This presentation is an in depth look at configuration of the z/VM TCP/IP server. Two separate methods of configuration will be discussed: A "simplified" approach using the IPWIZARD and IFCONFIG tools, as well as the more advanced method of updating the configuration files directly. Topics such as elementary routing, network hardware, and security are discussed in as much depth as necessary to provide an understanding of how to configure them on the z/VM TCPIP server. Some common configuration errors will also be addressed. While prior experience with z/VM TCP/IP is not necessary for attendees, some basic knowledge of z/VM mini disk structure is assumed.

References to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM's product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe on any of the intellectual property rights of IBM may be used instead. The evaluation and verification of operation in conjunction with other products, except those expressly designed by IBM, are the responsibility of the user.

The following terms are trademarks of IBM Corporation in the United States or other countries or both:
IBM    z/VM

Linux is a trademark of Linus Torvalds in the United States, other countries, or both. Other company, product, and service names, which may be denoted by double asterisks (**), may be trademarks or service marks of others.

© Copyright International Business Machines Corporation, 1998-2007
Agenda

- Creating an Initial TCP/IP Configuration
- General CMS TCP/IP Client Configuration
- Modifying the TCP/IP Server Definition
- Customizing the TCP/IP Configuration File
- Managing Network Interfaces Dynamically
Creating an Initial TCP/IP Configuration
The IPWIZARD Command

- IPWIZARD can be used to build an initial configuration for the TCP/IP stack
- A “Basic IP Connectivity Worksheet” is provided to assist in information gathering prior to install
  - See the “Plan Your Installation” chapter in “z/VM Guide for Automated Installation and Service”
- IPWIZARD is not a migration tool
  - Existing configuration will be backed up and replaced
- IPWIZARD creates a minimal initial configuration
  - Only basic connectivity through one interface
  - Static routing
Using IPWIZARD

- IPWIZARD is a full screen 3270 application
- Run the IPWIZARD command from MAINT as directed by the Guide for Automated Installation and Service
- Fill in the panels with the information from the worksheet
  - Most fields are required. IPWIZARD will prompt for missing information
- Have IPWIZARD process the input (PF5 on the last panel)
IPWIZARD Results

- Based on your input, the following files are created:
  - On the TCP/IP server's 198 disk:
    - PROFILE TCPIP
    - SYSTEM DTCPARMS
  - On the TCP/IP server's 592 disk:
    - TCPIP DATA

- Basic connectivity tests are performed
  - Should something fail, you have the option to return to the panels and verify and/or re-enter information
General CMS Client Configuration
TCPIP DATA File

- Information used by both clients and servers
  - Resolver Information
    - host name
    - domain name
    - resolver preferences
    - resolver tracing
    - DNS servers to use
  - Stack virtual machine name
  - E-mail servers

- Should reside on the TCP/IP Server's 592 disk (client code)
**Sample TCPIP DATA**

<table>
<thead>
<tr>
<th>TCPIPUSERID</th>
<th>TCPIP2</th>
</tr>
</thead>
<tbody>
<tr>
<td>NSINTERADDR</td>
<td>10.6.3.252</td>
</tr>
<tr>
<td>NSINTERADDR</td>
<td>10.6.3.253</td>
</tr>
<tr>
<td>HOSTNAME</td>
<td>vmhost1</td>
</tr>
<tr>
<td>DOMAINORIGIN</td>
<td>testnet.bigblue.com</td>
</tr>
</tbody>
</table>
Modifying the TCP/IP Server Definition
The DTCPARMS File

- Defines a server and various startup parameters
- Formatted as a CMS Names file (:<tag>.<value> pairs)
- Search Order
  - <userid> DTCPARMS
  - <nodeid> DTCPARMS
  - SYSTEM DTCPARMS
  - IBM DTCPARMS
- All but IBM DTCPARMS should reside on the TCP/IP server's 198 disk
Sample SYSTEM DTCPARMS

:nick.TCPIP :type.SERVER :class.STACK
:attach.1f08-1f09, 1e00-1e02
:vnic.e100 SYSTEM LAN1
:vctc.800 LINUX1 500,
    801 LINUX1 501
DTCPARMS Tags

- Required Tags
  - :NICK.
    - The user ID of the server being configured
  - :TYPE.
    - For IBM servers the type is “server”
  - :CLASS.
    - Defines what sort of server is being configured (FTP, TCP/IP Stack, MPRoute)
DTCPARMS Tags (cont.)

