IBM TRAINING

L45

Communication Controller for Linux (CCL) on System z: How to implement

Alfred B Christensen - alfredch@us.ibm.com

IBM
SYSTEM z9 AND zSERIES EXPO
October 9 - 13, 2006

Orlando, FL

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.

RETURN TO INDEX

Agenda

1. **CCL implementation and NCP migration planning**
2. **Sample SNA network before CCL**
3. **SNA LLC2 on an Ethernet LAN**
4. **CCL DLSw**
5. **CDLC channel connectivity to CCL**

CCL implementation and NCP migration planning

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

CCL project outline (continued)

- **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)

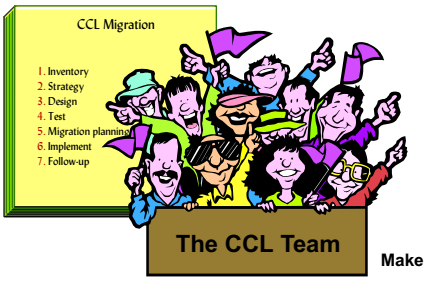
CCL provides improved options for redundancy design: you don't need to buy an extra IBM 3745 to deploy a stand-by NCP.

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

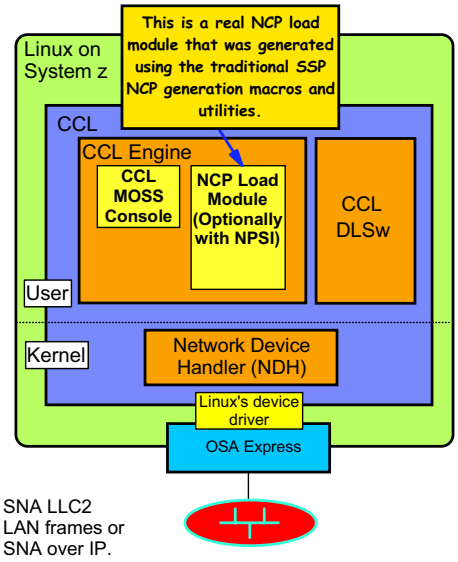
CCL project outline (continued)

- **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!

CCL overall structure and components - recap

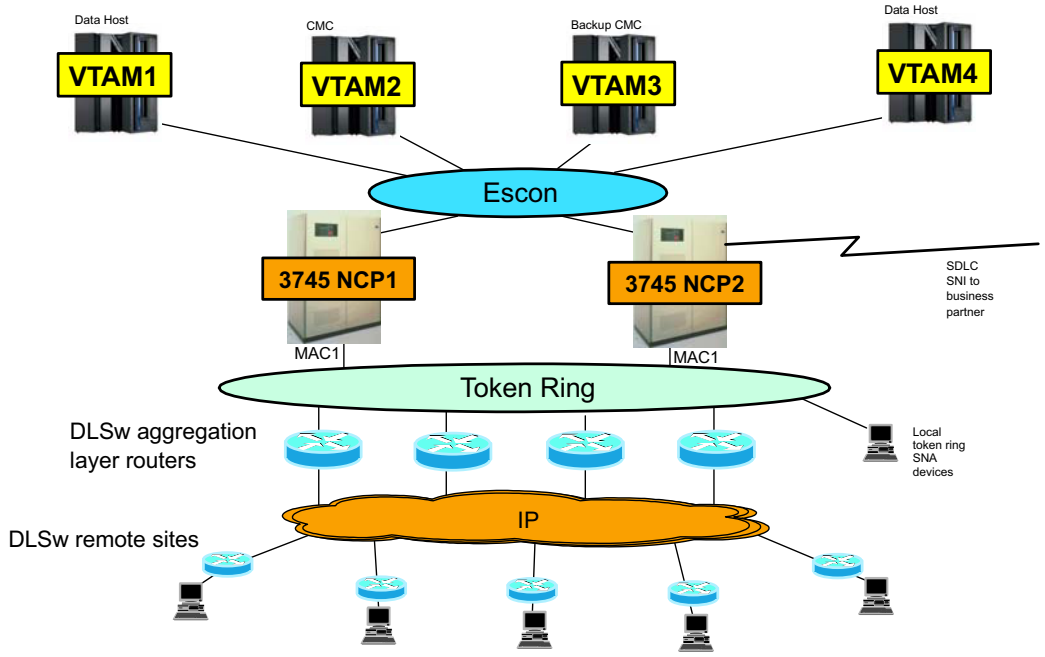


Note: You will continue to use ACF/SSP to generate, load, and dump an NCP load module.

