ICOS for x86-based Netberg Aurora platforms

NETBERG

NETBERG

Feature Highlight

- NOS-as-an-Application Open Source Linux Packages
- Linux Interface Management
- Linux Route Table Management
- Incremental Software Upgrade
- Text-based Configuration Files
- Industry-standard CLI Interface
- Zero-Touch Provisioning, Automation, and DevOps Tools
- VxLAN Routing

NOS-as-an-Application and Linux Apps

- On x86 platforms NOS runs as a Linux service. It can be treated as an app, just like any other application in Linux.
- The recommended approach is to use native Linux user authentication methods, and then use commands (ICOS-console or ICOS-cli) to manage ICOS from the Linux shell. ICOS authentication methods are also available and can be optionally enabled.
- It is also possible to deploy additional packages as per user requirements. ICOS OS on Ubuntu Linux-based images uses APT, the Advanced Packaging Tool, which comes preinstalled on the switch.
- Use the *apt-get install <package_name>* command to download and install requested package.
- The ICOS OS image comes with a variety of utilities that help the user manage the switch. Use the *dpkg -l* command to obtain the complete list of packages installed on the switch.

Linux Interface Management

- The physical interfaces are named fpti<unit>_<slot>_<port>.

Linux	ICOS OS
fpti1_0_1	0/1
fpti1_0_2	0/2

- Standard Linux interface administration commands can be used on these interfaces – *ethtool*, *ifconfig*, *ip*.
- The port-based routing interfaces are named rt<unit>_<slot>_<port>, and typical Linux interface administration commands can be used on these interfaces.
- The VLAN routing interfaces are named rt_v<vlan-id> and typical Linux interface administration commands can be used on these interfaces
- The bonding interfaces are named bond<bond-id>.

Linux	ICOS OS Logical Interface	ICOS OS Port-Channel Name
bond1	3/1	ch1
bond2	3/2	ch2

Saving Configuration via the Linux File

When Ubuntu Linux boots, a standard configuration program “ifupdown” reads the interface configuration from the */etc/network/interfaces* file and configures the interfaces. There are four types of hooks (pre-up, up, down, and post-down) that can be configured in */etc/network/interfaces*.

- **Configuring Physical Interface Properties**
  auto fpti1_0_1
  iface fpti1_0_1 inet static
  address 0.0.0.0
  mtu 1540
  pre-up /sbin/ethtool -s fpti1_0_1 speed 10 duplex half
  pre-down /sbin/ethtool -s fpti1_0_1 speed 10 duplex half

- **Configuring a Port-based Routing Interface**
  auto rt1_0_2
  iface rt1_0_2 inet static
  mtu 200
  address 4.2.2.2
  netmask 255.255.255.0
  up ip addr add 3.3.3.3/24 dev rt1_0_2
  up ip addr add 2001::1/64 dev rt1_0_2

- **Configuring a LAG Interface**
  auto bond1
  iface bond1 inet static
  address 0.0.0.0
  mtu 2000
  up ip link set fpti1_0_1 master bond1
  up echo 4 > /sys/class/net/bond1/bonding/mode
  up echo 2 > /sys/class/net/bond1/bonding/ min_links

Etc.