- Other Important Tags
  - :ATTACH.
    - Attach a device to your TCP/IP Server
  - :VNIC.
    - Define and couple a virtual network card to a guest LAN or VSWITCH
  - :VCTC.
    - Define and couple a virtual channel-to-channel device to another user

- Also Typical
  - :OWNER.
    - Define the owner of the TCP/IP Server (Default: TCPMAINT)
  - :EXIT.
    - Run a user defined exit
Server Profile Exits

- Global exit called for all servers: TCPRUNXT EXEC
- Server-specific exit called via :Exit. tag
- Input: when called, server class
- Output: DTCPARMS-type tags
- Call 'types': SETUP, BEGIN, END, ADMIN and ERROR
- Example:

```
arg calltype class .
if calltype = "SETUP" & class = "STACK" then
    begin
    /* ... find current node ID... */
    if nodeID = DRnodeID then
    /* Do some DR setup here */
    end
```
Customizing the TCP/IP Configuration File
PROFILE TCPIP

- Primary TCP/IP server configuration file
- Search order:
  - <userid> TCPIP
  - <nodeid> TCPIP
  - PROFILE TCPIP
- Only one is used
- Should reside on the TCP/IP server's 198 disk
Configuration Statements - Pool Sizes

- Pool size configuration statements MUST appear first in the TCP/IP configuration file.
- These may not be changed while the stack is running.
- Values determined by TCP/IP stack's workload.
- 16 different pool statements are available in 3 formats:
  - xxxBUFFERPOOLSIZE or xxxENVELOPEPOOLSIZE
    - <statement> <number of buffers allocated> <buffer size>
  - xxxPOOLSIZE
    - <statement> <number of control blocks allocated>
  - FIXEDPAGESTORAGEPOOL
    - FIXEDPAGESTORAGEPOOL <initial number> <maximum number>
Configuration Statements - Pool Sizes (cont.)

netstat pool
VM TCP/IP Netstat Level 520

TCPIP Free pool status:

<table>
<thead>
<tr>
<th>Object</th>
<th>No. alloc</th>
<th>No. free</th>
<th>Lo-water</th>
<th>Permit size</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACB</td>
<td>5006</td>
<td>4958</td>
<td>4794</td>
<td>500</td>
</tr>
<tr>
<td>CCB</td>
<td>755</td>
<td>662</td>
<td>654</td>
<td>75</td>
</tr>
<tr>
<td>Dat buf</td>
<td>1200</td>
<td>1165</td>
<td>1123</td>
<td>120</td>
</tr>
<tr>
<td>Sm dat buf</td>
<td>5000</td>
<td>4804</td>
<td>4787</td>
<td>500</td>
</tr>
<tr>
<td>Tiny dat buf</td>
<td>108</td>
<td>98</td>
<td>96</td>
<td>10</td>
</tr>
<tr>
<td>Env</td>
<td>1250</td>
<td>1229</td>
<td>1154</td>
<td>125</td>
</tr>
<tr>
<td>Lrg env</td>
<td>75</td>
<td>74</td>
<td>75</td>
<td>7</td>
</tr>
<tr>
<td>RCB</td>
<td>51</td>
<td>49</td>
<td>48</td>
<td>5</td>
</tr>
<tr>
<td>SCB</td>
<td>2014</td>
<td>1949</td>
<td>1823</td>
<td>201</td>
</tr>
<tr>
<td>SKCB</td>
<td>256</td>
<td>216</td>
<td>195</td>
<td>25</td>
</tr>
<tr>
<td>TCB</td>
<td>5000</td>
<td>4764</td>
<td>4740</td>
<td>500</td>
</tr>
<tr>
<td>UCB</td>
<td>512</td>
<td>501</td>
<td>499</td>
<td>51</td>
</tr>
<tr>
<td>Add Xlate</td>
<td>1512</td>
<td>1512</td>
<td>1512</td>
<td>5</td>
</tr>
<tr>
<td>NCB</td>
<td>1501</td>
<td>1501</td>
<td>1501</td>
<td>5</td>
</tr>
<tr>
<td>IP Route</td>
<td>3015</td>
<td>2837</td>
<td>2832</td>
<td>60</td>
</tr>
<tr>
<td>IPv6 Route</td>
<td>3018</td>
<td>3018</td>
<td>3018</td>
<td>60</td>
</tr>
<tr>
<td>Segment ACK</td>
<td>100000</td>
<td>99967</td>
<td>99877</td>
<td>10000</td>
</tr>
</tbody>
</table>