- **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

Sample SNA network before CCL

Sample SNA network (diagram)



Sample SNA network (description)

- **Multiple VTAMs,each with an ESCON connection to the NCPs**
  - One CMC, one backup CMC, and two data hosts
- **Two 3745s**
  - For redundancy
- **Some SNI connections to business partners**
  - Perhaps over SDLC leased lines
- **SNA BNN devices connect to NCPs over token ring**
  - Duplicate TIC config: 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
- **Most are remote**
  - DLSw router at the remote site terminates the SNA LLC 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 connection
    - 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
    - LLC connection between the 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

SNA LLC2 on an Ethernet LAN

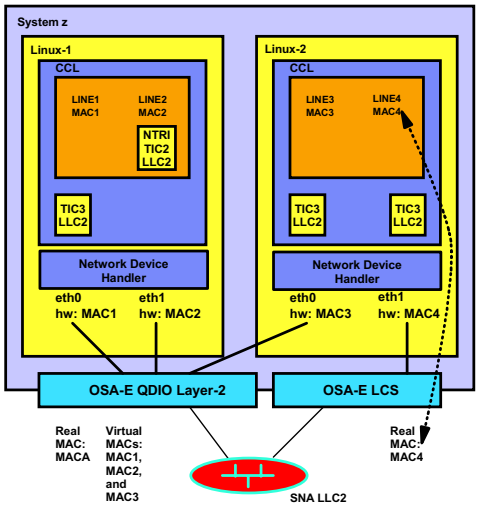
CCL SNA LLC2 LAN connectivity

- **NCP definitions for two types of LAN adapters:**
  - **TIC2** - NCP Token-ring Interface (NTRI) support
    - NCP instructions used for LLC2 logic
  - **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 !!!

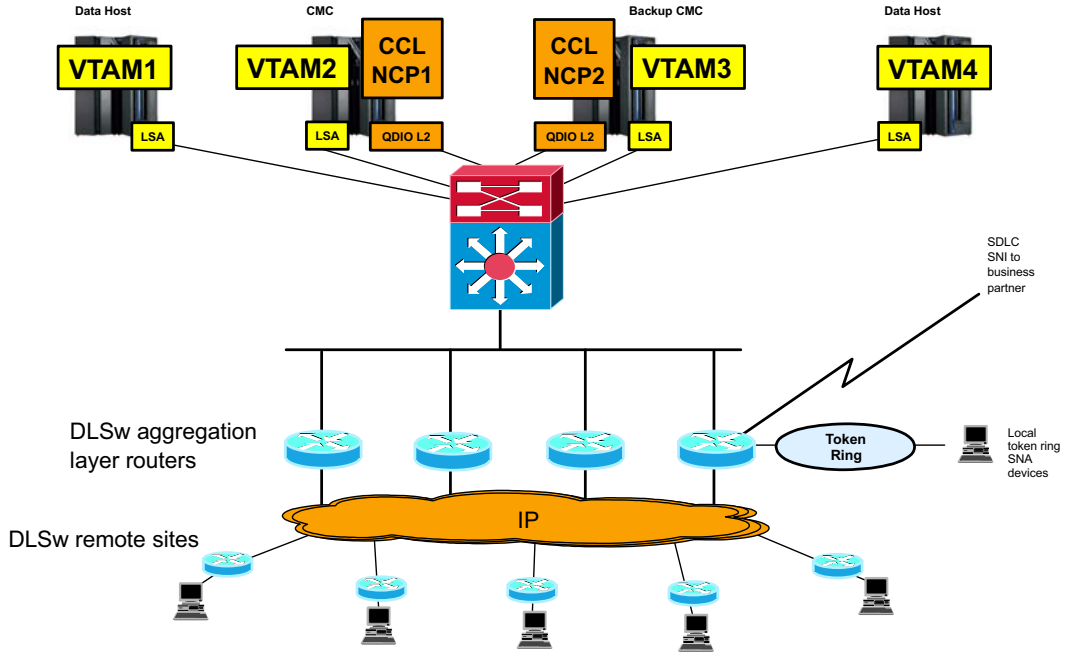
- **Linux has two different LAN device drivers for LLC2:**
  - **LAN Channel Station (LCS)**
    - Copper cabling, token ring or Ethernet
    - NCP physical LINE local MAC address specification must match OSA port's configured real MAC address (OSA/SF)
    - One NCP physical line per OSA LCS port
    - 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

