Configuring Layer 2 Switch on zVM

Sample Definitions for Communications Controller for Linux on System z9 and zSeries

Target Audience

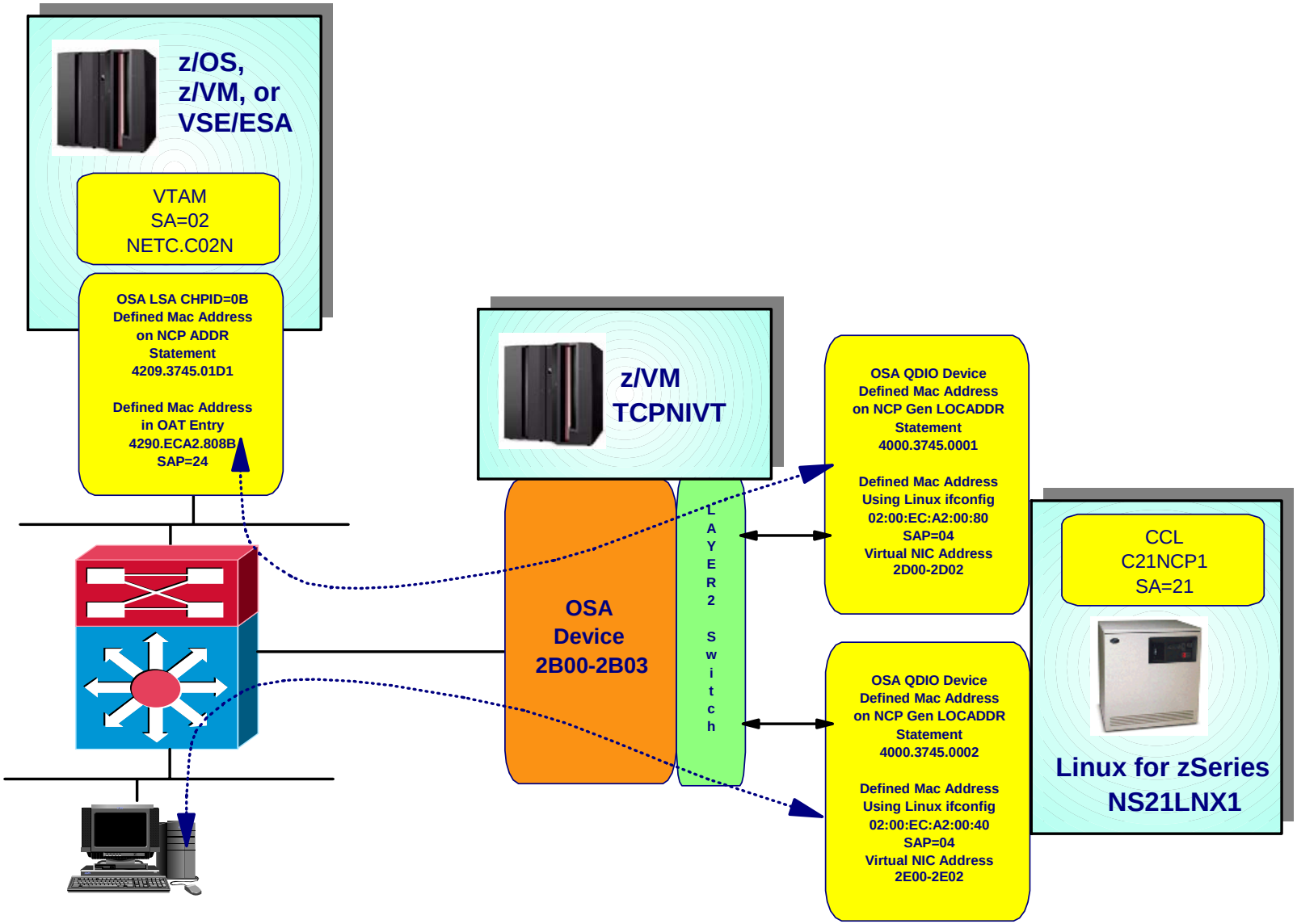
- Customers who wish to reduce the number of copper OSA adapters when using Communication Controller for Linux z/Series (CCL). QDIO in Layer 2 mode will support both INN and BNN connections.