FPSP total locked pages: 852, Unused locked pages: 270
FPSP allocation threshold: 41313, Low-water mark: 0
Ready; T=0.01/0.01 11:13:12
Configuration Statements - LargeEnvelopePoolSize

- `<buffer size>` must be at least as large as that of smaller envelope buffers
- Acts as an upper bound on the MTU value
- CTC connected hosts should have matching buffer sizes
- Example:
  - LARGEENVELOPEPOOLSIZE 100 32K
Configuration Statements - Privileged Users

- **INFORM**
  - Specifies users who should be informed of major stack events
    - Pool expansion
    - Denial of service attacks
    - Dynamic configuration changes

- **OBEY**
  - Specifies users which can issue privileged stack commands
    - OBEYFILE, NETSTAT OBEY
    - Use of raw sockets
    - Use privileged services on IBM servers (via SMSG)
    - Note: As of 5.3.0, OBEY authority is no longer needed to run TRACERTE

- **Examples:**
  - INFORM tcpmaint ENDINFORM
  - OBEY tcpmaint maint migueld mproute ENDOBEY
Configuration Statements - AssortedParms

- Various miscellaneous stack settings
- Look at what is available and decide what you need
- Of particular interest
  - EqualCostMultipath
  - IgnoreRedirect (if not running a dynamic router)
  - SourceVipa (if using virtual IP addresses)

Example:

ASSORTEDPARMS
  equalcostmultipath
  ignoreredirect
ENDASSORTEDPARMS
Configuration Statements - AUTOLOG, PORT

- **AUTOLOG**
  - Defines which servers to start when the stack comes up
  - Stack will restart the server if it is logged off

- **PORT**
  - Gives permission for a server to listen on a port
  - Low ports (0-1023) are restricted by default
  - Listed ports are monitored unless NOAUTOLOG is specified
  - Used to start the Telnet server (assign a port to INTCLIEN)

- **Examples:**
  - AUTOLOG
    - ftpserve 0
    - ENDAUTOLOG
  - PORT
    - 20 tcp ftpserve noautolog
    - 21 tcp ftpserve
Configuration Statements - INTERNALCLIENTPARMS

- Specifies settings for the Telnet server (internal client)
  - Timeout values
  - Exit routines
  - Linemode options
  - Dynamic TLS Settings (z/VM 5.3.0)
- Look through the options and decide what works best for you
- Example:
  - INTERNALCLIENTPARMS
    NOTN3270E
    PORT 23
    SECURECONNECTIONS allowed
    TLSCALLER mylabel
    ENDINTERNALCLIENTPARMS
Configuration Statements - Device and Link

- Configure network interfaces to the stack
- Examples:
  - Real QDIO Ethernet Device
    - DEVICE qdio0 OSD 1e00 PRIROUTER
      LINK eth0 QDIOETHERNET qdio0 MTU 1500
  - Virtual QDIO Ethernet Device
    - DEVICE qdio1 OSD e100 NONROUTER
      LINK veth0 QDIOETHERNET qdio1 MTU 32768
  - LCS Ethernet Device
    - DEVICE lcs0 LCS 1f08
      LINK eth1 ETHERNET 0 lcs0 MTU 1500
  - Virtual Channel to Channel Device
    - DEVICE ctc0 CTC 800
      LINK vctc0 CTC 1 ctc0 MTU 32760
**Configuration Statements - HOME**

- Configures IP addresses and subnet masks for each link
  - If a subnet mask is specified, a subnet route will be generated for that interface (i.e. You won't need a GATEWAY entry for it)
- Determines which VIPA address is associated with each link

**Examples:**

- **With VIPA**
  - HOME
    - 7.0.0.1 255.255.255.0 lcs1
    - 8.0.0.1 255.255.255.252 vipa1
    - 9.1.0.3 255.255.255.0 qdio1
    - 8.0.0.2 255.255.255.252 vipa2
    - 9.2.0.2 255.255.255.0 qdio2

- **Without VIPA**
  - HOME
    - 10.6.3.159 255.255.255.0 eth0
    - 192.8.12.19 255.255.255.240 veth0
    - 192.8.12.12/27 255.255.255.252 eth1
    - 192.4.0.1 255.255.255.252 vctc0
Configuration Statements - GATEWAY and START

