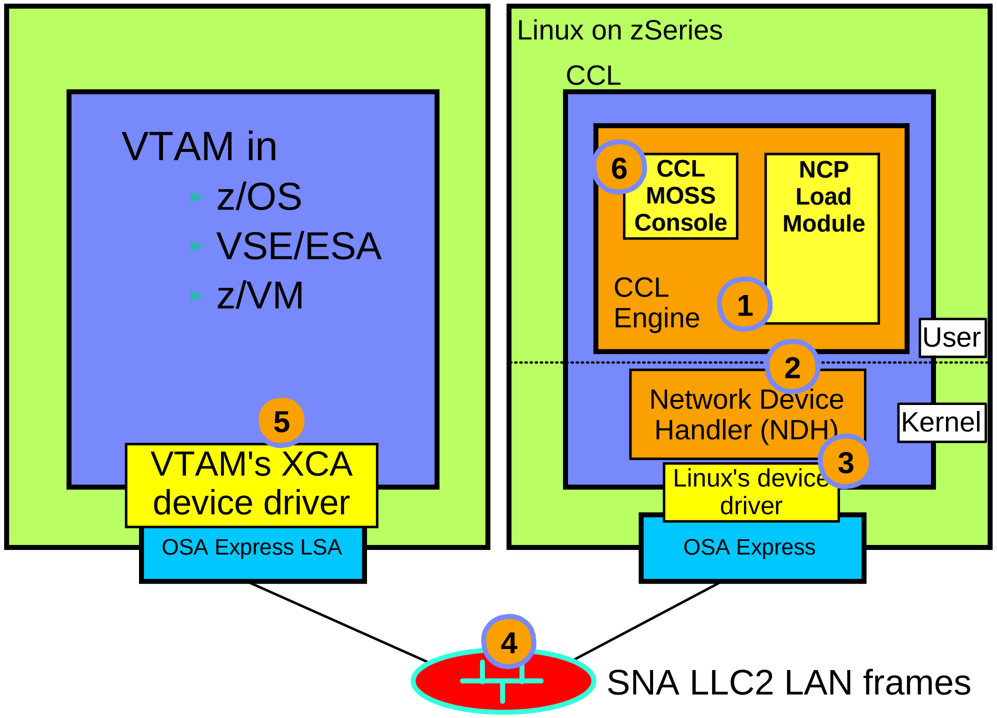
NDH statistics

```
[root@linux166]# cat /proc/net/ndh/statistics
SOCKETPAIR: 9647 9648  Name: eth1  MAC: 02:00:80:66:81:81
SAPS: 04
     Inbound User-To-NDH
          1966 packets          65707 byte
          0 packets discard 0 byte discard
          0 packed pkts     1966 nonpacked pkts
     Outbound NDH-To-User
          65 packets          1820 byte
          0 packets discard 0 byte discard
          0 packed pkts     65 nonpacked pkts
```

© 2006 IBM Corporation

Problem determination - diagnostic traces for SNA LLC2 LAN flows

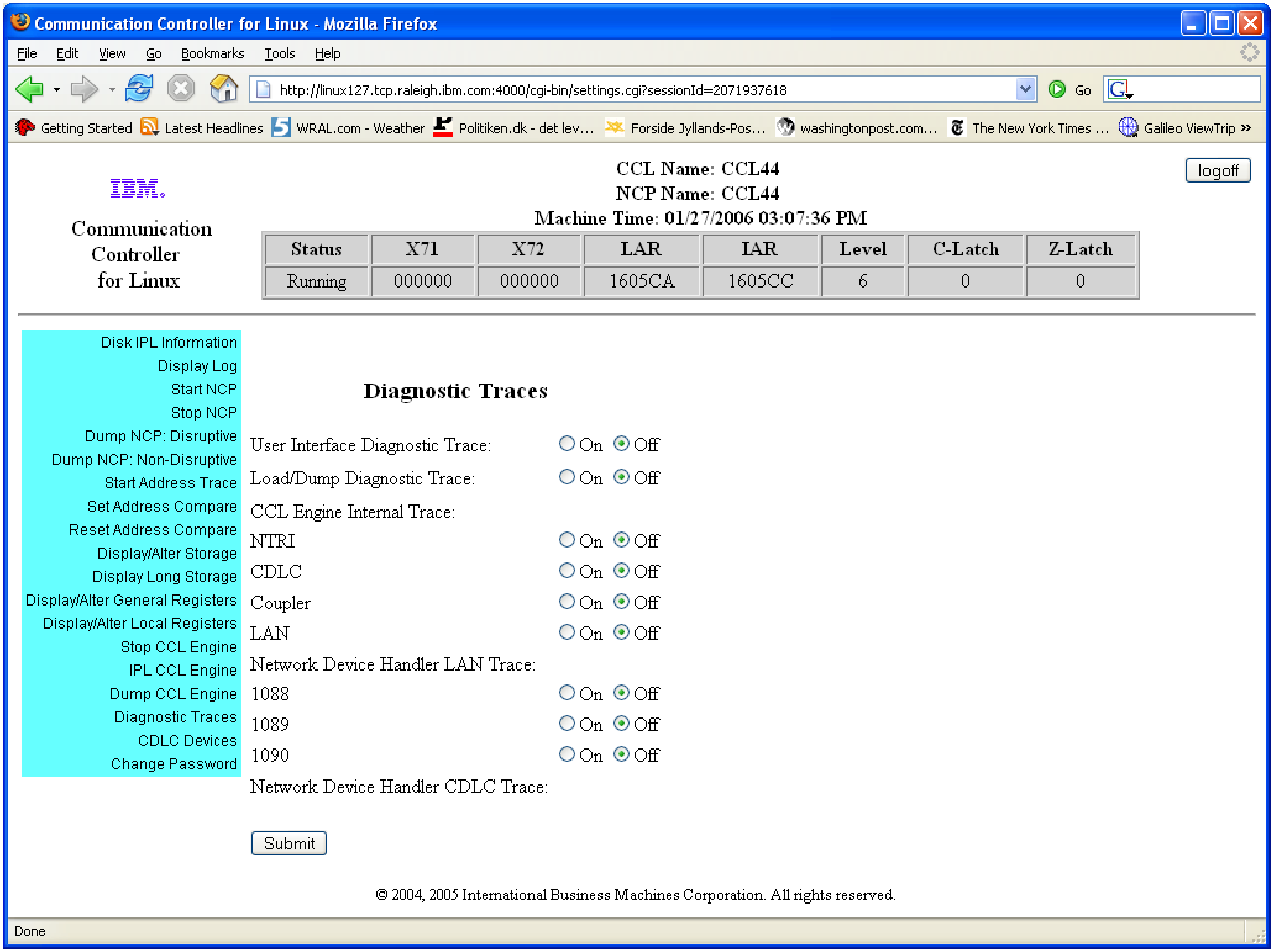
1. **Data flows between CCL Engine and NCP**
   - VTAM MODIFY TRACE, TYPE=LINE
2. **Data flows between CCL Engine and CCL NDH**
   - Trace data as seen from CCL Engine using Network Device Handler LAN Trace on MOSS Console or VTAM MODIFY TRACE, TYPE=SIT.
     - Trace data is written to Linux file
     - Format using the ccltap utility on Linux.
   - Trace data as seen from CCL NDH using CCL Engine Internal Trace on MOSS Console
3. **Data flows from CCL NDH to Linux LCS Device Driver**
   - change the debug value stored in /proc/net/ndh/debug
4. **Data flows on the network**
   - Sniffer trace (Ethereal may be used)
5. **Data flows in and out of VTAM**
   - VTAM MODIFY TRACE,TYPE=IO
   - VTAM MODIFY TRACE, TYPE=VTAM,OPTION=(PIU,LCS,CIO)
6. **Captures every HTTP request processed by the CCL Moss Console**
   - Enabled via User Interface Diagnostic Trace on MOSS Console



There are additional/specialized trace points for OSN and IPTG connectivity.

© 2006 IBM Corporation

MOSS Console - diagnostic traces



IBM.

Communication
Controller
for Linux

CCL Name: CCL44
NCP Name: CCL44
Machine Time: 01/27/2006 03:07:36 PM

Status	X71	X72	LAR	IAR	Level	C-Latch	Z-Latch
Running	000000	000000	1605CA	1605CC	6	0	0

Diagnostic Traces

User Interface Diagnostic Trace:	○ On ● Off
Load/Dump Diagnostic Trace:	○ On ● Off
CCL Engine Internal Trace:	
NTRI	○ On ● Off
CDLC	○ On ● Off
Coupler	○ On ● Off
LAN	○ On ● Off
Network Device Handler LAN Trace:	
1088	○ On ● Off
1089	○ On ● Off
1090	○ On ● Off
Network Device Handler CDLC Trace:	

© 2004, 2005 International Business Machines Corporation. All rights reserved.

© 2006 IBM Corporation

CCL problem determination - dumps

- **For an NCP ABEND, both an NCP dump and a CCL Engine dump will be taken automatically if the automatic dump/load switch is on.**
  - Both dumps will be in the ./dumps directory in the Linux file system under the CCL install directory
  - New dumps will overwrite existing dumps in the same directory
  - CCL Engine will automatically restart and reload the NCP
  - If the automatic dump/load switch is OFF, no dumps will be taken for an NCP ABEND and the CCL Engine will not automatically restart and reload the NCP.

- **VTAM operator can dump the NCP by using the MODIFY DUMP ACTION=STORE,OPTION=STATIC command.**
  - Dumps will be taken of both the NCP and CCL Engine
  - Both dumps will be in the ./dumps directory in the Linux file system under the CCL install directory
  - This is a disruptive dump. NCP processing stops and is deactivated. CCL Engine will automatically restart and reload the NCP. The NCP major nodes needs to be reactivated in VTAM.
  - Dump command will be rejected if an NCP dump exists in the Linux file system. Old dump needs to be previously purged.

- **VTAM operator can dump NCP by using the MODIFY DUMP ACTION=COMP,OPTION=DYNA, DUMPDS=name**
  - NCP processing continues while the contents of NCP are dumped
  - Dump is saved in VTAM host file identified by DUMPDS operand

- **MOSS Console can be used to dump NCP and CCL Engine**
  - Dump NCP Disruptive
  - Dump NCP Non-disruptive
  - Dump CCL Engine – non-disruptive and saved to Linux file system in the ./dumps directory

© 2006 IBM Corporation

CCL problem determination - messages

- **Messages generated by the CCL Engine**
  - ERROR, WARNING, and INFO messages from the CCL Engine are written to the Engine log
    - file in the ./logs subdirectory of the CCL install directory.
    - file name: *CCLEngineName.NCPname*.log
  - ERROR messages are also written to the Linux system messages log
    - /var/log/messages
  - Both log files can be viewed from the MOSS Console “Display Logs” panel
    - Engine log = *CCLEngineName.NCPname*.log
    - Syslog = /var/log/messages
- **All messages are listed in the CCL Implementation and Users Guide manual**
  - For example, this message is logged when NCP is loaded into CCL:

```
[Aug  1 07:14:25.394819]: UT 3454 INFO CCZ1006I - NCP load module is loaded
```

  - Message description in the manual:

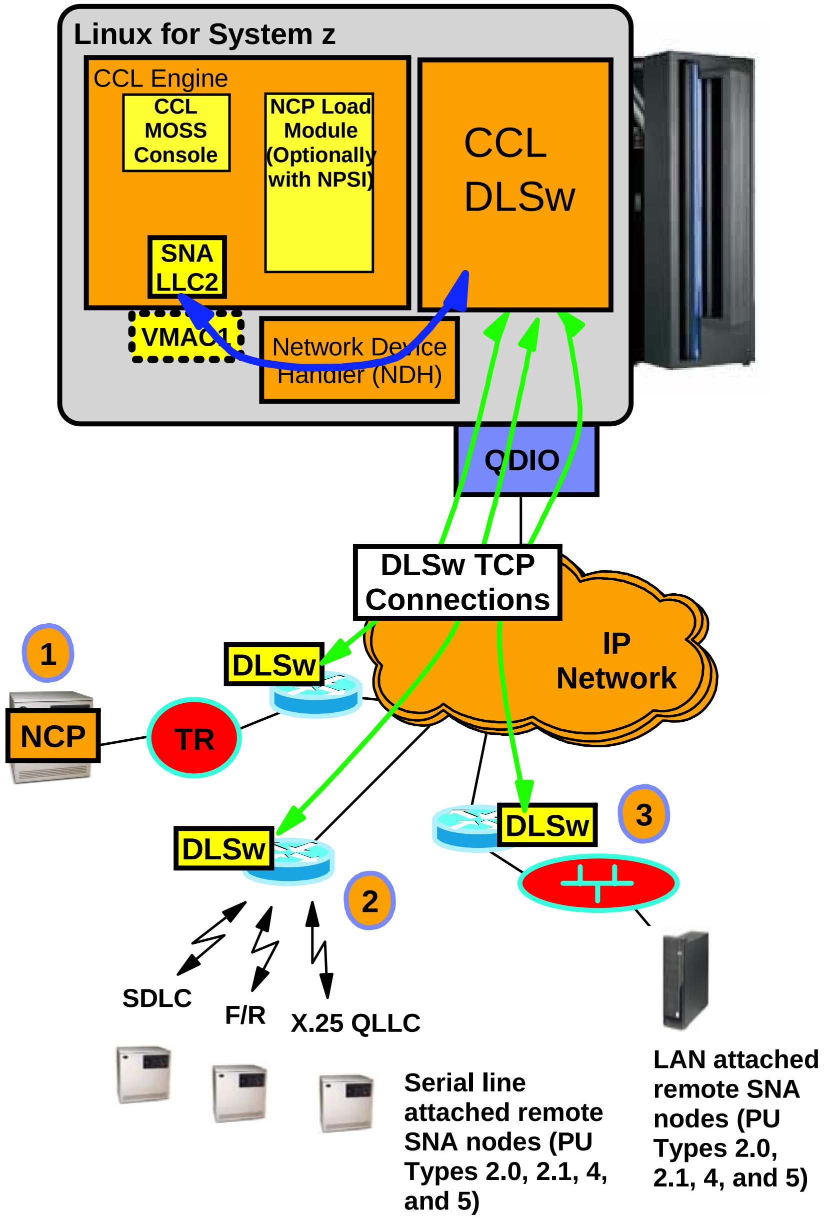
CCZ1006I	NCP load module is loaded
Explanation:	The NCP load module file has been loaded into CCL Engine storage.
System action:	None.
User response:	None.