Purpose of this Paper

The intent of this paper is to provide a sample solution for customers during the migration from 3745/3746-900 FEPs to Communication Controller for Linux z/Series (CCL). This document will provide working examples of the following:

- VTAM XCA Major Node – VTAM to CCL
- NCP Physical and Logical lines – CCL to 3745
  - VTAM to Communication Controller for Linux z/Series
  - 3745 to Communication Controller for Linux z/Series
- Commands to define OSA QDIO adapters in Layer 2 mode

Test Configuration



Resources Used for Solution Verification

- One z/OS Communications Server
- One Linux guest running under z/VM
  - 2048 mb of memory
  - 2 Real CPs
  - 2 3390-3 DASD volumes
- One OSA Fast Ethernet adapter
  - Required for LSA connection in z/OS.
- One QDIO OSA Adapter
- Layer 2 or Layer 3 Ethernet Switches

Configure the TCPNIVT ID to Control the Virtual Switch in the TCP/IP Profile

```
; ------------------------------------------------------------------------
; 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
```

Add the Virtual Switch IUCV in the USER.DIRECTORY for TCPNIVT ID

```
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.

Define the Layer 2 Virtual Switch in the PROFILE EXEC in the TCPNIVT ID and Grant Access to Linux Guests

```
/*************************/
/* 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
```

- DEFINE VSWITCH VSWLAY2 – defines the Virtual Switch named VSWLAY2
- RDEV 2B00 ETH – use real OSA device 2B00 – ETH means this is a Layer 2 switch
- CONTR TCPNIVT – z/VM userid defined as the controller for the Virtual Switch. In this example the ID is called TCPNIVT
- PORTNAME LAY2PORT – unique name that identifies the OSA Express adapter. Up to 3 names can be defined.

- The SET command authorizes userid NS21LNX1 to access the Virtual Switch

Add the NICDEF to the USER.DIRECTORY for the Linux guest NS21LNX1

```
USER NS21LNX1 NS21LNX1  512M 1024M BG 64
   INCLUDE IBMDFLT
   CPU 0 NODEDICATE
   CPU 1 NODEDICATE
   MACHINE ESA 4
   OPTION QUICKDSP
   DEDICATE 2EA2 2EA2
   DEDICATE 2EA3 2EA3
   DEDICATE 2F98 2F98
   DEDICATE 2F99 2F99
   DEDICATE 2F9A 2F9A
   DEDICATE 245D 245D
   DEDICATE 245E 245E
   DEDICATE 2EB2 2EB2
   DEDICATE 2EB3 2EB3
   LINK  MAINT  19B 19B RR
   MDISK 191 3390 1606 005 510W02 MR
   NICDEF 2D00 TYPE QDIO
   NICDEF 2E00 TYPE QDIO
```

NS21LNX1 – PROFILE EXEC -- Couple the NIC to the Layer 2 Virtual Switch

- 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).

Define the Layer2 Interface 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
```

Define the Layer2 Interface to Linux -- Network

Create this script ifcfg-qeth-bus-ccw-0.0.2d00 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 two scripts will need to be replicated for device address 2D00. Device 2D00 will have an LLADDR='02:00:ec:a2:00:80‘
- MAC Address defined on the NCP LOCADDR statement is the non-canonical version of this address – 4000.3745.0001

Define the Layer2 Interface to Linux -- Hardware

Create the script hwcfg-qeth-bus-ccw-0.0.2e00 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='0.0.2e00 0.0.2e01 0.0.2e02'

# 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
```

Define the Layer2 Interface to Linux -- Network

Create this script ifcfg-qeth-bus-ccw-0.0.2e00 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 two scripts will need to be replicated for device address 2E00. Device 2E00 will have an LLADDR='02:00:ec:a2:00:40‘
- MAC Address defined on the NCP LOCADDR statement is the non-canonical version of this address – 4000.3745.0002

C02XCA – XCA Major Node Definitions

```
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
```

C21NCP1 – NTRI Physical Line Definitions

```
************************************************************
* 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
```

C21NCP1 – NTRI Logical Lines – INN and BNN

```
*********************************************************************
* 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)
```

Starting CCL from Linux – CCLV1R1

- From the Linux console, change to the CCL directory:
  - cd /opt/ibm/Communication_Controller_for_Linux/
- Load the CCL kernel module
  - ./load_ndh.sh
    - You will receive the message :

      NDH kernel modules loaded. You are now able to run the cclengine
- Start the CCL engine
  - nohup ./cclengine –mC21NCP1 -p2021 SVTC21 &
    - If you use telnet or ssh into the Linux host you will want to preface the command with “nohup” so that the process will remain active even after the telnet/ssh session is terminated.

Starting CCL from Linux – CCLV1R2

- From the Linux console, change to the CCL directory:
  - cd /opt/ibm/ndh
- Load the CCL kernel module
  - ./load_ndh.sh
    - You will receive the message :

      NDH kernel modules loaded. You are now able to run the cclengine
- From the Linux console, change to the CCL directory:
  - cd /opt/ibm/Communication_Controller_for_Linux/
- Start the CCL engine
  - nohup ./cclengine –mC21NCP1 -p2021 SVTC21 &
    - If you use telnet or ssh into the Linux host you will want to preface the command with “nohup” so that the process will remain active even after the telnet/ssh session is terminated.

Activating NCP using XCA from NETC.C02N

- **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
```

Reference Documentation

Networking Overview for Linux on zSeries

- http://www.redbooks.ibm.com/redpapers/pdfs/redp3901.pdf