Migration to CCL using LLC2 over Ethernet (diagram)



Migration to CCL using LLC2 over Ethernet (description)

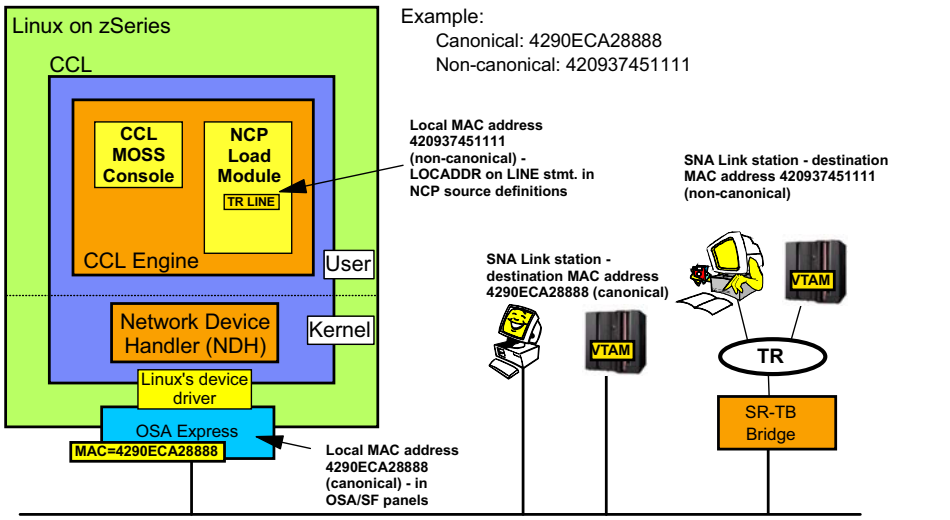
- **Install two CCLs, one in each of two CECs (for redundancy)**
  - Same logical subarea configuration (four VTAMs, two NCPs), but running NCP inside CCL instead of inside 3745
- **All SNA traffic now flows over a high speed ethernet core**
  - Removed the 3745s, the ESCON that was used by the 3745s, 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 100M) connectivity between the NCPs and the aggregation layer routers, instead of 16M token ring**
  - NCP gen 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 gen 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