© 2006 IBM Corporation

Migration Scenario 2:
CCL DLSw

© 2006 IBM Corporation

CCL imbedded DLSw support - recap


Linux for System z
CCL Engine
CCL MOSS Console
NCP Load Module (Optionally with NPSI)
SNA LLC2
VMAC1
Network Device Handler (NDH)
CCL DLSw
QDIO
DLSw TCP Connections
IP Network
1
NCP
TR
DLSw
DLSw
2
DLSw
3
SDLC
F/R
X.25 QLLC
Serial line attached remote SNA nodes (PU Types 2.0, 2.1, 4, and 5)
LAN attached remote SNA nodes (PU Types 2.0, 2.1, 4, and 5)

Network infrastructure simplification:

- Integration of data center DLSw functions with NCP functions in Linux on System z
- Avoids or reduces the need for separate data center DLSw router equipment

CCL DLSw is based on the open standards version of DLSw - RFC1795 & RFC2166:

- Interoperability with all vendors who have implemented DLSw according to those standards
- Will interoperate with DLSw+ nodes
  - DLSw+ nodes will adapt to standard DLSw protocols when connecting to an open standards-based DLSw implementation

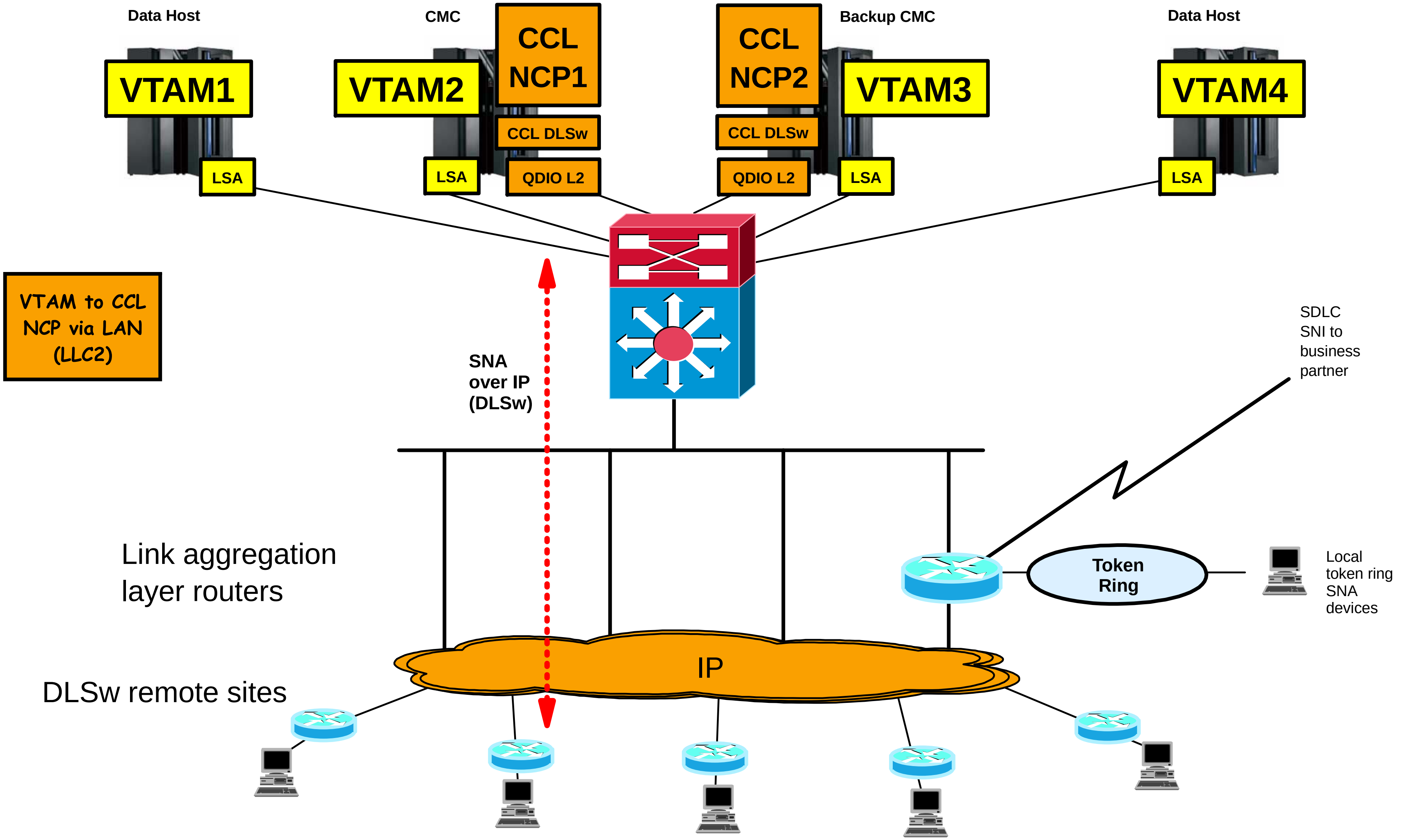
Typical CCL DLSw scenarios:

1. INN/SNI to NCPs in remote IBM 3745/46
2. Peripheral nodes attached via serial lines to DLSw router
3. Peripheral nodes attached via LAN to DLSw router

Virtualizes the SNA MAC address

- DLSw provides connectivity for all CCL NCP MAC addresses that do ***not*** match any MAC address on the Linux system
- Supports many virtual DLSw MAC addresses over a single physical MAC address (OSA port)
  - Especially of value on mainframe hardware platforms where QDIO layer-2 mode isn't available

Migration to CCL using CCL DLSw (diagram)



© 2006 IBM Corporation

Migration to CCL using CCL DLSw (description)

- **CCL DLSw is automatically installed as part of CCL V1.2.1**
  - Just another executable in the CCL install directory
  - Needs to be configured and started along with cclengine

- **Configure CCL DLSw's IP address as a peer in the remote DLSw routers**
  - You can choose whether or not to define the remote DLSw routers to CCL DLSw
    - CCL DLSw can dynamically learn about DLSw peers
  - Remote DLSw routers flow the SNA data over IP all the way into the Linux image
    - Instead of "IP to the aggregation router, and SNA LLC2 into the CCL Linux"

- **NCP definitions are unchanged (still have duplicate TICs defined)**

- **DO NOT create a Linux device with the NCP's duplicate TIC address on either system**
  - ***No need for separate VLANs***
    - NCP's duplicate MAC never gets onto the wire
    - CCL DLSw terminates the NCP's LLC2 and converts to IP before the SNA data leaves the Linux system

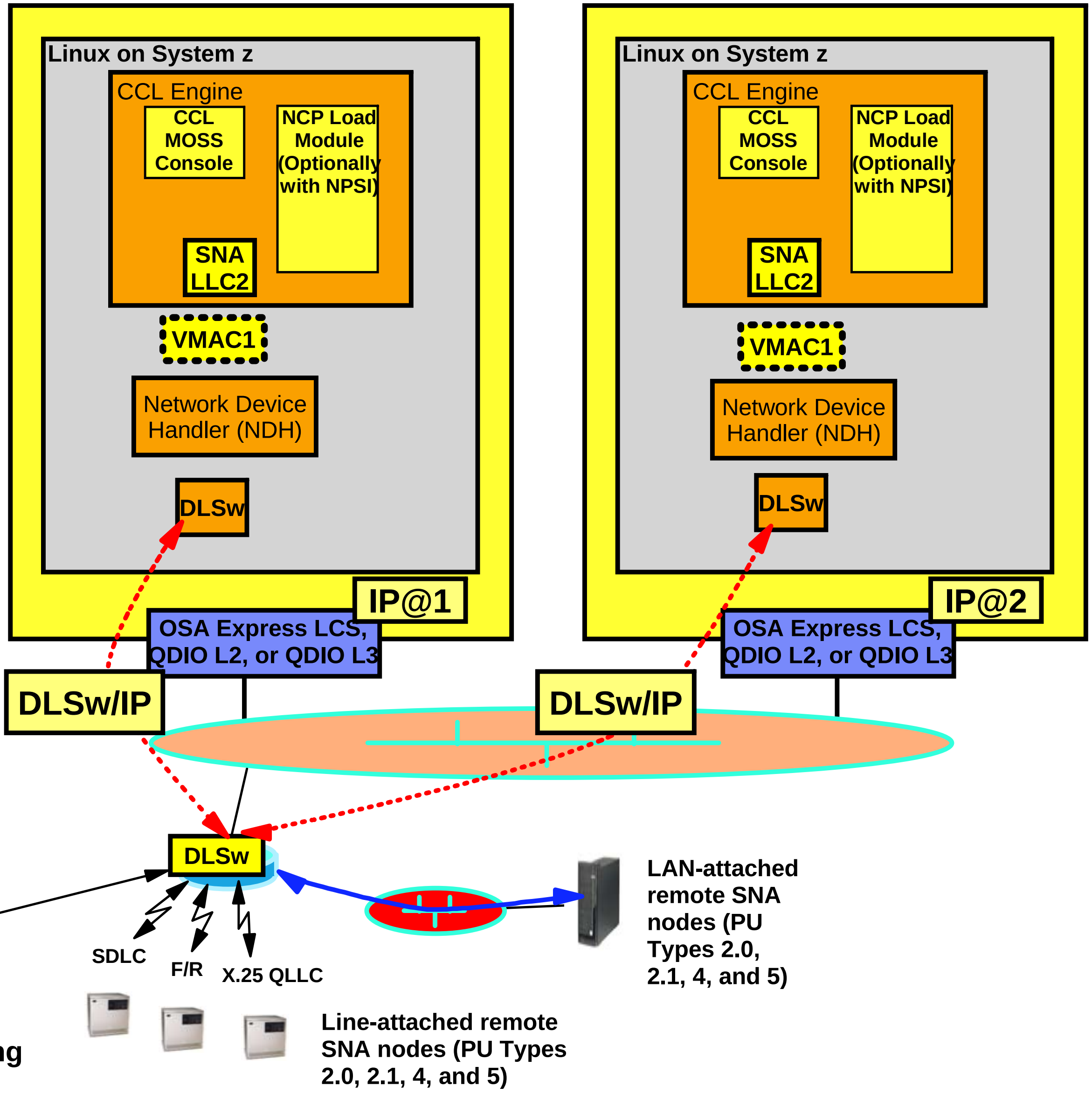
- **DLSw router supporting moved SDLC lines can also be peer to CCL DLSw**
  - Instead of using Local DLSw conversion to LAN LLC2

- **Use DLSw instead of SR-TB to get local token ring traffic into CCL NCP**
  - ***All downstream traffic comes into the mainframe as IP !***

© 2006 IBM Corporation

Duplicate MAC addressing for load-balancing and redundancy - imbedded DLSw support

- **Duplicate TIC addressing is a token-ring only technology**
  - It is not supported by an Ethernet LAN infrastructure
  - Duplicate TIC address topologies cannot be directly migrated to Ethernet
- **The same objectives can be addressed by using a DLSw infrastructure**
  - "Remote" DLSw node defines two CCL NCP DLSw peers
  - Both peers support the same SNA MAC address
    - Both CCL/NCPs have a token-ring LINE with that same MAC address defined
  - The remote DLSw node can be configured to balance LLC2 connections over the two peers when both are available
  - Simple round-robin and circuit-weighted load balancing are typically supported


Linux on System z
CCL Engine
CCL MOSS Console
NCP Load Module (Optionally with NPSI)
SNA LLC2
VMAC1
Network Device Handler (NDH)
DLSw
IP@1
OSA Express LCS, QDIO L2, or QDIO L3
DLSw/IP
Linux on System z
CCL Engine
CCL MOSS Console
NCP Load Module (Optionally with NPSI)
SNA LLC2
VMAC1
Network Device Handler (NDH)
DLSw
IP@2
OSA Express LCS, QDIO L2, or QDIO L3
DLSw/IP
DLSw
SDLC
F/R
X.25 QLLC
LAN-attached remote SNA nodes (PU Types 2.0, 2.1, 4, and 5)
Line-attached remote SNA nodes (PU Types 2.0, 2.1, 4, and 5)

Two remote peers:
- **IP@1**
- **IP@2**

Load-balancing either round-robin or circuit weighting

Configuring CCL DLSw

- **Edit the DLSw configuration file**
  - <CCL_Install_Directory>/dls-config/dlscfg.xml
    - Note: Do *not* modify the dlscfg.dtd file!

- **Decide what IP port number to use for the DLSw console**

```
<DLSw:console_listening_port value="2002"/>
```

- **Decide how many DLSw sessions to support**

```
<DLSw:max_dls_session value="1000"/>
```

- **Decide whether to allow DLSw peers that are not predefined**
  - to prevent: `<DLSw:dynamic_peer value="disabled"/>`
    - Can be enabled/disabled dynamically once DLSw is running

- **Predefine DLSw peers by adding enabled <DLSw:peer> records:**

```
<!-- DLSw TCP Peer configuration -->
             <DLSw:peer>
                <DLSw:enable value="yes"/>
                <DLSw:hostname value="192.168.1.100"/>
                <DLSw:connection_type value="passive"/>
                <DLSw:keepalive value="disabled"/>
                <DLSw:priority value="medium"/>
            </DLSw:peer>
```

  - DLSw peer definitions can be added dynamically once DLSw is running

© 2006 IBM Corporation

Configuring CCL DLSw...

- **Decide which SAPs to support over DLSw:**
  - Default config includes SAPs 4, 8, C

```
<!-- LLC Interface SAP Configuration -->
             <DLSw:LLC-Interface>
                <DLSw:sap_num value="4"/>
                <DLSw:sap_num value="8"/>
                <DLSw:sap_num value="c"/>
                <DLSw:sap_num value="14"/>
             </DLSw:LLC-Interface>
```

  - Add a sap_num record for any SAP value that NCP or any remote node might use
    - SAPs cannot be added dynamically, so do this before you start DLSw!

  - ***CCL DLSw does NOT support HPR***
    - SAP 200 (0xC8) does not need to be defined here

- **There are a multitude of other configuration parameters!**
  - The default values in the sample dlscfg.xml file work well for them.

© 2006 IBM Corporation

Operating CCL DLSw

Starting CCL DLSw

- Must be started from the CCL install directory
- Use 'nohup' and '&' (same considerations as cclengine)

```
cd /opt/ibm/Communication_Controller_for_Linux
nohup ./ccldls &
```

- Only ***one*** instance of CCL DLSw per Linux image
  - Even if there are multiple CCLs
- NDH must be running before ccldls is started
  - If not, ccldls prints error CCZD503E and exits
- ccldls can be started before or after the CCL engine(s)
- If XML parsing error is encountered, error messages are written to stdout and ccldls exits
  - The log file name is one of the parameters in the XML file, so if the XML doesn't parse correctly, ccldls does not know where the log entries should go!
  - If you use nohup, remember that ccldls stdout is piped to file "nohup.out"

Stopping CCL DLSw

```
pkill -9 ccldls
```

© 2006 IBM Corporation

Monitoring CCL DLSw - login to DLSw console

- **DLSw console is a text interface**
  - Telnet/SSH to the system running ccldls, then telnet 127.0.0.1 <console_listening_port value>
    - console_listening_port is in the dlscfg.xml file, default 2002
  - Login using default MOSS console password
    - Can set a DLSw-specific password using **`createPassword`** tool
  - Be sure to use a telnet client with lots of *scrollback* capability
    - Some DLSw console commands produce LOTS of output!

- **Example:**

```
[root@linux166 ~]# telnet localhost 2002
Trying 127.0.0.1...
Connected to localhost.localdomain (127.0.0.1).
Escape character is '^]'.

DLS Password: >
CCZD607I - DLS_607: ####################################################
CCZD608I - DLS_608: ##                                                ##
CCZD609I - DLS_609: ##                DLSw for Linux                  ##
CCZD608I - DLS_608: ##                                                ##
CCZD610I - DLS_610: ##            DLS  V1.2.1 (Build 04-19-06)        ##
CCZD608I - DLS_608: ##                                                ##
CCZD607I - DLS_607: ####################################################
DLSw>?
Data Link Switching: Top level commands
 add              add DLS parameters
 clear            clear DLS statastics
 delete           delete DLS parameters
 disable          disable DLS parameters
 enable           enable DLS parameters
 show             display DLS parameters
 ? (help)         print help information
 quit             quit DLS console

DLSw>
```

Monitoring CCL DLSw - 'show tcp' commands

- **Which DLSw peer nodes am I connected to?**

```
DLSw>show tcp peer
   Multicast
   IP Address      IP Address      Conn State     CST Version  ActSes SesCreates
   --------------- --------------- -------------- --- -------- ------ ----------
1                  9.42.103.140    ESTABLISHED     a  AIW V2R0      1          1

DLSw>
```

- **How much data am I sending to the DLSw peer at 9.42.103.140?**

```
DLSw>show tcp stat 9.42.103.140
                                     Transmitted           Received
                                     -----------          -----------
Data Messages                                 13                   14
Data Bytes                                   869                  822
Control Messages                              11                    8

CanYouReach Explorer Messages                  1                    2
ICanReach Explorer Messages                    1                    1
DLSw>
```

© 2006 IBM Corporation

Monitoring CCL DLSw - 'show dls' commands ...