- **GATEWAY** – Defines static routes
- **START** – Starts devices
- **Examples:**
  - GATEWAY
    - `10.6.3.0   255.255.255.0 = eth0  0`
    - `192.8.12.19 255.255.255.240 = veth0  0`
    - `192.8.12.12/27 = eth1  0`
    - `192.4.0.2  HOST = ctc0  0`
    - **DEFAULTNET**
      - `10.6.3.1  eth0  0`
    - **DEFAULTNET**
      - `192.8.12.1  eth1  0`
  - **START qdio0**
  - **START qdio1**
  - **START lcs0**
  - **START ctc0**
z/VM TCP/IP 5.3.0 and Interface High Availability

- IP takeover is supported to minimize the impact of an hardware interface failure
  - QDIO ethernet and LCS ethernet devices only
- No special parameters or options necessary
  - If the TCP/IP stack determines two interfaces are on the same network, IP takeover will be enabled for those interfaces
  - For IPv4, determination is based on the IP addresses and subnet masks of the interfaces
    - Subnet masks may be defined on the HOME statement, the GATEWAY statement, or in the MROUTE CONFIG file
Host 10.1.1.3 forms a connection with 10.1.1.1 (OSA1)
IP Takeover Details (cont.)

- OSA1 Fails
- OSA2 informs host that traffic for 10.1.1.1 should be sent through this interface.
IP Takeover Details (cont.)

10.1.1.3 starts sending packets to OSA2
Verifying the Configuration

- Use NETSTAT once the stack is up and running
  - NETSTAT GATE
    - Display the stack's routing table
  - NETSTAT HOME
    - Display the stack's HOME list
  - NETSTAT DEVLINKS
    - Display the DEVICE and LINK information and status
  - NETSTAT CONFIG ALL
    - Display information on ASSORTEDPARMS, INTERNALCLIENTPARMS, OBEY, PERMIT, PORT, RESTRICT, TRACE and TRANSLATE statements.
Sample PROFILE TCPIP

LARGEENVELOPEPOOLSIZE 100 32K

OBEY tcpmaint maint migueld ENDOBEY

INFORM tcpmaint ENDINFORM

ASSORTEDPARMS
equalcostmultipath
ignoreredirect
ENDASSORTEDPARMS

INTERNALCLIENTPARMS
notn3270E
port 23
ENDINTERNALCLIENTPARMS

PORT
23 tcp INTCLIEN
Sample PROFILE TCPIP (cont.)

DEVICE qdio0 OSD 1e00 PRIROUTER
    LINK eth0 QDIOETHERNET qdio0 MTU 1500

DEVICE qdio1 OSD e100 NONROUTER
    LINK veth0 QDIOETHERNET qdio1 MTU 32768

DEVICE lcs0 LCS 1f08
    LINK eth1 ETHERNET 0 lcs0 MTU 1500

DEVICE ctc0 CTC 800
    LINK vctc0 CTC 1 ctc0 MTU 32760

HOME
    10.6.3.159   255.255.255.0   eth0
    192.8.12.19  255.255.255.240 veth0
    192.8.12.12  255.255.255.240 eth1
    192.4.0.1    255.255.255.252 vctc0
Sample PROFILE TCPIP (cont.)

GATEWAY
192.4.0.1 HOST 192.4.0.2 ctc0 0
DEFAULTNET 10.6.3.1 eth0 0
DEFAULTNET 192.8.12.1 eth1 0

START qdio0
START qdio1
START lcs0
START ctc0
Managing Network Interfaces Dynamically
The IFCONFIG Command

- IFCONFIG simplifies management of network interfaces
- Any user may use IFCONFIG to display network interface information
- Users in the TCP/IP server's OBEY list can use IFCONFIG make changes to the configuration without stopping the TCP/IP server
  - Add new network interfaces
  - Modify existing network interfaces
- Most device types are supported by IFCONFIG
- Command syntax is keyword driven
IFCONFIG — Gotchas

- Changes made are NOT permanent!
  - An IPL will revert to configuration in PROFILE TCPIP
  - Commands could be issued by a service machine at IPL to setup network environment

- Does not fully support dynamic routing (MPRoute)
  - Queries work fine
  - Adding or modifying interfaces requires more work on your part
Useful IFCONFIG Options

—SHOW

Displays the TCP/IP server configuration file statements that are required to make the changes specified by the command, but does NOT change the running system.