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

Ethernet considerations - canonical or non-canonical MAC address




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

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)
```

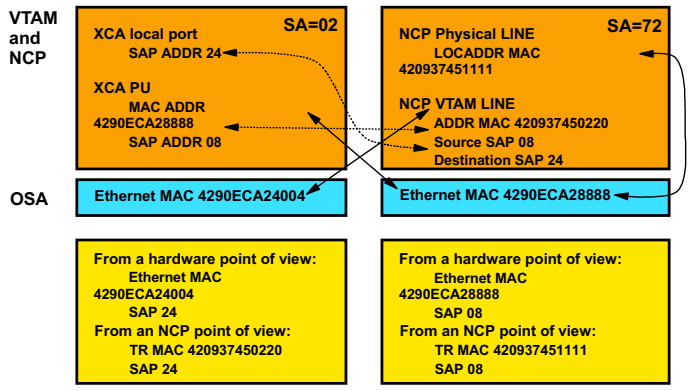
NCP name in this example is C72DUPE

- **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**

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



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=

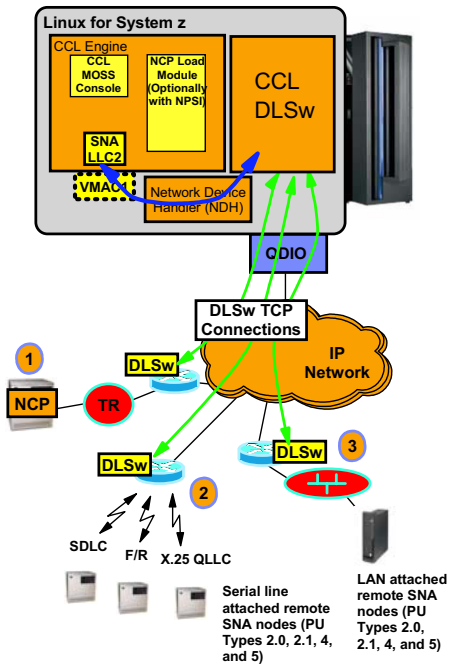
Sample VTAM activation and display commands

```
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.

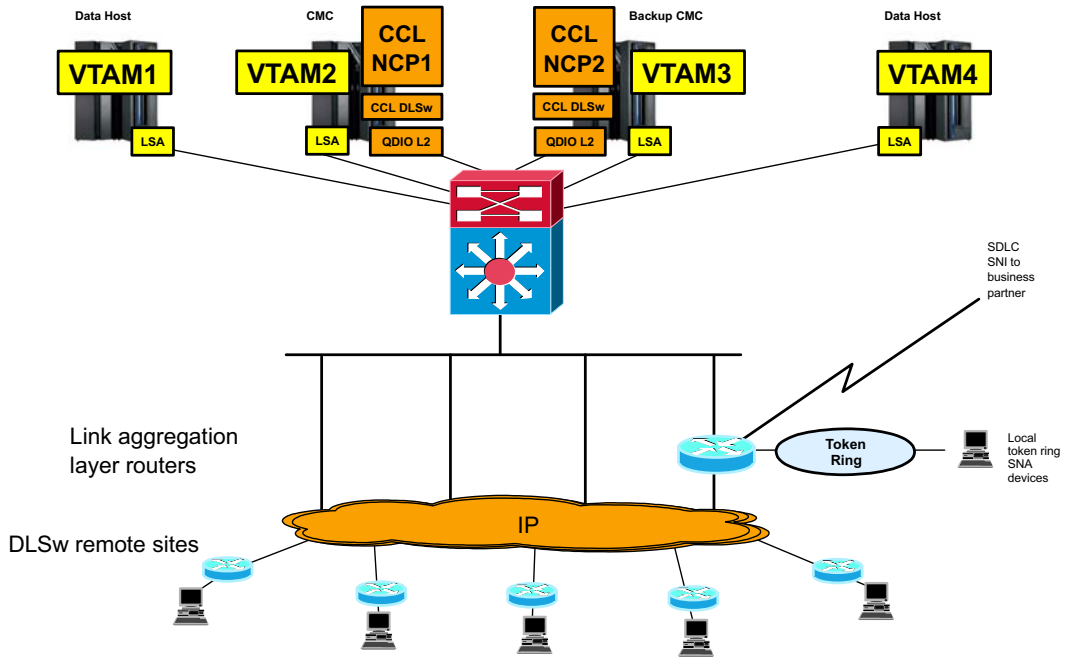
CCL DLSw

CCL imbedded DLSw support



- **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)