➢ What connections exist to DLSw attached SNA nodes?

```
DLSw>sh dls sess
         Source          Destination      State     Flags    Dest IP Addr    Id
     ---------------  ---------------  ---------  -------  --------------  ----
   1 41000d14dddd 14  40000d16dddd 04  CONNECTED           9.42.103.140       0
DLSw>
```

➢ What is happenning on DLSw session with Id=0?

```
DLSw>sh dls sess detail 0
         Source          Destination      State     Flags    Dest IP Addr    Id
     ---------------  ---------------  ---------  -------  --------------  ----
   1 41000d14dddd 14  40000d16dddd 04  CONNECTED           9.42.103.140       0

      Personality:      ORIGINATOR
      XIDs sent:        3
      XIDs rcvd:        2
      Datagrams sent:   0
      Datagrams rcvd:   0
      Info frames sent: 471
      Info frames rcvd: 474
      RIF:
      Local CID :       0057f590:7e000000
      Remote CID:       00586ce0:7e000000
      Priority:         MEDIUM

          Receiver                       Sender
      -----------------            -----------------
      InitialWindowAdv: 12         InitialWindowRcv: 12
      CurrentWindow:    15         CurrentWindow:    17
      GrantedUnits:     25         GrantedUnits:     17
      LargestWindow:    15         LargestWindow:    17
      RcvQBytes:        0          SendQBytes:       0
      RcvQLimit:        20480      SendQLimit:       32768
      HalveOpsSent:     0          HalveOpsRecv:     0
      ResetOpsSent:     0          ResetOpsRecv:     0
DLSw>
```

© 2006 IBM Corporation

Monitoring CCL DLSw - 'show llc' commands ...

What LLC connections exist to CCL?

```
DLSw>sh llc sess
Sessions for SAP 0:
No sessions for SAP 0.
Sessions for SAP 4:
Session ID                                          Local
(int-sap-id)  Remote MAC          Local MAC         SAP     State
0000-04-0000  40:00:0d:16:dd:dd  41:00:0d:14:dd:dd  14      LINK_OPENED
Sessions for SAP 8:
No sessions for SAP 8.
Sessions for SAP c:
No sessions for SAP c.
Sessions for SAP 14:
No sessions for SAP 14.
DLSw>
```

What is happenning on LLC session with ID=0000-04-0000?

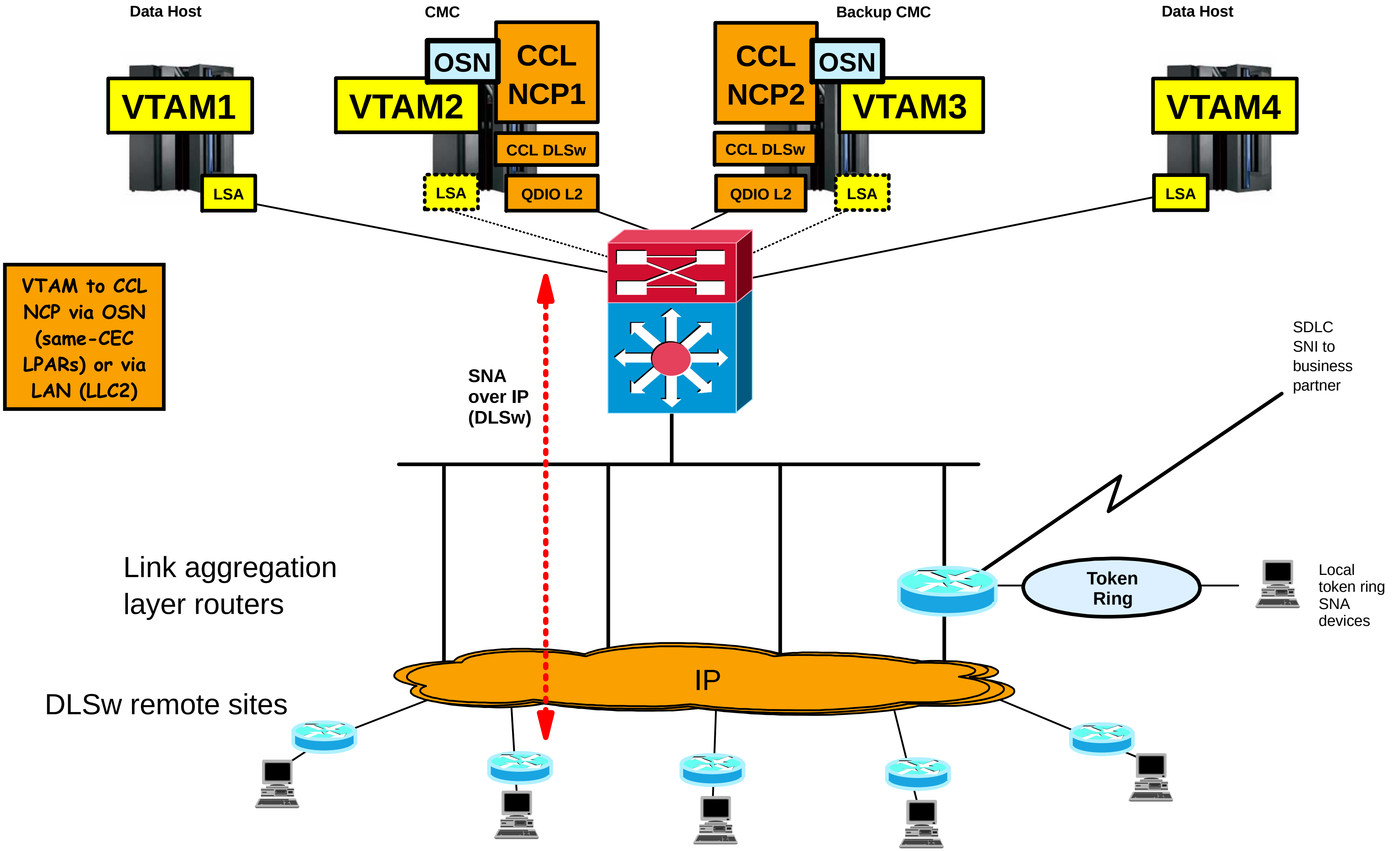
```
DLSw>sh llc sess 0000-04-0000
Session ID:                        0000-04-0000
Interface:                         05,ndh0
Local MAC addr:                   41:00:0d:14:dd:dd
Remote MAC addr:                   40:00:0d:16:dd:dd
Local SAP:                          14
Remote SAP:                         04
RIF:                               None
Access Priority:                   0
State:                             LINK_OPENED
Reply Timer(T1):                   3 sec
Receive ACK Timer(T2):             3 100milisec
Inactivity Timer(Ti):              30 sec
MAX I-Field Size(N1):              16384
MAX retry Value(N2):               8
Rcvd I-frames before Ack(N3):      2
Transmit Window Size(Tw):          7
Working Transmit Size(Ww):         7
Acks Needed to Inc Ww(Nw):         1
```

```
Current Send Seq (Vs):             14
Current Rcv Seq (Vr):              13
Last ACK'd sent frame(Va):         14
No. of frames in ACK pend q:       0
No. of frames in Tx pend q:        0
Local Busy:                        NO
Remote Busy:                       NO
Poll Retry count:                  8
Appl output flow stopped:          NO
Send process running:              YES

Frame Type           Xmt       Rcvd
I-frames:             14         13
RR-frames:             5          6
RNR-frames:            0          0
REJ-frames:            0          0
I-frames Discarded by LLC:        0
I-frames Refused by LLC user:     0
DLSw>
```

Appendix A:
CDLC channel connectivity to CCL

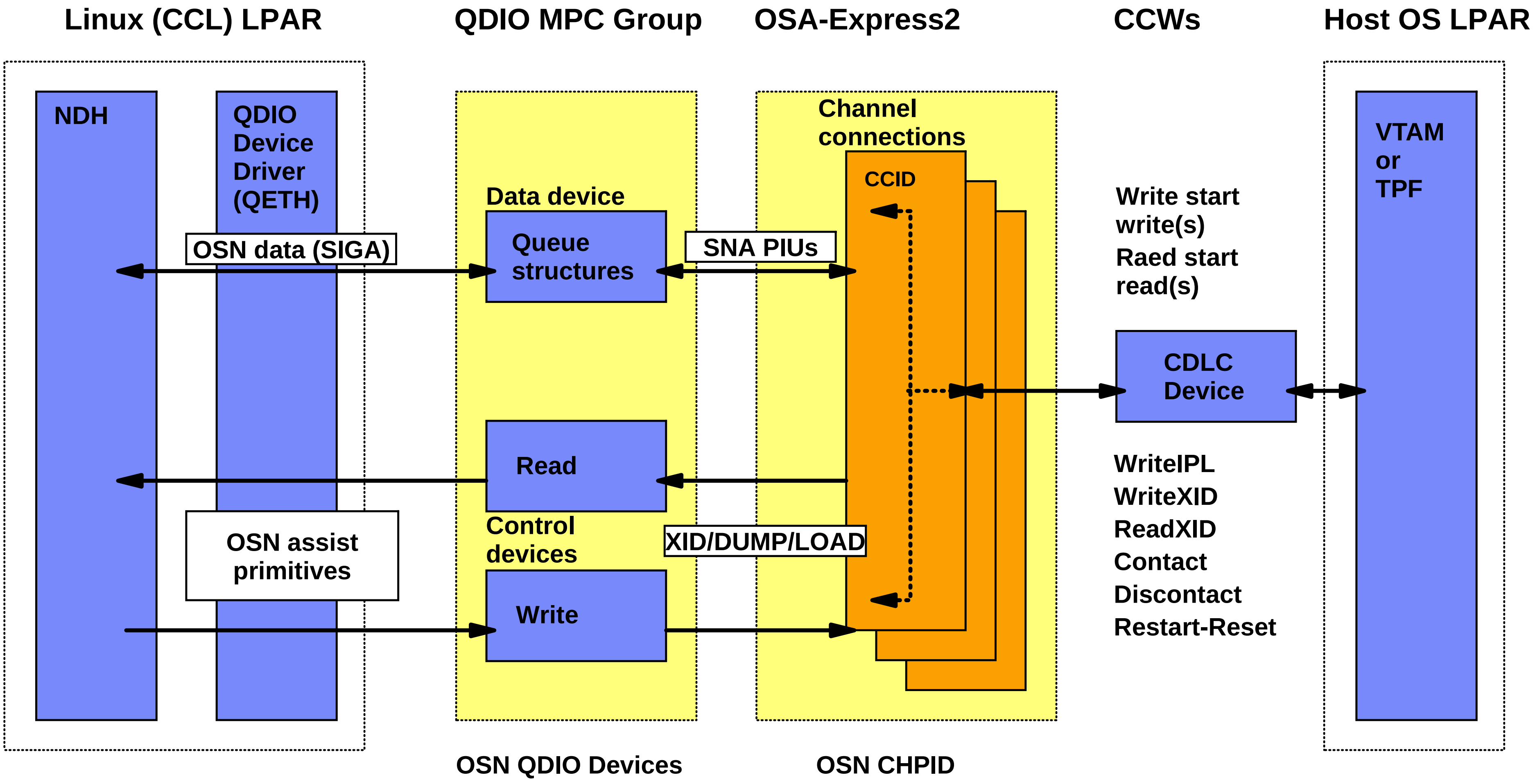
© 2006 IBM Corporation

Migration to CCL using CCL DLSw and OSN CDLC (diagram)



© 2006 IBM Corporation

OSN component overview



Linux (CCL) LPAR

QDIO MPC Group

OSA-Express2

CCWs

Host OS LPAR

NDH

QDIO Device Driver (QETH)

OSN data (SIGA)

OSN assist primitives

Data device

Queue structures

Read

Control devices

Write

SNA PIUs

XID/DUMP/LOAD

Channel connections

CCID

Write start
write(s)
Raed start
read(s)

CDLC Device

WriteIPL
WriteXID
ReadXID
Contact
Discontact
Restart-Reset

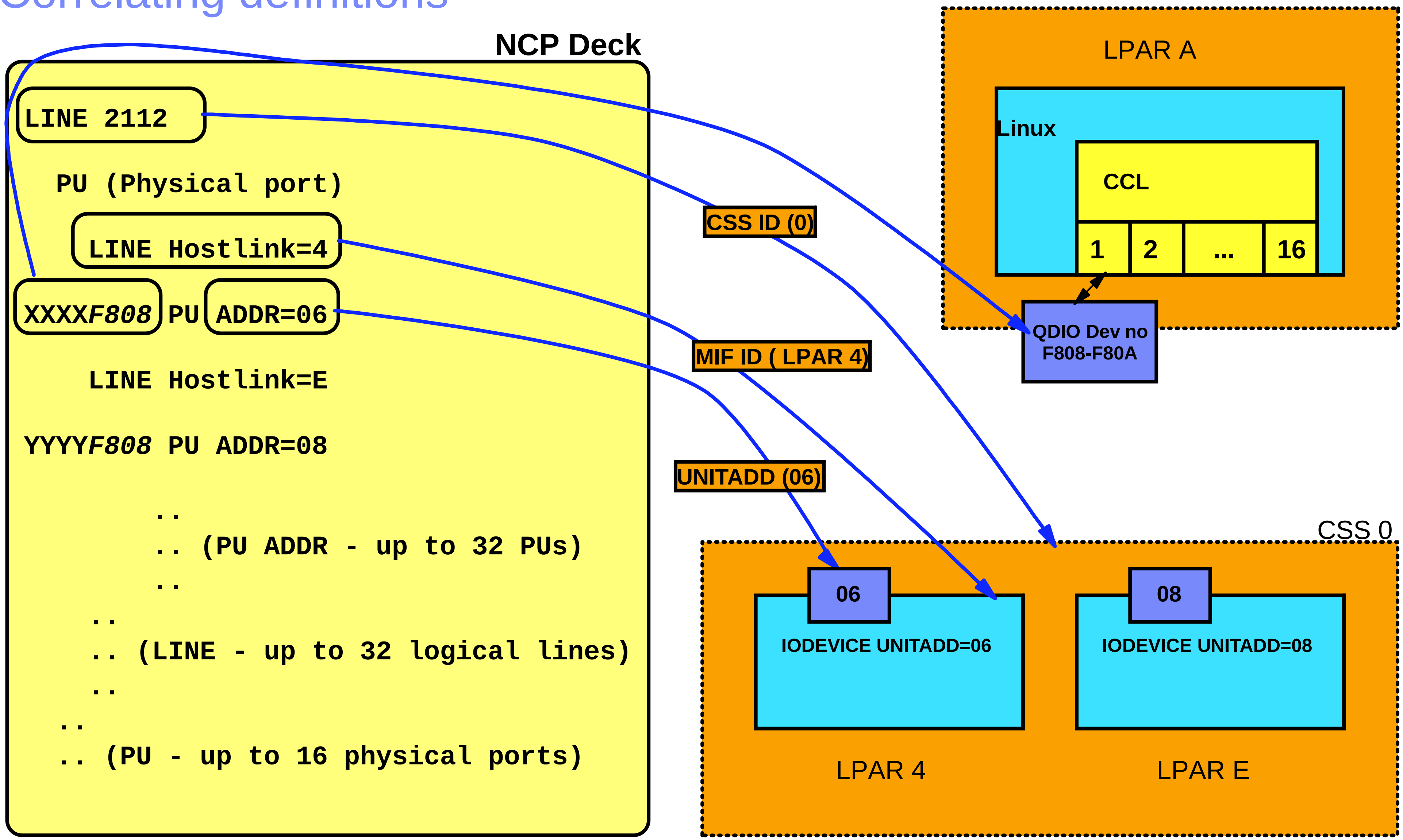
VTAM or TPF

OSN QDIO Devices

OSN CHPID

© 2006 IBM Corporation

Correlating definitions



If the above naming and parameter guidelines cannot be met, mapping between NCP definitions and the information needed in the OSN context, can be encoded in a CCLDEFS text file.