—VERBOSE

Displays any CP or NETSTAT commands that are used while changing the running system.

—FORCE

Specifies that IFCONFIG should ignore the sense data returned from the device when attempting to create a new interface.
More Useful IFCONFIG Options

- **–MDISKPW password**
  - Specifies the read password for the 'A' disk of the user issuing the IFCONFIG command

- **–TCP userid**
  - Directs the IFCONFIG command to the specified TCP/IP server

- **–ALL**
  - Displays configuration information for all interfaces, regardless of their status

- **–REMOVE (z/VM 5.3.0)**
  - Deletes an interface from the TCP/IP stack's configuration
Displaying Interfaces With IFCONFIG

ifconfig
ETH0
  inet addr: <NONE> mask: ?
  UP BROADCAST MULTICAST MTU: 1500
  vdev: 3300 type: QDIO ETHERNET portname: UNASSIGNED
  ipv4 router type: NONROUTER
  ipv6 router type: NONROUTER ipv6: ENABLED
  LAN owner: TCPIP06 name: LAN1
  cpu: 0 forwarding: ENABLED
  RX bytes: 0 TX bytes: 816
  IPv6 Addresses:
    10:0:0:0:0:0:0:1
    FE80:0:0:0:209:5700:100:3D
Ready; T=0.04/0.05 10:37:28

ifconfig eth1
ETH1
  inet addr: 10.10.0.1 mask: 255.255.255.0
  DOWN MTU: 4000
  vdev: FF00 type: HIPERS
  ipv6: DISABLED
  LAN owner: TCPIP06 name: LAN2
  cpu: 0 forwarding: ENABLED
  RX bytes: 0 TX bytes: 0
Ready; T=0.04/0.05 10:37:38
Displaying All Interfaces

ifconfig -all

ETH0  inet addr: <NONE>  mask:  ?
      UP BROADCAST MULTICAST MTU: 1500
      vdev: 3300  type: QDIO ETHERNET portname: UNASSIGNED
      ipv4 router type: NONROUTER
      ipv6 router type: NONROUTER  ipv6: ENABLED
      LAN owner: TCPIP06  name: LAN1
      cpu: 0  forwarding: ENABLED
      RX bytes: 0  TX bytes: 816
      IPv6 Addresses:
      10:0:0:0:0:0:0:1
      FE80:0:0:0:209:5700:100:3D

ETH1  inet addr: 10.10.0.1  mask: 255.255.255.0
      DOWN MTU: 4000
      vdev: FF00  type: HIPERS
      ipv6: DISABLED
      LAN owner: TCPIP06  name: LAN2
      cpu: 0  forwarding: ENABLED
      RX bytes: 0  TX bytes: 0
      Ready; T=0.05/0.07 10:39:05
Creating a Virtual HiperSockets Device

ifconfig vhsio 10.2.0.1/25 vhs 3904 system lantest mtu 8192 -v -s
* NETSTAT TCP TCPIP06 CP DEFINE LAN LANTEST OWNER SYSTEM TYPE HIPERS MFS 16K
* NETSTAT TCP TCPIP06 CP DEFINE NIC 3904 HIPERS
* NETSTAT TCP TCPIP06 CP COUPLE 3904 SYSTEM LANTEST
; Generated by <IFCONFIG vhsio 10.2.0.1/25 vhs 3904 system lantest mtu 8192 -v
; -s>
; 4 Aug 2006 10:42:21
DEVICE DEV@3904 HIPERS 3904
LINK VHSIO QDIOIP DEV@3904 MTU 8192
HOME
10.2.0.1 255.255.255.128 VHSIO
START DEV@3904
Ready; T=0.05/0.06 10:42:21
Modifying an Interface

ifconfig eth1
ETH1  inet addr: 10.10.0.1 mask: 255.255.255.0
      UP BROADCAST MULTICAST MTU: 4000
      vdev: FF00 type: HIPERS
      ipv6: DISABLED
      LAN owner: TCPIP06 name: LAN2
      cpu: 0 forwarding: ENABLED
      RX bytes: 0 TX bytes: 0
Ready: T=0.06/0.07 11:09:17