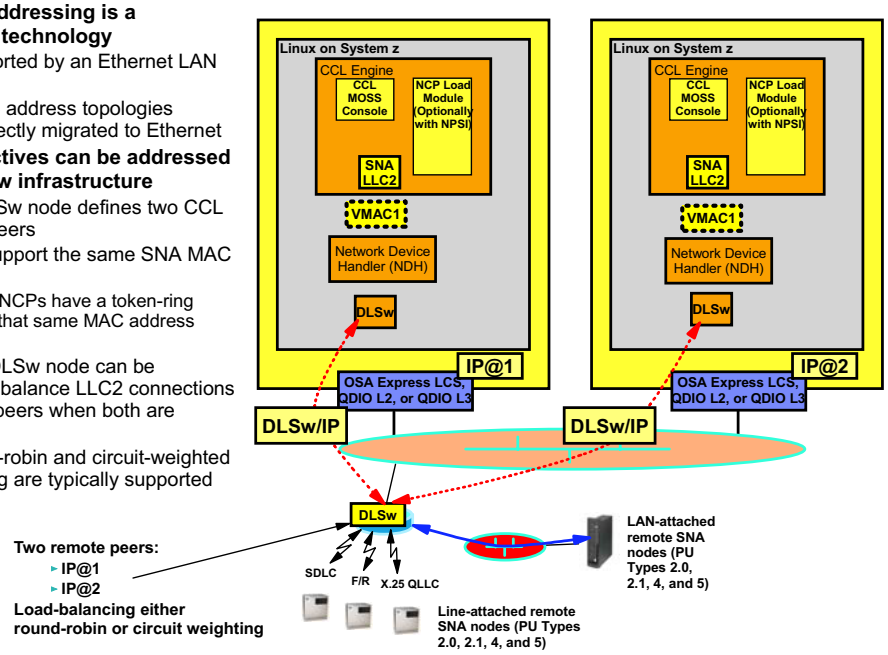


Migration to CCL using CCL DLSw (description)

- **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 gens 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

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



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

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?**

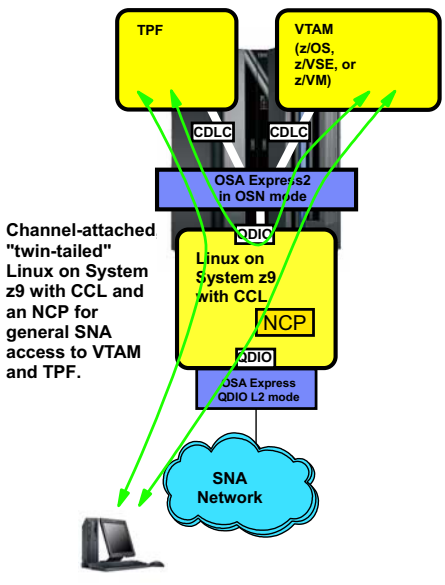
```
DLSw>show tcp peer
   Multicast
   IP Address      IP Address      Conn State     CST Version  ActSes SesCreates
   --------------- --------------- -------------- --- -------- ------ ----------
1                  9.42.103.140    ESTABLISHED     a  AIW V2R0      1          1
DLSw>
```