© 2006 IBM Corporation

Loading an NCP over the CDLC channel

- **CCL Load/Dump can be performed in one of two ways.**
  - The CCL can be loaded with an NCP
  - The CCL can be running without an NCP

- **If the CCL is loaded with an active NCP and a WIPL command is received on the connection defined as the IPL port, then the CCU and communication threads are terminated, and the CCL load/dump threads are started to continue the load/dump process.**

- **To start a CCL without NCP, the CCL must be started using the following:**
  - ./cclengine CCLEngineName -m cclcldp [-p xxxx where xxxx is port address]
    - -m cclcldp is a reserved load module name that indicates to the CCL that the engine is being started without an NCP, and the load/dump threads should be started to monitor for a Write IPL.

- **For a load operation, once the load is completed, the load/dump threads will be terminated, and the CCL engine will be restarted with the newly loaded NCP.**

- **For a dump operation, once the dump operation is complete, the CCL will be placed back into a 'Monitor for WIPL' state, to await a reload of the CCL.**

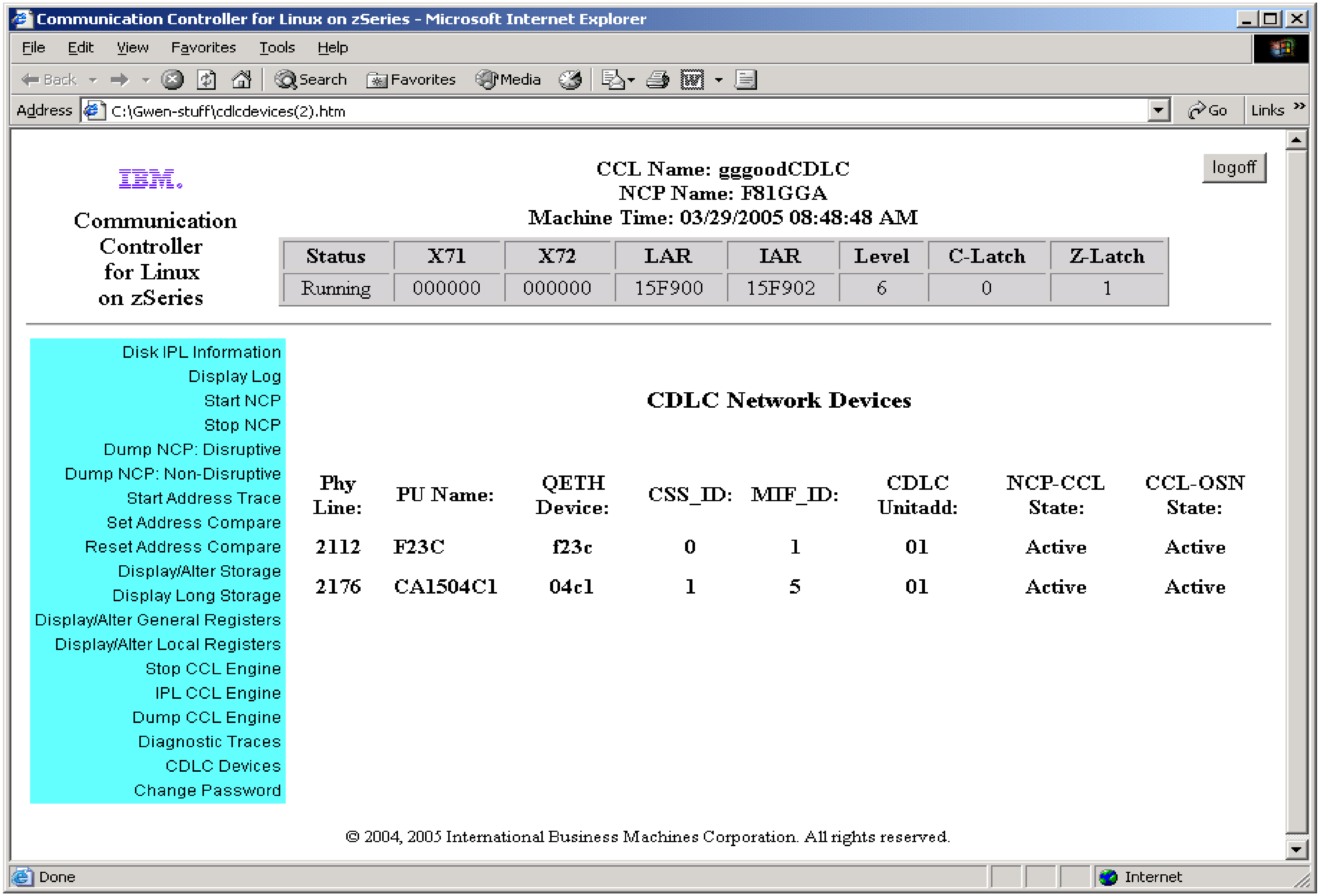
- **The CCL will also be placed back into a 'Monitor for WIPL' state if the load or dump operations fails.**

- **CCL CDLC load supports the following:**
  - Load the NCP, no save to disk
  - Load an NCP that is already on the disk, but the load/dump control byte indicates 'no save to disk'.
  - Load the NCP from disk
  - Load the NCP, save to the disk.

- **Note, that loading an NCP with 'no save to disk', still requires sufficient disk space to temporarily save the NCP loadmodule being loaded. The loadmodule will not be permanently saved to the disk. If enough disk space does not exist to save the temporary load module, then the load operation will fail.**

© 2006 IBM Corporation

MOSS console interface to CDLC network devices


Communication Controller for Linux on zSeries - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address C:\Gwen-stuff\cdlcdevices(2).htm
Go Links
logoff
Done
Internet

IBM.

Communication Controller for Linux on zSeries

CCL Name: gggoodCDLC
NCP Name: F81GGA
Machine Time: 03/29/2005 08:48:48 AM

Status	X71	X72	LAR	IAR	Level	C-Latch	Z-Latch
Running	000000	000000	15F900	15F902	6	0	1

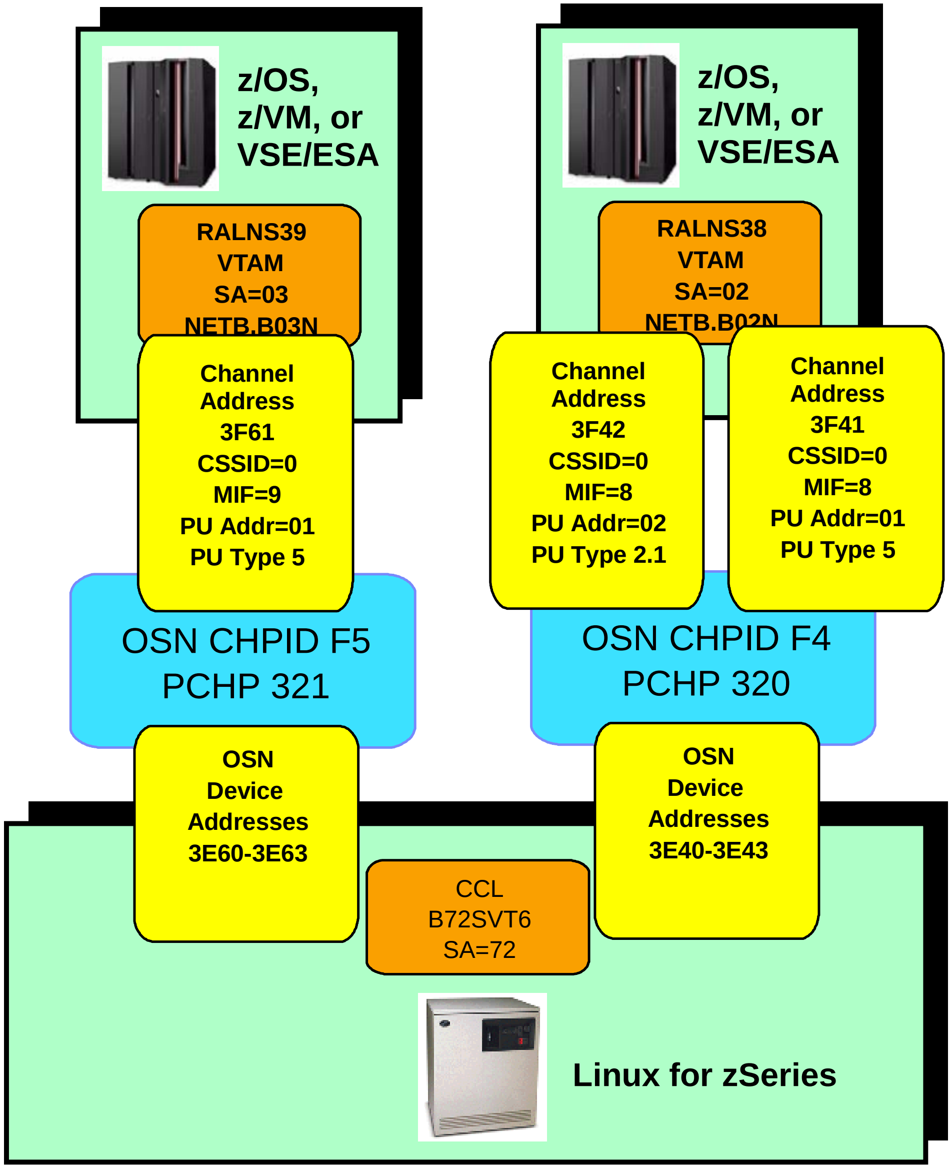
- Disk IPL Information
- Display Log
- Start NCP
- Stop NCP
- Dump NCP: Disruptive
- Dump NCP: Non-Disruptive
- Start Address Trace
- Set Address Compare
- Reset Address Compare
- Display/Alter Storage
- Display Long Storage
- Display/Alter General Registers
- Display/Alter Local Registers
- Stop CCL Engine
- IPL CCL Engine
- Dump CCL Engine
- Diagnostic Traces
- CDLC Devices
- Change Password

CDLC Network Devices

Phy Line:	PU Name:	QETH Device:	CSS_ID:	MIF_ID:	CDLC Unitadd:	NCP-CCL State:	CCL-OSN State:
2112	F23C	f23c	0	1	01	Active	Active
2176	CA1504C1	04c1	1	5	01	Active	Active

© 2004, 2005 International Business Machines Corporation. All rights reserved.

CDLC channel connectivity to CCL: OSN configuration scenario



z/OS, z/VM, or VSE/ESA

RALNS39
VTAM
SA=03
NETB.B03N

Channel Address
3F61
CSSID=0
MIF=9
PU Addr=01
PU Type 5

OSN CHPID F5
PCHP 321

OSN Device Addresses
3E60-3E63

z/OS, z/VM, or VSE/ESA

RALNS38
VTAM
SA=02
NETB.B02N

Channel Address
3F42
CSSID=0
MIF=8
PU Addr=02
PU Type 2.1

Channel Address
3F41
CSSID=0
MIF=8
PU Addr=01
PU Type 5

OSN CHPID F4
PCHP 320

OSN Device Addresses
3E40-3E43

CCL
B72SVT6
SA=72

Linux for zSeries

© 2006 IBM Corporation

IOCP definitions for OSN

```
RESOURCE PART=((CSS(0),(RANS38,8),(RANS39,9),(RALNS27,D)))
*
CHPID PCHID=320,PATH=(CSS(0,1),F4),TYPE=OSN,SHARED
CHPID PCHID=321,PATH=(CSS(0,1),F5),TYPE=OSN,SHARED
*
CNTLUNIT CUNUMBR=3E40,PATH=((CSS(0),F4),(CSS(1),F4)),UNIT=OSN
CNTLUNIT CUNUMBR=3E60,PATH=((CSS(0),F5),(CSS(1),F5)),UNIT=OSN
*
IODEVICE  ADDRESS=(3E40,32),CUNUMBR=3E40,UNIT=OSN,            *
      UNITADD=20
IODEVICE  ADDRESS=(3F41,10),CUNUMBR=3E40,UNIT=3745,           *
      UNITADD=01
IODEVICE  ADDRESS=(3F4E,1),CUNUMBR=3E40,UNIT=OSAD,            *
      UNITADD=FE
*
IODEVICE  ADDRESS=(3E60,32),CUNUMBR=3E60,UNIT=OSN,            *
      UNITADD=20
IODEVICE  ADDRESS=(3F61,10),CUNUMBR=3E60,UNIT=3745,           *
      UNITADD=01
IODEVICE  ADDRESS=(3F6E,1),CUNUMBR=3E60,UNIT=OSAD,            *
      UNITADD=FE
```

OSN CHPIDs

OSN Linux devices

IBM 3745
VTAM/TPF devices

© 2006 IBM Corporation

Defining OSN devices to Linux

- **Create file hwcfg-qeth-bus-ccw-0.0.3e60 in /etc/sysconfig/hardware**

```
#-------------------------------------------------------
# hwcfg-qeth-bus-ccw-0.0.3e60
#
# Hardware configuration for a qeth device at 0.0.3e60
# Automatically generated by netsetup
#--------------------------------------------------------

STARTMODE="auto"
MODULE="qeth"
MODULE_OPTIONS=""
MODULE_UNLOAD="yes"

# Scripts to be called for the various events.
SCRIPTUP="hwup-ccw"
SCRIPTUP_ccw="hwup-ccw"
SCRIPTUP_ccwgroup="hwup-qeth"
SCRIPTDOWN="hwdown-ccw"
# CCW_CHAN_IDS sets the channel IDs for this device
# The first ID will be used as the group ID
CCW_CHAN_IDS="0.0.3e60 0.0.3e61 0.0.3e62"

# CCW_CHAN_NUM set the number of channels for this device
# Always 3 for an qeth device
CCW_CHAN_NUM=3

# CCW_CHAN_MODE sets the port name for an OSA-Express device
CCW_CHAN_MODE="GIGE3E60“
```

Similar configurations will need to be made for other OSN devices - such as in this sample setup: 3e40

Defining OSN devices to Linux (continued)

- **Create file ifcfg-qeth-bus-ccw-0.0.3e60 in /etc/sysconfig/network**

```
BOOTPROTO="static"
UNIQUE=""
STARTMODE="onboot“
```

- **The previous definition will allow the OSN device to become active at startup.**

- **In order for the new OSN device to be recognized at startup, you must modify the /etc/sysconfig/hardware/scripts/hwup-ccw as follows:**
  - Find Line: 1731/01|1731/05
  - Change to: 1731/01|1731/05|1731/06

- **There is an alternative way of defining the OSN devices to Linux by echo'ing information into /sys/bus/ccwgroup/drivers/qeth/group**
  - echo 0.0.3e60,0.0.3e61,0.0.3e62 > /sys/bus/ccwgroup/drivers/qeth/group
  - echo 1 > /sys/bus/ccwgroup/drivers/qeth/0.0.3E60/online
  - ifconfig -e
  - ifconfig osn0 up

- **If using this method, one needs to repeat those echo commands after each IPL of Linux**

© 2006 IBM Corporation

Establishing controls for load/dump of an NCP over the CDLC channel

- **You can use the CCL load/dump program (cclcldp) to load a specified NCP into the CCL engine using a CDLC connection.**

- **You must define an iplportdefs configuration file for each CCL Engine that will be loaded from VTAM over a CDLC connection.**
  - File name is iplportdefs and it must reside in the CCL engine directory

```
IPLPORTDEFS
*
*-----------------------------------------------------------------
*        NCP DEFINITION:   ADDRESS=2112, HOSTLINK=9, ADDR=1
*        OSN DEFINTION:    CSS_ID=X'0' MIF_ID=X'09' UNITADD=X'01'
*                          CCID=X'00090001'
*                          DEVICE=X'3E60'
*-----------------------------------------------------------------
*
      ADDRESS    2112
      HOSTLINK   9
      ADDR       01
      DEVICE     3e60
```