ifconfig eth1 mask 255.255.255.128 mtu 1500
Ready: T=0.12/0.14 11:09:28

ifconfig eth1
ETH1  inet addr: 10.10.0.1 mask: 255.255.255.128
      UP BROADCAST MULTICAST MTU: 1500
      vdev: FF00 type: HIPERS
      ipv6: DISABLED
      LAN owner: TCPIP06 name: LAN2
      cpu: 0 forwarding: ENABLED
      RX bytes: 0 TX bytes: 0
Ready: T=0.06/0.07 11:09:32
Starting and Stop an Interface

```
ifconfig eth1
ETH1   inet addr: 10.10.0.1 mask: 255.255.255.128
       UP BROADCAST MULTICAST MTU: 1500
       vdev: FF00 type: HIPERS
       ipv6: DISABLED
       LAN owner: TCPIP06 name: LAN2
       cpu: 0 forwarding: ENABLED
       RX bytes: 0 TX bytes: 0
Ready; T=0.06/0.07 11:12:55
ifconfig eth1 down
Ready; T=0.12/0.14 11:13:15
ifconfig eth1
ETH1   inet addr: 10.10.0.1 mask: 255.255.255.128
       DOWN MTU: 1500
       vdev: FF00 type: HIPERS
       ipv6: DISABLED
       LAN owner: TCPIP06 name: LAN2
       cpu: 0 forwarding: ENABLED
       RX bytes: 0 TX bytes: 0
Ready; T=0.06/0.07 11:13:18
ifconfig eth1 up
Ready; T=0.12/0.14 11:13:22
ifconfig eth1
ETH1   inet addr: 10.10.0.1 mask: 255.255.255.128
       UP BROADCAST MULTICAST MTU: 1500
       vdev: FF00 type: HIPERS
       ipv6: DISABLED
       LAN owner: TCPIP06 name: LAN2
       cpu: 0 forwarding: ENABLED
       RX bytes: 0 TX bytes: 0
Ready; T=0.06/0.07 11:13:23
```
IP Version 6 Support

- Support for creating and displaying IPv6 interfaces
  - Dual protocol (IPv4/IPv6) support also available
- Support for multiple IP addresses per interface
  - Displaying all IP addresses for an interface
  - Adding or deleting IP addresses to or from an interface
Creating an IPv6 Interface

ifconfig eth2 50c0:c2c1:1010::6/64 veth fe00 tcpip06 lan1 -v -s
* NETSTAT TCP TCPIP06 CP DEFINE NIC FE00 QDIO
* NETSTAT TCP TCPIP06 CP COUPLE FE00 TCPIP06 LAN1
; Generated by <IFCONFIG eth2 50c0:c2c1:1010::6/64 veth fe00 tcpip06 lan1 -v -s>
; 4 Aug 2006 10:43:26
DEVICE DEV@FE00 OSD FE00
LINK ETH2 QDIOETHERNET DEV@FE00 MTU 0 ENABLEIPV6
HOME
50C0:C2C1:1010:0:0:0:0:6
ETH2
START DEV@FE00
Ready; T=0.05/0.06 10:43:26
Adding An IP Address

```bash
ifconfig eth0
ETH0     inet addr: <NONE> mask: 255.255.255.255
UP BROADCAST MULTICAST MTU: 1500
vdev: 3300 type: QDIO ETHERNET portname: UNASSIGNED
ipv4 router type: NONROUTER
ipv6 router type: NONROUTER ipv6: ENABLED
LAN owner: TCP/IP06 name: LAN1
cpu: 0 forwarding: ENABLED
RX bytes: 752 TX bytes: 1170
IPv6 Addresses:
    10::0:0:0:0:0:0:1
    FE80::0:0:209:5700:100:3D
```

Ready; T=0.05/0.06 10:46:15

```bash
ifconfig eth0 add 50c0:c2c1:1010::1/64
Ready; T=0.10/0.12 10:46:57
```

```bash
ifconfig eth0
ETH0     inet addr: <NONE> mask: 255.255.255.255
UP BROADCAST MULTICAST MTU: 1500
vdev: 3300 type: QDIO ETHERNET portname: UNASSIGNED
ipv4 router type: NONROUTER
ipv6 router type: NONROUTER ipv6: ENABLED
LAN owner: TCP/IP06 name: LAN1
cpu: 0 forwarding: ENABLED
RX bytes: 752 TX bytes: 1266
IPv6 Addresses:
    10::0:0:0:0:0:0:1
    50c0:c2c1:1010:0:0:0:0:1
    FE80::0:0:209:5700:100:3D
```