- **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>
```

CDLC channel connectivity to CCL

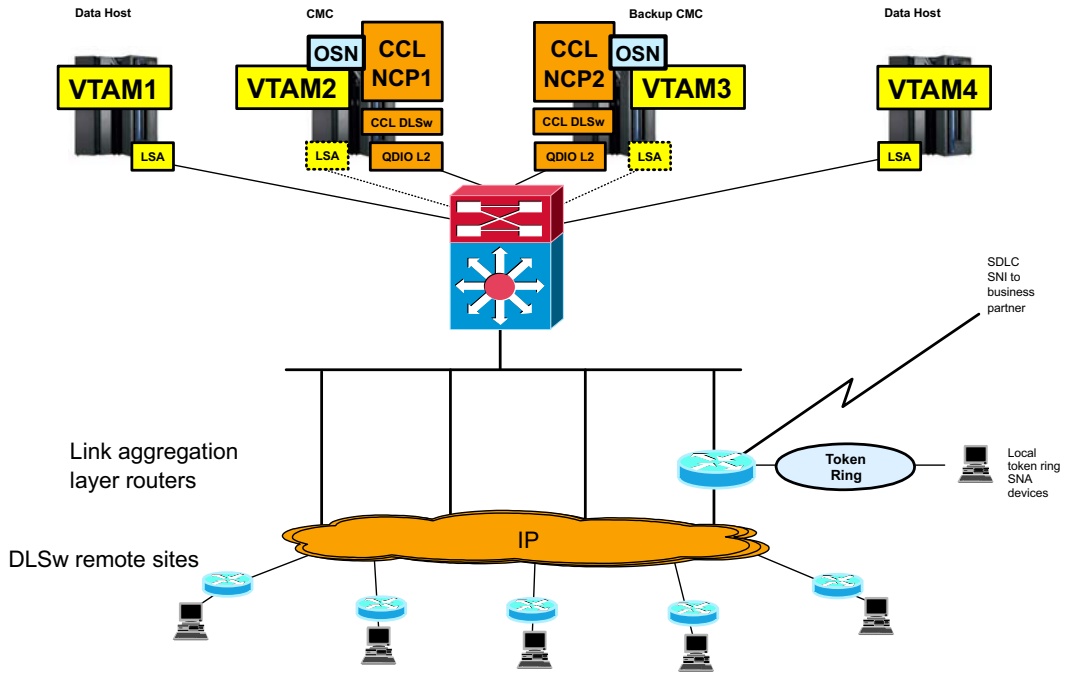
OSA for NCP (OSN) connectivity on System z9 - CDLC



Channel-attached "twin-tailed" Linux on System z9 with CCL and an NCP for general SNA access to VTAM and TPF.

- **OSA Express2 on System z9 implements an OSA CHPID type - known as OSA for NCP (OSN).**
  - OSA-Express2 1000Base-T and 1 Gigabit Ethernet features on z9-109
- **TPF and VTAM see the OSA Express OSN port as a channel-attached IBM 3745 to which they communicate using the usual CDLC channel protocol.**
  - OSA CHPID defined as OSN
  - TPF and VTAM device number defined as IBM 3745
  - Linux device numbers (three in a set) defined as OSN (accessed through QDIO from Linux)
  - Same OSN port can be shared among more SNA host systems and more CCL NCPs
- **The OSA microcode relays the CDLC data over a QDIO interface to CCL, which presents it to the NCP as though it had arrived over an IBM 3746 channel interface.**
  - No cable needs to be attached to an OSA port that is defined in OSN mode
- **The fact that the NCP runs in CCL instead of an IBM 3745/46 is transparent to TPF and VTAM.**
  - Existing configuration definitions are used unchanged
  - Existing activation and management flows continue to work as before
- **The normal Load/Dump functions over a channel are supported**
  - No need to FTP an NCP load module to the Linux file system
- **TPF or VTAM must reside on the same System z9 CEC as where CCL resides**
  - This is a same-CEC connectivity technology

Migration to CCL using CCL DLSw (diagram)



IOCP definitions for OSN

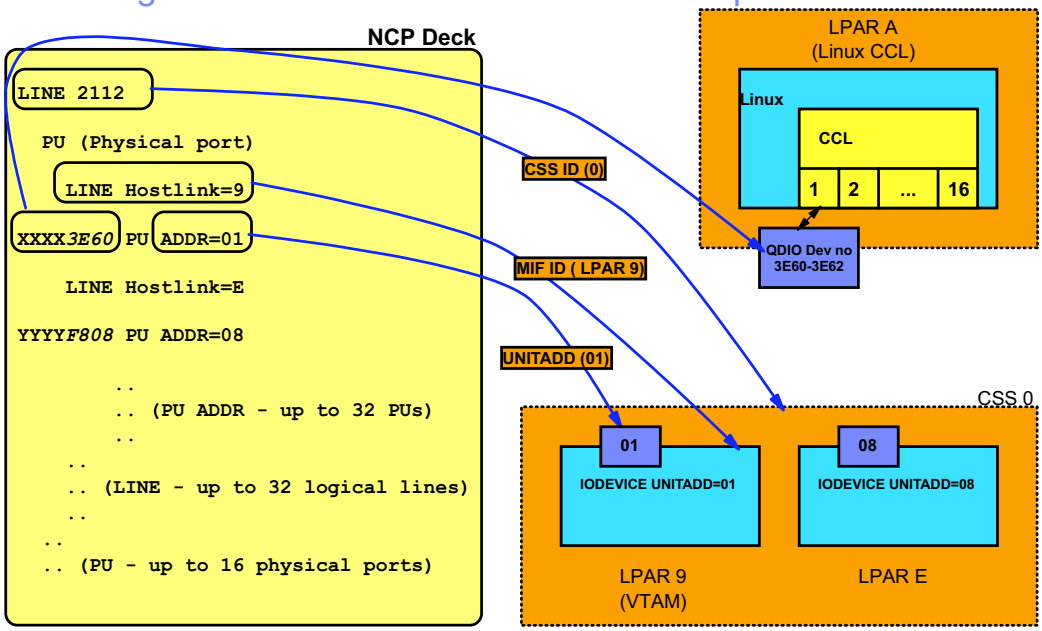
```
RESOURCE PART=((CSS(0),(RANS38,8),(RANS39,9),(RALNS27,D)))
*
CHPID PCHID=320,PATH=(CSS(0,1),F4),TYPE=OSN,SHARED
CHPID PCHID=321,PATH=(CSS(0,1),F5),TYPE=OSN,SHARED
*
CNTLUNIT CUNUMBR=3E40,PATH=((CSS(0),F4),(CSS(1),F4)),UNIT=OSN
CNTLUNIT CUNUMBR=3E60,PATH=((CSS(0),F5),(CSS(1),F5)),UNIT=OSN
*
IODEVICE  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