Optionally customize how to correlate NCP definitions with OSA CDLC resources

- **The ESCON logical PU definitions in the NCP gen must be mapped to the OSA CDLC resources so that when a given NCP PU is activated, that PU connects the NCP to the desired VTAM (or TPF) link station (the specific 3745 device that is defined to the SNA host).**

- **The CDLC support in CCL is designed so you can encode this information in the NCP gen. During ESCON resource activation, NCP passes this information to CCL, which passes it to the (correct) OSA, which establishes the CDLC connection. The values are encoded as follows:**
  - The physical LINE ADDRESS value maps to SNA host’s CSS ID
  - The logical LINE HOSTLINK value maps to SNA host’s MIF ID
  - The logical PU ADDR value maps to UNITADD value of 3745 device
  - The last four characters of logical PU name identifies the QETH device

- **If this mapping doesn't match, you need to create a CCLDEFS file in which you code the correlation between the NCP definitions and the OSA CDLC resources.**
  - File name is ncp_load_module_name.CCLDEFS and it must reside in the CCL engine directory

© 2006 IBM Corporation

CCL NCP definitions

```
***********************************************************************
*              CCL CDLC PHYSICAL LINE 2112                           *
***********************************************************************
B72GRP   GROUP LNCTL=CA,ANS=CONT
B72C2112 LINE  ADDRESS=2112,ANS=CONT,SRT=(32765,32765),                *
               XMONLNK=YES,SPEED=18000000
B72P2112  PU   PUTYPE=1
*
************************************************************
*   Logical Group for Physical Line 2112                   *
************************************************************
B72CALG1 GROUP LNCTL=CA,PHYSRSC=B72P2112,MAXPU=32,NPACOLL=NO,ANS=CONT, *
               TIMEOUT=180,DELAY=0.0,CASDL=10,SRT=(32765,32765)
*
***********************************************************************
* CONNECTION TO RALNS39 --- CSS ID = 0: MIF = 9: PUADDR=01
***********************************************************************
B72LL03  LINE  ADDRESS=NONE,HOSTLINK=09,SPEED=18000000,MONLINK=YES
C3P13E60 PU    PUTYPE=5,ADDR=01,TRANSFR=140,TGN=1,MONLINK=YES
*
***********************************************************************
* CONNECTION TO RALNS38 --- CSS ID = 0: MIF = 8: PUADDR=01
***********************************************************************
B72LL02  LINE  ADDRESS=NONE,HOSTLINK=08,SPEED=18000000,MONLINK=YES
C2P13E40 PU    PUTYPE=5,ADDR=01,TRANSFR=140,TGN=1,MONLINK=YES
C2P23E40 PU    PUTYPE=2,ADDR=02
```

- **TYPEGEN=NCP instead of TYPEGEN=NCP-R in the BUILD macro**

- **The VERSION keyword on the BUILD macro must have the "F" extension for ODLC lines**

The last 4 characters of the PU Name (3E60) equates to the READ device address of the OSN CHPID defined for the Linux host

The last 4 characters of the PU Name (3E40) equates to the READ device address of the OSN CHPID defined for the Linux host

© 2006 IBM Corporation

Subarea 03 VTAM definitions

- **Create a channel-attached major node**

```
B03CA    VBUILD    TYPE=CA
B03GRP   GROUP     LNCTL=NCP
*
****************************************************************
*   C3P13E60 PU ADDR = 01: CSS ID = 0: MIF = 9:           ******
****************************************************************
*
B03CALN  LINE      ADDRESS=3F61,MAXBFRU=36
B03PU    PU        CHANCON=COND,MAXDATA=32768,TGN=1
```

© 2006 IBM Corporation

Subarea 02 VTAM definitions

- **Create a channel-attached major node**

```
B02CA    VBUILD    TYPE=CA
B02GRP   GROUP     LNCTL=NCP
*
****************************************************************
*   C2P13E40 PU ADDR = 01: CSS ID = 0: MIF = 8:           ******
****************************************************************
*
B02CALN  LINE      ADDRESS=3F41,MAXBFRU=36
B02PU    PU        CHANCON=COND,MAXDATA=32768,TGN=1
```

- **Create a local major node (for APPN activation)**

```
B02LCL   VBUILD TYPE=LOCAL
*
****************************************************************
*   C2P23E40 PU ADDR = 02: CSS ID = 0: MIF = 8:           ******
****************************************************************
*
B02LCLP1 PU    PUTYPE=2,CUADDR=3F42,ISTATUS=ACTIVE,XID=YES,          *
               VPACING=0,SSCPFM=USSSCS,MAXBFRU=255,DYNLU=YES,        *
               CONNTYPE=APPN,CPCP=YES
```

© 2006 IBM Corporation

Activating the NCP over the CDLC channel

- **Start the CCL engine with the NCP load module already loaded (in this example: B72SVT6):**

```
nohup ./cclengine –mB72SVT6 -p2072 SVTB72 &
```

- **From NETB.B03N, activate the local CA major node**

```
V NET,ACT,ID=B03CA,ALL
IST097I VARY ACCEPTED
IST093I B03CA ACTIVE
IST464I LINK STATION B03PU HAS CONTACTED B72SVT6 SA 72
IST093I B03PU ACTIVE
```

- **From NETB.B03N, activate the NCP major node**

```
V NET,ACT,ID=B72SVT6,ALL
IST097I VARY ACCEPTED
IST093I B72SVT6 ACTIVE
IST093I B72NPPU ACTIVE
IST093I B72P2112 ACTIVE
IST464I LINK STATION C2P13E40 HAS CONTACTED B02N SA 2
IST093I C2P13E40 ACTIVE
IST464I LINK STATION C3P13E60 HAS CONTACTED ISTPUS SA 3
IST093I C3P13E60 ACTIVE
```

Activating the NCP over the CDLC channel (continued)

- **To contact the NCP from VTAM subarea 02 (NETB.B02N) via the channel major node, activate the local CA major node**

```
V NET,ACT,ID=B02CA,ALL
IST097I VARY ACCEPTED
IST093I B02CA ACTIVE
IST464I LINK STATION B02PU HAS CONTACTED B72SVT6 SA 72
IST093I B02PU ACTIVE
```

- **To contact the NCP from VTAM subarea 02 (NETB.B02N) via the local major node (as an APPN node), activate the Local major node**

```
V NET,ACT,ID=B02LCL,ALL
IST097I VARY ACCEPTED
IST093I B02LCL ACTIVE
IST1086I APPN CONNECTION FOR NETB.B03N IS ACTIVE - TGN = 21
IST093I B02LCLP1 ACTIVE
IST1096I CP-CP SESSIONS WITH NETB.B03N ACTIVATED
```

Activating and loading the NCP over the CDLC channel

- **Start the CCL engine - but use a load module name of cclcldp to instruct the CCL engine that the load will come from the VTAM activation command:**

```
nohup ./cclengine –mcclcldp -p2072 SVTB72 &
```

- **From NETB.B03N, load and activate the NCP Major Node**

```
V NET,ACT,ID=B72SVT6,ALL,LOAD=YES,U=3F61
IST097I VARY ACCEPTED
IST461I ACTIVATE FOR U/RNAME ENTRY ID = 3F61-S STARTED
IST897I LOAD OF B72SVT6 STARTED
IST270I LOAD OF B72SVT6 COMPLETE - LOAD MODULE = B72SVT6
IST464I LINK STATION 3F61-S HAS CONTACTED B72SVT6 SA 72
IST093I B72SVT6 ACTIVE IST093I B72NPPU ACTIVE
IST093I B72P2112 ACTIVE
IST464I LINK STATION C2P13E40 HAS CONTACTED B02N SA 2
IST093I C2P13E40 ACTIVE
IST464I LINK STATION C3P13E60 HAS CONTACTED ISTPUS SA 3
IST093I C3P13E60 ACTIVE
```

© 2006 IBM Corporation

Appendix B:
QDIO Layer 2 access

© 2006 IBM Corporation

QDIO Layer 2 - introduction

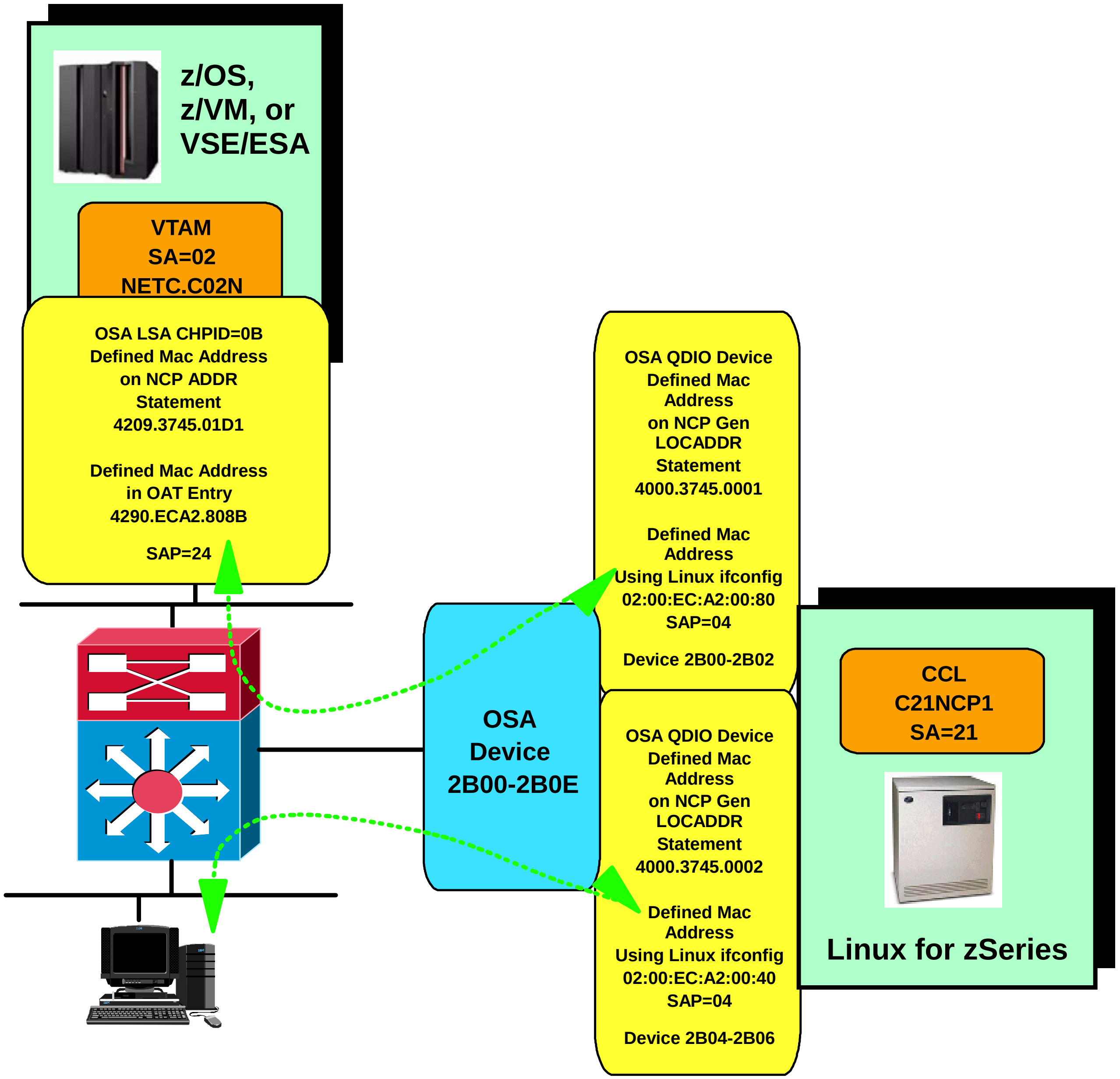
- **There are no IOCP definitions to indicate layer 2 mode.**
  - QDIO Layer 2 or QDIO layer 3 mode is a QDIO device driver option
  - Currently, only the Linux QDIO device driver and z/VM's VSWITCH support layer 2 mode

- **Each Linux ETHn interface requires a triplet of OSD device numbers**
  - For each ETHn interface in layer 2 mode, Linux can set the MAC address either dynamically through an ifconfig command, or statically via options in configuration files

- **QDIO layer 2 scenarios:**
  - Using QDIO layer-2 in a native LPAR
    - Each Linux ETHn interface requires 3 OSD device numbers
    - One Virtual MAC per ETHn interface
  - Using QDIO layer-2 under z/VM without the VSWITCH
    - Each Linux ETHn interface requires 3 OSD device numbers
    - One Virtual MAC per ETHn interface
    - Each set of triple OSD device numbers attached to one Linux guest
  - Using QDIO layer-2 under z/VM with the VSWITCH
    - Each Linux ETHn interface requires 3 OSD device numbers
    - One Virtual MAC per ETHn interface
    - The VSWITCH requires a triplet of IOCP-defined OSD device numbers
    - Each Linux triplet of OSD device numbers are virtual (defined via NICDEF statements)

© 2006 IBM Corporation

QDIO layer 2 mode: configuration overview - native LPAR



z/OS, z/VM, or VSE/ESA

VTAM
SA=02
NETC.C02N

OSA LSA CHPID=0B
Defined Mac Address on NCP ADDR Statement
4209.3745.01D1

Defined Mac Address in OAT Entry
4290.ECA2.808B

SAP=24

OSA Device
2B00-2B0E

OSA QDIO Device
Defined Mac Address on NCP Gen LOCADDR Statement
4000.3745.0001

Defined Mac Address Using Linux ifconfig
02:00:EC:A2:00:80
SAP=04

Device 2B00-2B02

OSA QDIO Device
Defined Mac Address on NCP Gen LOCADDR Statement
4000.3745.0002

Defined Mac Address Using Linux ifconfig
02:00:EC:A2:00:40
SAP=04

Device 2B04-2B06

CCL
C21NCP1
SA=21

Linux for zSeries

Define QDIO layer 2 devices 2B00-2B02 to Linux - hardware

- **Create the script hwcfg-qeth-bus-ccw-0.0.2b00 in the /etc/sysconfig/hardware directory**

```
#!/bin/sh
#
STARTMODE='auto'
MODULE='qeth'
MODULE_OPTIONS=''
MODULE_UNLOAD='yes'

SCRIPTUP='hwup-ccw'
SCRIPTUP_ccw='hwup-ccw'
SCRIPTUP_ccwgroup='hwup-qeth'
SCRIPTDOWN='hwdown-ccw'

# CCW_CHAN_IDS are the device addresses
CCW_CHAN_IDS='0.0.2b00 0.0.2b01 0.0.2b02'

# CCW_CHAN_NUM set the number of channels for this device
CCW_CHAN_NUM='3'

# CCW_CHAN_MODE sets the port name for an OSA-Express device
CCW_CHAN_MODE='GIGE2B00'

# QETH_LAYER2_SUPPORT enables Layer2 support for this device.
QETH_LAYER2_SUPPORT=1
```

QDIO device addresses 2B00, 2B01, and 2B02

This is where you specify to the Linux QDIO device driver that you want to operate this QDIO port in layer 2 mode.