Ready; T=0.05/0.06 10:47:01
Putting It All Together

ifconfig eth4
DTCIFC2612E Unknown interface: eth4
Ready(00012); T=0.04/0.05 10:47:58
ifconfig eth4 10.0.0.1/27 add 50c0:c2c1:1010::7/64 veth fc00 tcpip06 lan3
Ready; T=0.14/0.17 10:48:57
ifconfig eth4
ETH4 inet addr: 10.0.0.1 mask: 255.255.255.224
UP BROADCAST MULTICAST MTU: 1500
vdev: FC00 type: QDIO ETHERNET portname: UNASSIGNED
ipv4 router type: NONROUTER
ipv6 router type: NONROUTER ipv6: ENABLED
LAN owner: TCPIP06 name: LAN3
cpu: 0 forwarding: ENABLED
RX bytes: 0 TX bytes: 2376
IPv6 Addresses:
50c0:c2c1:1010::0:0:0:7
FE80:0:0:0:209:5700:100:40
Ready; T=0.06/0.07 10:49:04
Removing an Interface – z/VM 5.3.0

ifconfig eth1
ETH1 inet addr: 10.10.0.1 mask: 255.255.255.128
UP BROADCAST MULTICAST MTU: 1500
vdev: FF00 type: HIPERS
ipv6: DISABLED
LAN owner: TCPIP06 name: LAN2
cpu: 0 forwarding: ENABLED
RX bytes: 0 TX bytes: 0
Ready; T=0.02/0.02 12:56:56

netstat devlinks
VM TCP/IP Netstat Level 530

Device DEV@FF00 Type: HIPERS Status: Ready
Queue size: 0 CPU: 0 Address: FF00 Port name: UNASSIGNED
IPv4 Router Type: NonRouter Arp Query Support: Yes
Link ETH1 Type: QDIOIP Net number: 0
BytesIn: 0 BytesOut: 0
Forwarding: Enabled MTU: 1500 IPv6: Disabled
Maximum Frame Size : 16384
Broadcast Capability: Yes
Multicast Capability: Yes
Group
----- --------
224.0.0.1 1

Ready; T=0.01/0.01 12:57:01

ifconfig eth1 -remove
DTCIFC2668E -REMOVE cannot be specified for an active interface
Ready(00008); T=0.01/0.01 12:57:07
Removing an Interface – z/VM 5.3.0 (Continued)

```c
ifconfig eth1 down
Ready; T=0.03/0.04 12:59:40
ifconfig eth1 -remove
Ready; T=0.02/0.02 12:59:48
ifconfig eth1
DTCIFC2612E Unknown interface: eth1
Ready(00012); T=0.01/0.01 12:59:54
netstat devlinks
VM TCP/IP Netstat Level 530
```

Ready; T=0.01/0.01 13:00:00

**Details**

- Uses the new SIOCDINTERFACE ioctl() subcommand
  - Available to both REXX & C programs
  - Removes control block definitions and releases associated memory in the stack
Summary

- Use IPWIZARD to create your initial configuration
- Customize the three (3) configuration files as needed
  - A DTCPARMS file (<userid>, <nodeid>, or SYSTEM)
  - A TCPIP file (<userid>, <nodeid>, or PROFILE)
  - A TCPIP DATA file
- Manage network interfaces dynamically with IFCONFIG
- Communication with networking team is essential to having z/VM happily running on the network
Read More About It

- **z/VM Networking Website**

- **TCP/IP Configuration and IFCONFIG**
  - [TCP/IP Planning and Customization](http://www.vm.ibm.com/networking/)

- **IPWIZARD**
  - *Guide for Automated Installation and Service*

- **IETF RFCs**
  - [http://www.rfc-editor.org](http://www.rfc-editor.org)

- **TCP/IP Illustrated, Vol. 1**, Stevens, Addison Wesley,
  ISBN 0-201-63346-9
Contact Information

- By e-mail: Alan_Altmark@us.ibm.com
- In person: USA 607.429.3323
- Mailing lists:
  
  A number of mailing lists relevant to z/VM are available. Information on how to subscribe and view/search archives can be found at the following website:

  Of particular interest:
  IBMVM@listserv.uark.edu
  IBMTCP-L@vm.marist.edu
  LINUX-390@vm.marist.edu