Correlating definitions between IOCP and NCP parameters



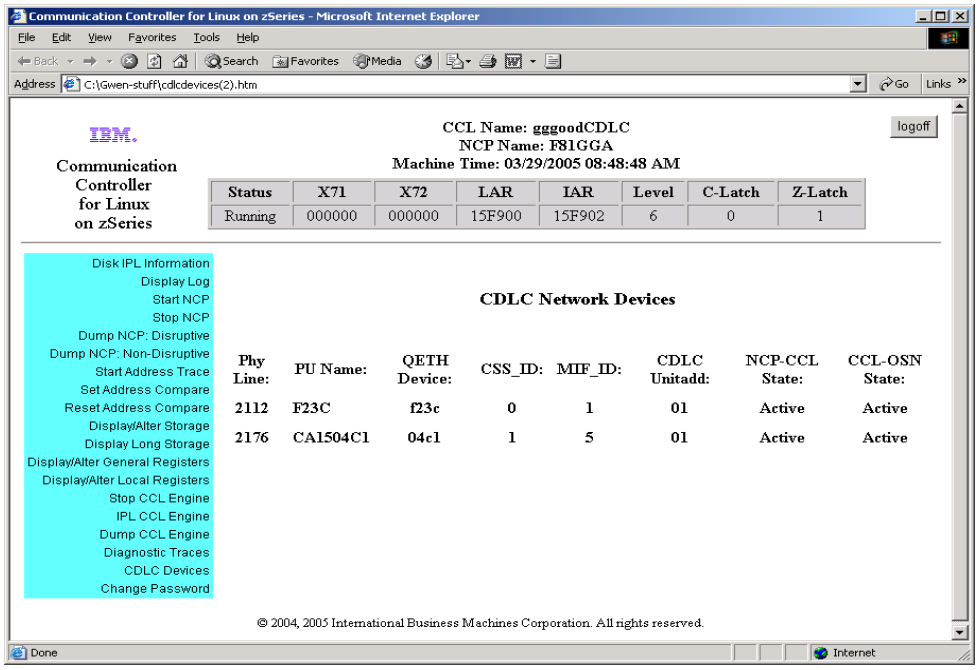
If the above naming and parameter guidelines cannot be met, mapping between NCP definitions and the information needed in the OSN context, can be encoded in a CCLDEFS text file.

Fixed relationship between Escon LINE addressesd and CSS IDs: LINE 2112: CCS 0, LINE 2176: CSS 1, LINE 2240: CCS 2, etc.

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.**

MOSS console interface to CDLC network devices



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
```

CCL NCP ESCON channel 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

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
```

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)