Define QDIO layer 2 devices 2B00-2B02 to Linux - network

- **Create this script ifcfg-qeth-bus-ccw-0.0.2b00 in the /etc/sysconfig/network directory**

```
LLADDR='02:00:ec:a2:00:80‘
BOOTPROTO='none'
STARTMODE='onboot'
UNIQUE='‘
```

- **By using LLADDR, we can set the MAC address to any value necessary. This keyword may be different in Red Hat releases.**

- **The local MAC address can also be set using an ifconfig command:**

```
ifconfig eth0 hw ether 02:00:ec:a2:00:80 up
```

- **MAC Address defined on the NCP LOCADDR statement is the non-canonical version of this address – 4000.3745.0001**

© 2006 IBM Corporation

Define QDIO layer 2 devices 2B04-2B06 to Linux - hardware

- **Create the script hwcfg-qeth-bus-ccw-0.0.2b04 in the /etc/sysconfig/hardware directory**

```
#!/bin/sh
#
STARTMODE='auto'
MODULE='qeth'
MODULE_OPTIONS=''
MODULE_UNLOAD='yes'

SCRIPTUP='hwup-ccw'
SCRIPTUP_ccw='hwup-ccw'
SCRIPTUP_ccwgroup='hwup-qeth'
SCRIPTDOWN='hwdown-ccw'

# CCW_CHAN_IDS are the device addresses
CCW_CHAN_IDS='0.0.2b04 0.0.2b05 0.0.2b06'

# CCW_CHAN_NUM set the number of channels for this device
CCW_CHAN_NUM='3'

# CCW_CHAN_MODE sets the port name for an OSA-Express device
CCW_CHAN_MODE='GIGE2B00'

# QETH_LAYER2_SUPPORT enables Layer2 support for this device.
QETH_LAYER2_SUPPORT=1
```

QDIO device addresses 2B04, 2B05, and 2B06

This is where you specify to the Linux QDIO device driver that you want to operate this QDIO port in layer 2 mode.

© 2006 IBM Corporation

Define QDIO layer 2 devices 2B04-2B06 to Linux - network

- **Create this script ifcfg-qeth-bus-ccw-0.0.2b04 in the /etc/sysconfig/network directory**

```
LLADDR='02:00:ec:a2:00:40‘
BOOTPROTO='none'
STARTMODE='onboot'
UNIQUE='‘
```

- **By using LLADDR, we can set the MAC address to any value necessary. This keyword may be different in Red Hat releases.**

- **The local MAC address can also be set using an ifconfig command:**

  ```
  ifconfig eth0 hw ether 02:00:ec:a2:00:40 up
  ```

- **MAC Address defined on the NCP LOCADDR statement is the non-canonical version of this address – 4000.3745.0002**

© 2006 IBM Corporation

VTAM channel-attached major node definitions

- **Create a channel-attached major node in VTAM on subarea 02:**

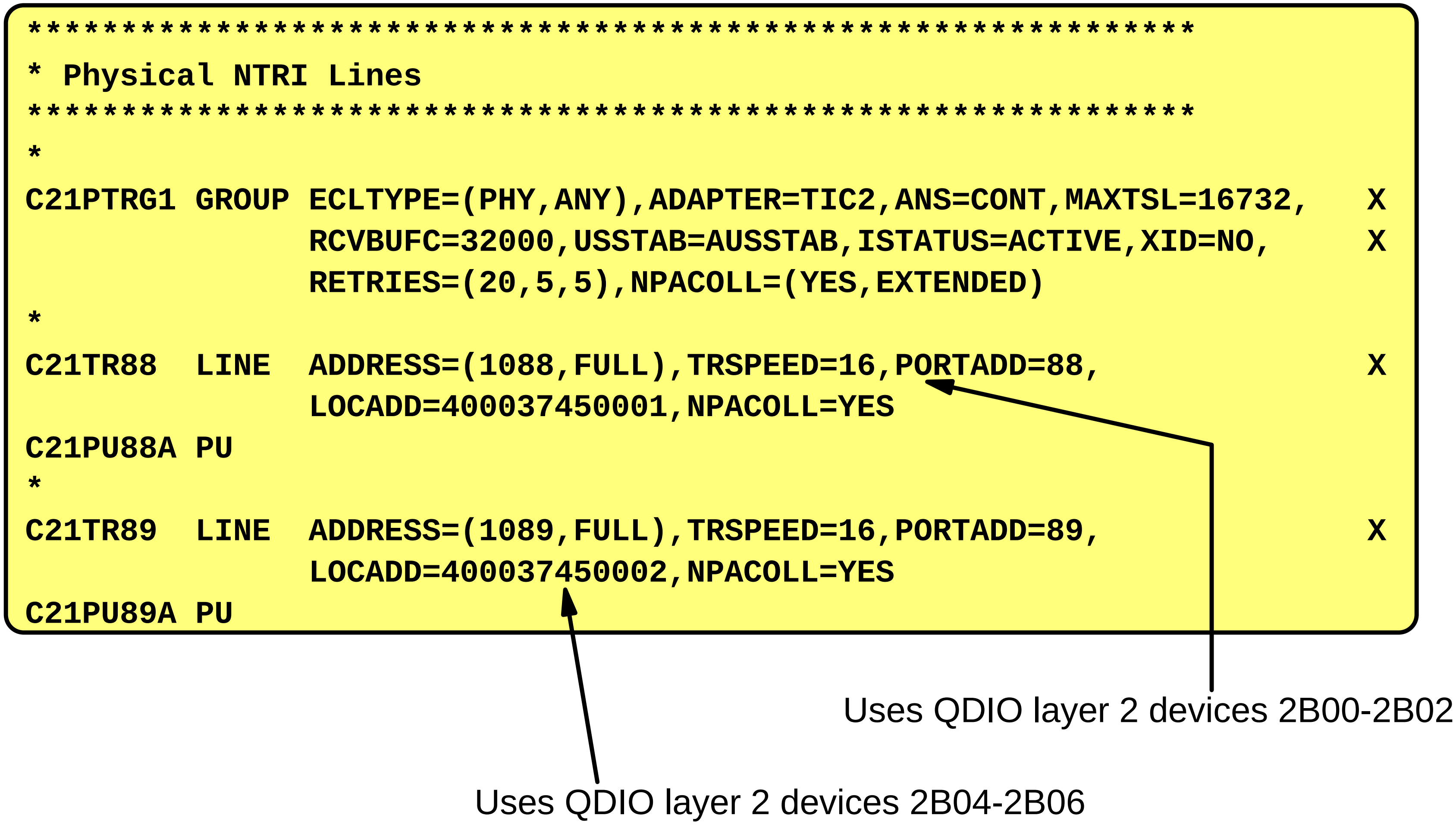
```
C02XCA   VBUILD  TYPE=XCA
*
C02ETHPT PORT   MEDIUM=CSMACD,ADAPNO=0,SAPADDR=24,CUADDR=2EBA,          X
                TIMER=100
C02ETHGP GROUP DIAL=NO,ISTATUS=ACTIVE
*
C02ETHL2 LINE   USER=SNA,ISTATUS=ACTIVE
C02ETHP2 PU     MACADDR=0200ECA20080,PUTYPE=5,SUBAREA=21,TGN=1,         X
                SAPADDR=04,ALLOWACT=YES
```

- **In this sample setup, VTAM communicates with the NCP over QDIO devices 2B00-2B02 - using the default SAP of 04.**

© 2006 IBM Corporation

NCP physical line interface definitions (NTRI)

- **Define the NCP physical line interfaces:**



```
***************************************************************
* Physical NTRI Lines
***************************************************************
*
C21PTRG1 GROUP ECLTYPE=(PHY,ANY),ADAPTER=TIC2,ANS=CONT,MAXTSL=16732,   X
               RCVBUFC=32000,USSTAB=AUSSTAB,ISTATUS=ACTIVE,XID=NO,     X
               RETRIES=(20,5,5),NPACOLL=(YES,EXTENDED)
*
C21TR88  LINE  ADDRESS=(1088,FULL),TRSPEED=16,PORTADD=88,              X
               LOCADD=400037450001,NPACOLL=YES
C21PU88A PU
*
C21TR89  LINE  ADDRESS=(1089,FULL),TRSPEED=16,PORTADD=89,              X
               LOCADD=400037450002,NPACOLL=YES
C21PU89A PU
```

Uses QDIO layer 2 devices 2B00-2B02

Uses QDIO layer 2 devices 2B04-2B06

© 2006 IBM Corporation

NCP logical line interface definitions

- **Define the NCP logical line interfaces:**

```
***********************************************************************
* LOGICAL BNN Lines                                                   *
***********************************************************************
*
C21BNNG1 GROUP ECLTYPE=LOGICAL,ANS=CONTINUE,AUTOGEN=250,CALL=INOUT,    X
               ISTATUS=ACTIVE,PHYSRSC=C21PU89A,                        X
               USSTAB=AUSSTAB,RETRIES=(10,10,10,20),XMITDLY=NONE,      X
               MODETAB=AMODETAB,NPACOLL=YES

***********************************************************************
*     NTRI INN LOGICAL LINES FOR TOKEN RING PORT 1088                 *
***********************************************************************
*
C21INNG1 GROUP ECLTYPE=(LOGICAL,SUBAREA),ANS=CONT,PHYSRSC=C21PU88A,    X
               LOCALTO=13.5,REMOTTO=18.2,T2TIMER=(0.2,0.2,3),         X
               ISTATUS=ACTIVE,SDLCST=(C21PRI,C21SEC),NPACOLL=YES,      X
               MONLINK=CONT
*
C21LG2A  LINE  TGN=1,TGCONF=SINGLE
C21PG2A  PU    ADDR=184209374501D1,SSAP=(04,H)
```

- **You start the CCL engine and loads the NCP using the usual cclengine command:**

```
nohup ./cclengine –mC21NCP1 -p2021 SVTC21 &
```

Activating the NCP from VTAM

- **From NETC.C02N activate the XCA major node**

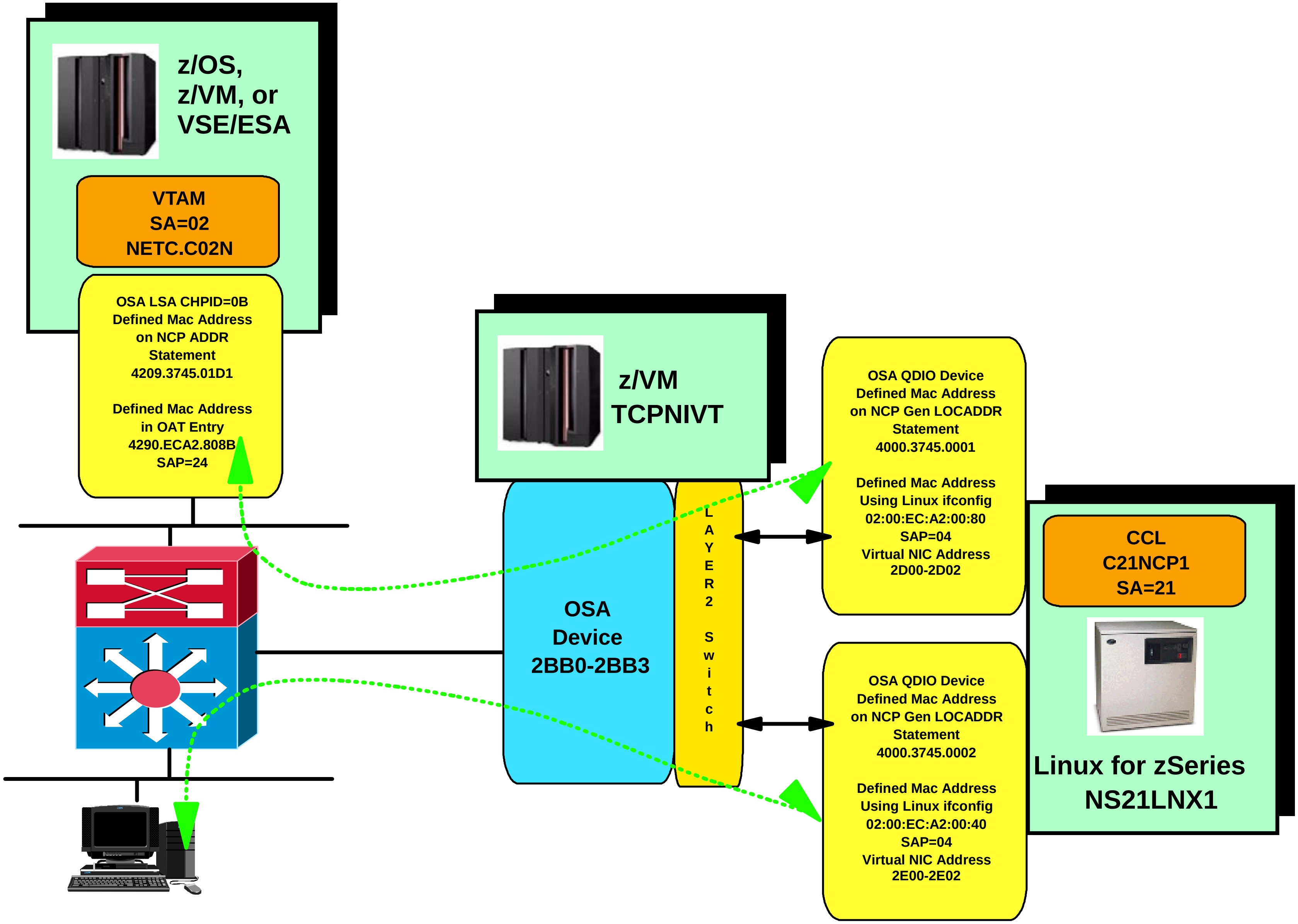
```
V NET,ACT,ALL,ID=C02XCA
IST097I VARY ACCEPTED
IST093I C02XCA ACTIVE
IST464I LINK STATION C02ETHP2 HAS CONTACTED C21NCP1 SA 21
IST093I C02ETHP2 ACTIVE
```

- **From NETC.C02N activate the NCP**

```
V NET,ACT,ID=C21NCP1,ALL
IST097I VARY ACCEPTED
IST093I C21NCP1 ACTIVE
IST093I C21PU88A ACTIVE
IST093I C21PU89A ACTIVE
IST093I C21NPPU ACTIVE
IST464I LINK STATION C21PG2A HAS CONTACTED C02NPU SA 2
IST093I C21PG2A ACTIVE
```

© 2006 IBM Corporation

QDIO layer 2 mode: configuration overview - z/VM with VSWITCH



z/OS, z/VM, or VSE/ESA

VTAM
SA=02
NETC.C02N

OSA LSA CHPID=0B
Defined Mac Address
on NCP ADDR
Statement
4209.3745.01D1

Defined Mac Address
in OAT Entry
4290.ECA2.808B
SAP=24

z/VM
TCPNIVT

OSA
Device
2BB0-2BB3

LAYER 2 Switch

OSA QDIO Device
Defined Mac Address
on NCP Gen LOCADDR
Statement
4000.3745.0001

Defined Mac Address
Using Linux ifconfig
02:00:EC:A2:00:80
SAP=04
Virtual NIC Address
2D00-2D02

OSA QDIO Device
Defined Mac Address
on NCP Gen LOCADDR
Statement
4000.3745.0002

Defined Mac Address
Using Linux ifconfig
02:00:EC:A2:00:40
SAP=04
Virtual NIC Address
2E00-2E02

CCL
C21NCP1
SA=21

Linux for zSeries
NS21LNX1

Setting up the VSWITCH

- **Create the userid TCPNIVT and define the IUCV in the z/VM user directory.**

```
USER TCPNIVT TCPNIVT 128M 128M ABCG
INCLUDE TCPCMSU
OPTION QUICKDSP SVMSTAT MAXCONN 1024 DIAG98 APPLMON
SHARE RELATIVE 3000
IUCV ALLOW
IUCV ANY PRIORITY
IUCV *CCS PRIORITY MSGLIMIT 255
IUCV *VSWITCH MSGLIMIT 65535
LINK TCPMAINT 591 591 RR
LINK TCPMAINT 592 592 RR
LINK TCPANIVT 198 198 RR

MDISK 191 3390 1307 005 510W02
```

- **The IUCV *VSWITCH is required on VM IDs that are to be considered switch controllers.**

- **UpdateTCP/IP statements on z/VM system to define control of a Virtual Switch. This is done on the TCPNIVT ID.**

```
;----------------------------------------------------------------
; Define whether or not a stack is available to control a
; CP-defined Virtual Switch's connection to a real LAN segment
; through an OSA Express device.  The range of virtual addresses
; that are to be used for such a connection can optionally be
; specified with the VSWITCH statement.
;-----------------------------------------------------------------
OBEY
   OPERATOR MAINT
   OPERATOR TCPNIVT
ENDOBEY

VSWITCH CONTROLLER ON
```

Setting up the VSWITCH

- **Define the Virtual Switch to the TCPNIVT ID. These definitions are extracted from the PROFILE EXEC.**

```
/**************************/
/* DETACH VIRTUAL SWITCH  */
/**************************/

DETACH VSWITCH VSWLAY2

/**************************/
/* DEFINE VIRTUAL SWITCH  */
/**************************/

DEFINE VSWITCH VSWLAY2 RDEV 2B00 ETH CON CONTR TCPNIVT PORT LAY2PORT

/***********************************************************/
/*GRANT AUTHORITY FOR IDS TO COUPLE TO THE VIRTUAL SWITCH  */
/***********************************************************/


SET VSWITCH VSWLAY2 GRANT NS21LNX1
```

- **Keyword explanations**
  - DEFINE VSWITCH VSWLAY2
    - defines a Virtual Switch named VSWLAY2
  - RDEV 2B00 ETH
    - uses real OSA device 2B00 and type Ethernet
  - CONTR TCPNIVT
    - z/VM userid defined as the controller for the Virtual Switch. In my network, the ID is called TCPNIVT
  - PORTNAME LAY2PORT
    - unique name that identifies the OSA Express adapter. Up to 3 names can be defined for a single OSA.
  - The SET command authorizes userid NS21LNX1 to access the Virtual Switch
  - At this point, the virtual switch setup is complete

Setting up the Linux guest IDs to z/VM

- **Define the NICDEFs to the Linux guest in the z/VM User Directory.**

```
USER NS21LNX1 NS21LNX1  512M 1024M BG 64
INCLUDE IBMDFLT
CPU 0 NODEDICATE
CPU 1 NODEDICATE
MACHINE ESA 4
OPTION QUICKDSP
DEDICATE 2EA2 2EA2
...........
DEDICATE 2EB3 2EB3
LINK  MAINT  19B 19B RR
MDISK 191 3390 1606 005 510W02 MR
NICDEF 2D00 TYPE QDIO
NICDEF 2E00 TYPE QDIO
```

- **The following example is the PROFILE EXEC from Linux guest NS21LNX1. The NICDEF statements in the user directory attach devices 2D00 and 2E00 to this guest when the guest is logged on.**

- **The virtual NIC is then coupled to the Virtual Switch.**

```
'CP TERM MORE 0 0'
SET PF12 RETRIEVE
SET PF24 RETRIEVE
COUPLE 2D00 TO SYSTEM VSWLAY2
COUPLE 2E00 TO SYSTEM VSWLAY2
```

- **When the Linux guest is IPL’d, two new QDIO OSA adapters are available to the system (2D00 and 2E00).**

© 2006 IBM Corporation

Define QDIO layer 2 devices 2D00-2D02 to Linux - hardware

- **Create the script hwcfg-qeth-bus-ccw-0.0.2d00 in the /etc/sysconfig/hardware directory**

```
#!/bin/sh
#
STARTMODE='auto'
MODULE='qeth'
MODULE_OPTIONS=''
MODULE_UNLOAD='yes'

SCRIPTUP='hwup-ccw'
SCRIPTUP_ccw='hwup-ccw'
SCRIPTUP_ccwgroup='hwup-qeth'
SCRIPTDOWN='hwdown-ccw'

# CCW_CHAN_IDS are the device addresses
CCW_CHAN_IDS='0.0.2d00 0.0.2d01 0.0.2d02'

# CCW_CHAN_NUM set the number of channels for this device
CCW_CHAN_NUM='3'

# CCW_CHAN_MODE sets the port name for an OSA-Express device
CCW_CHAN_MODE='GIGE2B00'

# QETH_LAYER2_SUPPORT enables Layer2 support for this device.
QETH_LAYER2_SUPPORT=1
```

QDIO device addresses 2D00, 2D01, and 2D02

From here on the setup is similar to the setup for use of QDIO layer 2 mode in native LPARs.

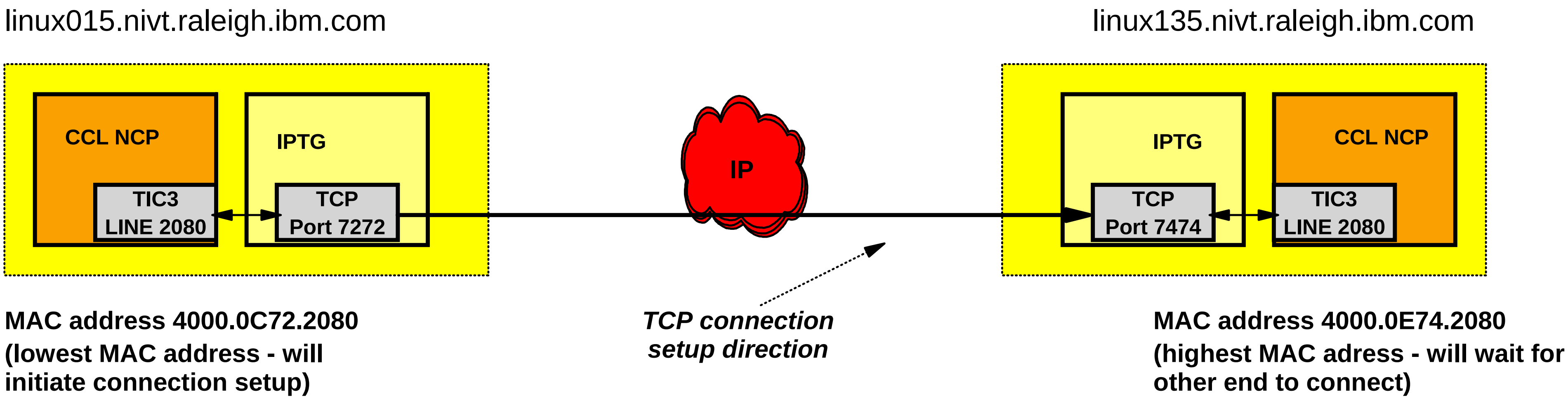
This is where you specify to the Linux QDIO device driver that you want to operate this QDIO port in layer 2 mode.

Appendix C:
IPTG between two SNI NCPs

© 2006 IBM Corporation

IPTG Basics

- **CCL V1R2 IPTG simulates IBM 3746-900 Token-ring processor (TRP) - TIC3**
  - Far fewer NCP cycles per SNA packet than for TIC2
  - Link level packet handling and error recovery totally left up to the simulated adapter
  - Line address 2080 is reserved for IPTG
- **Uses TCP/IP socket to partner CCL instead of SNA LLC2**
  - NCP won't care - thinks the adapter is doing LLC2
  - IP transport avoids SNA LLC2 on the wire
    - will work with LCS and QDIO layer 2, but they're not required
  - Good long-term solution for SNI over IP
    - CCL-to-CCL subarea connections only (not APPN)
      - All connections predefined in the NCP gen on both sides
- **Requires additional coordinated sysdef**
  - Coordinate with local NCP gen and with partner CCL definitions



MAC address 4000.0C72.2080 (lowest MAC address - will initiate connection setup)

TCP connection setup direction

MAC address 4000.0E74.2080 (highest MAC adress - will wait for other end to connect)

© 2006 IBM Corporation

NCP definitions for use with IPTG

- **Define one physical token ring LINE with ADDRESS=2080**

```
GROUP ECLTYPE=(PHY,SUB)
LINE  ADDRESS=2080,LOCADD=mymacaddress
PU    ADDR=01
```

- **Define one logical LINE & PU for each partner node**

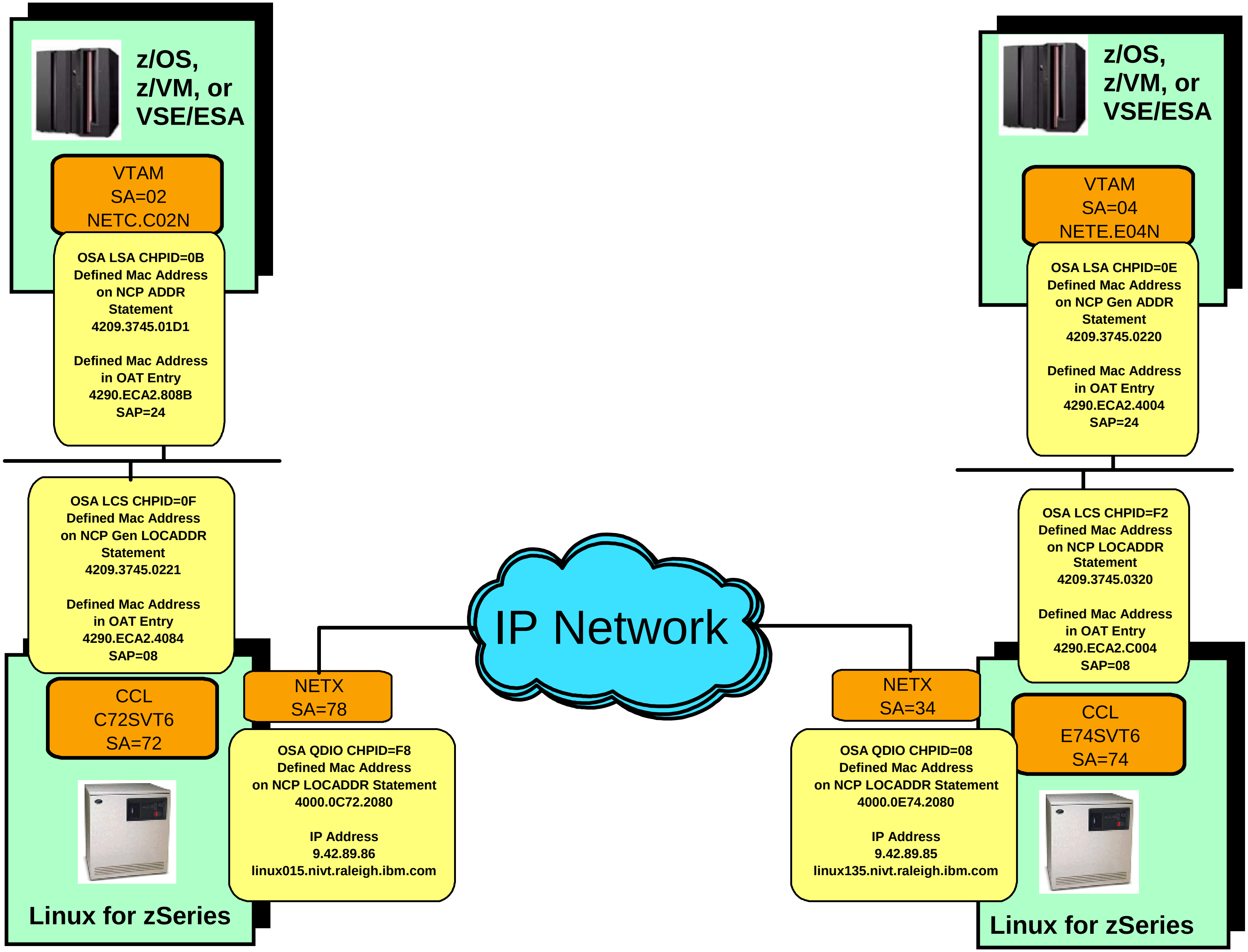
```
GROUP ECLTYPE=(LOG,SUB),PHYSRSC=IPTGPhysicalPU,TGCONF=SINGLE
LINE
PU    ADDR=04partmacaddr1,TGN=1
LINE
PU    ADDR=04partmacaddr2,TGN=1,NETID=NETX
```

- **MAC addresses do not map to real adapter MAC addresses**
  - Can be any value that passes NDH validation
  - Node with lower MAC initiates the TCP connection
  - MACs are exchanged for authentication

The NCP sees the local IPTG process through what it believes is a token-ring interface (a TIC3 port) in an IBM 3746. The TIC3 LINE address for IPTG is predetermined as 2080.

© 2006 IBM Corporation

IPTG between two SNI NCPs - configuration overview



z/OS, z/VM, or VSE/ESA

VTAM
SA=02
NETC.C02N

OSA LSA CHPID=0B
Defined Mac Address on NCP ADDR Statement
4209.3745.01D1

Defined Mac Address in OAT Entry
4290.ECA2.808B
SAP=24

OSA LCS CHPID=0F
Defined Mac Address on NCP Gen LOCADDR Statement
4209.3745.0221

Defined Mac Address in OAT Entry
4290.ECA2.4084
SAP=08

CCL
C72SVT6
SA=72

Linux for zSeries

NETX
SA=78

OSA QDIO CHPID=F8
Defined Mac Address on NCP LOCADDR Statement
4000.0C72.2080

IP Address
9.42.89.86
linux015.nivt.raleigh.ibm.com

IP Network

z/OS, z/VM, or VSE/ESA

VTAM
SA=04
NETE.E04N

OSA LSA CHPID=0E
Defined Mac Address on NCP Gen ADDR Statement
4209.3745.0220

Defined Mac Address in OAT Entry
4290.ECA2.4004
SAP=24

OSA LCS CHPID=F2
Defined Mac Address on NCP LOCADDR Statement
4209.3745.0320

Defined Mac Address in OAT Entry
4290.ECA2.C004
SAP=08

NETX
SA=34

OSA QDIO CHPID=08
Defined Mac Address on NCP LOCADDR Statement
4000.0E74.2080

IP Address
9.42.89.85
linux135.nivt.raleigh.ibm.com

CCL
E74SVT6
SA=74

Linux for zSeries

NCP C72SVT6 - TIC3 physical and logical line definitions

Physical LINE and PU

```
**********************************************************************
* PHYSICAL TOKEN RING INTERFACE FOR TCP/IP CONNECTIONS - TIC3 2080   *
**********************************************************************
*
C72IPPG  GROUP ECLTYPE=(PHY,SUB),ADAPTER=TIC3,ANS=CONT,ISTATUS=ACTIVE, X
               RCVBUFC=32000,MAXTSL=16732,RETRIES=(20,5,5)
*
C72IPPL  LINE  ADDRESS=(2080,FULL),PORTADD=80,                         X
               LOCADD=40000C722080,NPACOLL=NO
C72IPPP  PU    ADDR=01,XMONLNK=YES
```

Logical LINE and PU

```
**********************************************************************
* LOGICAL INN TCP/IP CONNECTIONS                                     *
**********************************************************************
*
C72IPLG  GROUP ECLTYPE=(LOGICAL,SUBAREA),ANS=CONT,ISTATUS=ACTIVE,      X
               PHYSRSC=C72IPPP,SDLCST=(C72PRI,C72SEC),NPACOLL=NO,      X
               T2TIMER=(1.5,2.0,3),LOCALTO=13.5,REMOTTO=18.2
*
*---------------------------------------------------------------------
* Linkstation to E74 – IPTG INN Connection
*---------------------------------------------------------------------
*
C72IPLL5 LINE  TGN=1,TGCONF=SINGLE,MONLINK=YES
C72IPLP5 PU    ADDR=0840000E742080,SSAP=(08,H),NETID=NETX
```

© 2006 IBM Corporation

NCP C72SVT6 - CCLDEFS definitions

- **CCLDEFS file for IPTG to SNI partner**

```
ccldefs
  TCPDEFS
*--------------------------------------------------
* Define Local IPTG Port
*--------------------------------------------------
*
    LOCALNODE
      IPPORT              7272
      IPTOS              LOWDELAY
*
*--------------------------------------------------
* Define Remote IPTG Port
*--------------------------------------------------
*
      REMOTENODE
        IPTOS           LOWDELAY
        PUNAME          C72IPLP5
        HOST            linux135.nivt.raleigh.ibm.com
        IPPORT          7474
     ENDTCPDEFS
endccldefs
```

This is the endpoint with the lowest MAC address, so this is the endpoint that initiates the connection setup towards the other endpoint.

Local TCP port number where TCP connection will come from

Remote hostname (to connect to remote IPTG)

Remote IPTG's TCP port number to which local IPTG connects

- **CCL V1R2.1 added IPADDR keyword to the LOCALNODE definitions to control which local IP address the TCP connection will use**
  - Predefined local IP address, remote IP address, and "server" TCP port number - easier firewall administration

NCP C74SVT6 - TIC3 physical and logical line definitions

Physical LINE and PU

```
**********************************************************************
* PHYSICAL TOKEN RING INTERFACE FOR TCP/IP CONNECTIONS - TIC3 2080   *
**********************************************************************
*
E74IPPG  GROUP ECLTYPE=(PHY,SUB),ADAPTER=TIC3,ANS=CONT,ISTATUS=ACTIVE, X
               RCVBUFC=32000,MAXTSL=16732,RETRIES=(20,5,5)
*
E74IPPL  LINE  ADDRESS=(2080,FULL),PORTADD=80,                         X
               LOCADD=40000E742080,NPACOLL=NO
E74IPPP  PU    ADDR=01,XMONLNK=YES
```

Logical LINE and PU

```
**********************************************************************
* LOGICAL INN TCP/IP CONNECTIONS                                     *
**********************************************************************
*
E74IPLG  GROUP ECLTYPE=(LOGICAL,SUBAREA),ANS=CONT,ISTATUS=ACTIVE,      X
               PHYSRSC=E74IPPP,SDLCST=(E74PRI,E74SEC),NPACOLL=NO,      X
               T2TIMER=(1.5,2.0,3),LOCALTO=13.5,REMOTTO=18.2
*
*---------------------------------------------------------------------
* Linkstation to C72 – IPTG INN Connection
*---------------------------------------------------------------------
*
E74IPLL5 LINE  TGN=1,TGCONF=SINGLE,MONLINK=YES
E74IPLP5 PU    ADDR=0840000C722080,SSAP=(08,H),NETID=NETX
```

© 2006 IBM Corporation

NCP C74SVT6 - CCLDEFS definitions

- **CCLDEFS file for IPTG to SNI partner**

```
ccldefs
  TCPDEFS
*--------------------------------------------------
* Define Local IPTG Port
*--------------------------------------------------
*
      LOCALNODE
        IPPORT     7474
        IPTOS      LOWDELAY
*
*--------------------------------------------------
* Define Remote IPTG Port
*--------------------------------------------------
*
 REMOTENODE
      PUNAME     E74IPLP5      (TR logical PU)
      HOST       linux015.nivt.raleigh.ibm.com
      IPPORT     7272
ENDTCPDEFS
endccldefs
```

This is the endpoint with the highest MAC address, so this is the endpoint that will wait for the other endpoint to connect to it.

Local TCP port number where TCP connection from remote IPTG will be established to

Hostname of remote IPTG

TCP port number of remote IPTG

Activation sequence from VTAM subarea 02

- **Start the CCL engine with the NCP load module already loaded (in this example: C72SVT6):**

```
nohup ./cclengine -mC72SVT6 -p2072 SVTC72 &
```

- **From NETC.C02N activate the XCA major node**

```
V NET,ACT,ID=C02XCA,ALL
IST097I VARY ACCEPTED
IST093I C02XCA ACTIVE
IST464I LINK STATION C02ETHP1 HAS CONTACTED SA 72
IST093I C02ETHP1 ACTIVE
```

- **From NETC.C02N activate the NCP**

```
V NET,ACT,ID=C72SVT6,ALL
IST097I VARY ACCEPTED
IST093I C72SVT6 ACTIVE
IST093I C72PU88A ACTIVE
IST093I C72PU89A ACTIVE
IST093I C72IPPP ACTIVE
IST464I LINK STATION C72PG2A HAS CONTACTED C02NPU SA 2
IST093I C72PG2A ACTIVE
IST720I C72IPLP5 HAS CONTACTED E74SVT6 IN NETX, SA 34
IST093I C72IPLP5 ACTIVE
```

© 2006 IBM Corporation

Activation sequence from VTAM subarea 04

- **Start the CCL engine with the NCP load module already loaded (in this example: E74SVT6):**

```
nohup ./cclengine -mE74SVT6 -p2072 SVTC72 &
```

- **From NETE.E04N activate the XCA major node**

```
V NET,ACT,ID=E04XCA,ALL
IST093I E04XCA ACTIVE
IST464I LINK STATION E04ETHPU HAS CONTACTED E74SVT6 SA 74
IST093I E04ETHPU ACTIVE
```

- **From NETE.E08N activate the NCP**

```
V NET,ACT,ALL,ID=E74SVT6
IST097I VARY ACCEPTED
IST093I E74SVT6 ACTIVE
IST093I E74PU92A ACTIVE
IST093I E74PU93A ACTIVE
IST093I E74IPPP ACTIVE
IST464I LINK STATION E74PG1A HAS CONTACTED E04NPU SA 4
IST093I E74PG1A ACTIVE
IST720I E74IPLP5 HAS CONTACTED C72SVT6 IN NETX, SA 78
IST093I E74IPLP5 ACTIVE
```

© 2006 IBM Corporation

The IPTG TCP connection can be secured using STUNNEL on both the two Linux systems

- **Install STUNNEL Package using YAST (on SUSE):**
  - From YAST main menu, select “Software”
  - Select “Install and Remove Software”
  - In the Search field enter “stunnel”
  - In the package list check the box next to stunnel
  - Click the Accept button to install the stunnelpackage

- **Generate PEM Files**
  - cd/etc/stunnel
  - opensslreq -newkeyrsa:1024 –keyoutkey.pem–nodes –x509 –days 365 –out cert.pem
  - cat key.pemcert.pem>stunnel.pem
  - rmkey.pemcert.pem

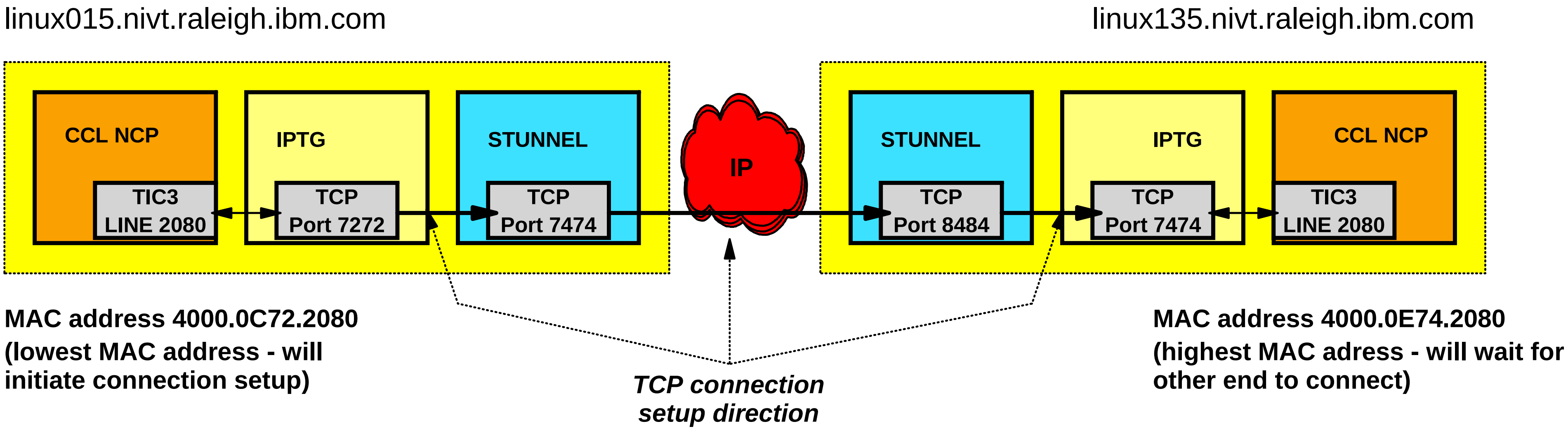
- **When defining the STUNNEL configuration, you are permitted to configure both secure and non-secured connections in the same profile by defining multiple LOCALNODE definitions. Each LOCALNODE definition would have a unique port.**

- **For an IPTG connection, the side with largest TIC3 physical LOCADDR will listen for a connection from the other side.**

© 2006 IBM Corporation

STUNNEL flows

- Initiating IP-TG PU connects to local stunnel (the one with the lowest MAC address)
- Local stunnel opens connection to remote stunnel
- Remote stunnel connects to the target IP-TG port
- Data on IPTG connection is sent to local stunnel
- Local stunnel encrypts and forwards data to remote stunnel
- Remote stunnel decrypts and forwards data to the target IP-TG port



MAC address 4000.0C72.2080 (lowest MAC address - will initiate connection setup)

TCP connection setup direction

MAC address 4000.0E74.2080 (highest MAC adress - will wait for other end to connect)

© 2006 IBM Corporation

NCP C72SVT6 - TIC3 physical and logical line definitions

Physical LINE and PU

```
***********************************************************************
* PHYSICAL TOKEN RING INTERFACE FOR TCP/IP CONNECTIONS - TIC3 2080    *
***********************************************************************
*
C72IPPG  GROUP ECLTYPE=(PHY,SUB),ADAPTER=TIC3,ANS=CONT,ISTATUS=ACTIVE, X
               RCVBUFC=32000,MAXTSL=16732,RETRIES=(20,5,5)
*
C72IPPL  LINE  ADDRESS=(2080,FULL),PORTADD=80,                         X
               LOCADD=40000C722080,NPACOLL=NO
C72IPPP  PU    ADDR=01,XMONLNK=YES
```

Logical LINE and PU

```
***********************************************************************
* LOGICAL INN TCP/IP CONNECTIONS                                      *
***********************************************************************
*
C72IPLG  GROUP ECLTYPE=(LOGICAL,SUBAREA),ANS=CONT,ISTATUS=ACTIVE,      X
               PHYSRSC=C72IPPP,SDLCST=(C72PRI,C72SEC),NPACOLL=NO,      X
               T2TIMER=(1.5,2.0,3),LOCALTO=13.5,REMOTTO=18.2
*
*----------------------------------------------------------------------
* Linkstation to E74 – IPTG INN Connection
*----------------------------------------------------------------------
*
C72IPLL5 LINE  TGN=1,TGCONF=SINGLE,MONLINK=YES
C72IPLP5 PU    ADDR=0840000E742080,SSAP=(08,H),NETID=NETX
```

© 2006 IBM Corporation

C72SVT6 - CCLDEFS and STUNNEL definitions for IPTG

```
ccldefs
  TCPDEFS
*--------------------------------------------------
* Define Local IPTG Port
*--------------------------------------------------
*
    LOCALNODE
      IPPORT              7272
      IPTOS              LOWDELAY
*
*--------------------------------------------------
* Define stunnel IPTG Port
*--------------------------------------------------
*
      REMOTENODE
        IPTOS           LOWDELAY
        PUNAME          C72IPLP5
        HOST            linux015.nivt.raleigh.ibm.com
        IPPORT          7474
     ENDTCPDEFS
endccldefs
```

CCLDEFS file for defining IPTG connectivity to SNI Partner NCP

This is the endpoint with the lowest MAC address, so this is the endpoint that initiates the connection setup towards the other endpoint.

Local hostname (to connect to local STUNNEL)

Local STUNNEL's port to which local IPTG connects

STUNNEL configuration file for defining secured IPTG connectivity to SNI Partner NCP

```
stunnel_out.conf
   client = yes
   pid = /var/run/stunnel_out.pid
   [ccl2ipll]
   accept  = 7474
   connect = linux135.nivt.raleigh.ibm.com:8484
```

Remote STUNNEL hostname and port to connect to

© 2006 IBM Corporation

NCP E74SVT6 - TIC3 physical and logical line definitions

Physical LINE and PU

```
**********************************************************************
* PHYSICAL TOKEN RING INTERFACE FOR TCP/IP CONNECTIONS - TIC3 2080   *
**********************************************************************
*
E74IPPG  GROUP ECLTYPE=(PHY,SUB),ADAPTER=TIC3,ANS=CONT,ISTATUS=ACTIVE, X
               RCVBUFC=32000,MAXTSL=16732,RETRIES=(20,5,5)
*
E74IPPL  LINE  ADDRESS=(2080,FULL),PORTADD=80,                         X
               LOCADD=40000E742080,NPACOLL=NO
E74IPPP  PU    ADDR=01,XMONLNK=YES
```

Logical LINE and PU

```
**********************************************************************
* LOGICAL INN TCP/IP CONNECTIONS                                     *
**********************************************************************
*
E74IPLG  GROUP ECLTYPE=(LOGICAL,SUBAREA),ANS=CONT,ISTATUS=ACTIVE,      X
               PHYSRSC=E74IPPP,SDLCST=(E74PRI,E74SEC),NPACOLL=NO,      X
               T2TIMER=(1.5,2.0,3),LOCALTO=13.5,REMOTTO=18.2
*
*---------------------------------------------------------------------
* Linkstation to C72 – IPTG INN Connection
*---------------------------------------------------------------------
*
E74IPLL5 LINE  TGN=1,TGCONF=SINGLE,MONLINK=YES
E74IPLP5 PU    ADDR=0840000C722080,SSAP=(08,H),NETID=NETX
```

© 2006 IBM Corporation

C72SVT6 - CCLDEFS and STUNNEL definitions for IPTG

```
ccldefs
  TCPDEFS
*--------------------------------------------------
* Define Local IPTG Port
*--------------------------------------------------
*
    LOCALNODE
      IPPORT        7474
      IPTOS        LOWDELAY
*

     ENDTCPDEFS
endccldefs
```

CCLDEFS file for defining IPTG connectivity to SNI Partner NCP

This is the endpoint with the highest MAC address, so this is the endpoint that will wait for the other endpoint to connect to it.

Local IPTG's port to which local STUNNEL will connect when remote STUNNEL has connected to local STUNNEL

Local STUNNEL's port to which remote STUNNEL connects

```
stunnel_in.conf
   [ccl2iptg]
   accept  = 8484
   connect = 7474
   TIMEOUTclose = 0
```

STUNNEL configuration file for defining secured IPTG connectivity to SNI Partner NCP

© 2006 IBM Corporation

For more information....

URL	Content
http://www.ibm.com/servers/eserver/zseries	IBM eServer zSeries Mainframe Servers
http://www.ibm.com/servers/eserver/zseries/networking	Networking: IBM zSeries Servers
http://www.ibm.com/servers/eserver/zseries/networking/technology.html	IBM Enterprise Servers: Networking Technologies
http://www.ibm.com/software/network/commserver	Communications Server product overview
http://www.ibm.com/software/network/commserver/zos/	z/OS Communications Server
http://www.ibm.com/software/network/commserver/z_lin/	Communications Server for Linux on zSeries
http://www.ibm.com/software/network/ccl	Communication Controller for Linux on zSeries
http://www.ibm.com/software/network/commserver/library	Communications Server products - white papers, product documentation, etc.
http://www.redbooks.ibm.com	ITSO Redbooks
http://www.ibm.com/software/network/commserver/support	Communications Server technical Support
http://www.ibm.com/support/techdocs/	Technical support documentation (techdocs, flashes, presentations, white papers, etc.)
http://www.rfc-editor.org/rfcsearch.html	Request For Comments (RFC)

© 2006 IBM Corporation