Linux Route Table Management 1/2

```
root@host:/home/admin# ip route add 9.9.9.0/24 via 2.0.0.3
root@host:/home/admin# route add -net 4.4.4.0 netmask
255.255.255.0 gw 7.0.0.5
root@host:/home/admin# route -n
Kernel IP routing table
Destination Gateway Genmask Flags Metric Ref Use Iface
0.0.0.0 10.130.84.129 0.0.0.0 UG 100 0 0 eth0
10.130.84.128 0.0.0.0 255.255.255.128 U 0 0 0 eth0
7.0.0.0 0.0.0.0 255.255.255.0 U 0 0 0 rt1_0_11
2.0.0.0 0.0.0.0 255.255.255.0 U 0 0 0 rt1_0_18
4.4.4.0 7.0.0.5 255.255.255.0 UG 0 0 0 rt1_0_11
9.9.9.0 2.0.0.3 255.255.255.0 UG 0 0 0 rt1_0_18
root@host:/home/admin#
root@host:/home/admin#icos-cli
(host) #show ip route
Route Codes: R - RIP Derived, O - OSPF Derived, C - Connected, S
- Static
B - BGP Derived, IA - OSPF Inter Area
E1 - OSPF External Type 1, E2 - OSPF External Type 2
N1 - OSPF NSSA External Type 1, N2 - OSPF NSSA External Type
2
S U - Unnumbered Peer, L - Leaked Route, K - Kernel
C 7.0.0.0/24 [0/1] directly connected, 0/11
C 2.0.0.0/24 [0/1] directly connected, 0/18
K 9.9.9.0/24 [1/6493] via 2.0.0.3, 00d:00h:1m, 0/18
K 4.4.4.0/24 [1/6493] via 7.0.0.5, 00d:00h:1m, 0/11
```

```
root@host:/home/admin#
root@host:/home/admin# route add -6 3001:33:3::/64 gw 2009:1::13
root@host:/home/admin# route add -6 3001:33:3::/64 gw 2044:1::14
root@host:/home/admin# route add -6 3001:33:3::/64 gw 2044:1::15
root@host:/home/admin# route -6 -n
Kernel IPv6 routing table
Destination Next Hop Flag Met Ref Use If
2009:1::/64 :: U 256 0 6 rt1_0_11
2044:1::/64 :: U 256 0 7 rt1_0_13
3001:33:3::/64 2009:1::13 UG 1 0 0 rt1_0_11
3001:33:3::/64 2044:1::14 UG 1 0 0 rt1_0_13
3001:33:3::/64 2044:1::15 UG 1 0 0 rt1_0_13
root@host:/home/admin#icos-cli
(host) #show ipv6 route
IPv6 Routing Table - 3 entries
Codes: C - connected, S - static, 6To4 - 6to4 Route, B - BGP Derived
O - OSPF Intra, OI - OSPF Inter, OE1 - OSPF Ext 1, OE2 - OSPF Ext 2
ON1 - OSPF NSSA Ext Type 1, ON2 - OSPF NSSA Ext Type 2, K - kernel
C 2009:1::/64 [0/0]
via ::, 0/11
C 2044:1::/64 [0/0]
via ::, 0/13
K 3001:33:3::/64 [1/0]
via 2009:1::13, 00h:07m:25s, 0/11
via 2044:1::14, 00h:07m:25s, 0/13
via 2044:1::15, 00h:07m:25s, 0/13
```

Linux Route Table Management 2/2

Redistribution of Kernel Routes with BGP (ICOS OS BGP)

IPv4 Kernel Routes example:

In the originating node, Router-1, add a route to the kernel:
root@host:/home/admin # route add -net 6.5.5.0 netmask 255.255.255.0 gw 2.2.2.9

Check the *icos-cli* and to see the kernel route added to the hardware:
(Routing) (Config-router)#show ip route

Go to Router BGP Config mode and enable the redistribution of kernel routes:
(Routing) (Config)#router bgp 20
(Routing) (config-router)#redistribute kernel

Go to the peer, Router-2, and verify that this kernel route appears as a BGP route:
(Routing) #show ip route
Route Codes: R - RIP Derived, O - OSPF Derived, C - Connected, S - Static
B - BGP Derived, IA - OSPF Inter Area
E1 - OSPF External Type 1, E2 - OSPF External Type 2
N1 - OSPF NSSA External Type 1, N2 - OSPF NSSA External Type 2
S U - Unnumbered Peer, L - Leaked Route, K - Kernel
C 2.2.2.0/24 [0/0] directly connected, 0/9
B 6.5.5.0/24 [20/0] via 2.2.2.10, 00h:00m:07s, 0/9

Routing using Quagga package

Configuring IPv6 Routes in Quagga:
(localhost) #telnet 127.0.0.1 2601
…
Hello, this is Quagga (version 0.99.18).
….
Router(config)# ipv6 route 5555::/64 2001::2
Dec 18 15:41:08 localhost-1 VR_AGENT[procLOG]: rto_netlink.c(1180) 656 %% Successfully added
kernel IPv6 route 5555::/64 via 2001::2 on IntIf 25.
Exit

From kernel (Quagga configured route added to kernel):
root@localhost:/home/admin# route -6
Kernel IPv6 routing table
Destination Next Hop Flag Met Ref Use If
2001::/64 :: U 256 0 1 rt1_0_25
2001::/64 :: UAe 256 0 0 dtl0
5555::/64 2001::2 UG 1024 0 0 rt1_0_25

From ICOS OS (route gets propagated from kernel to ICOS OS, and then to hardware):
(localhost) #show ipv6 route
IPv6 Routing Table - 2 entries
Codes: C - connected, S - static, 6To4 - 6to4 Route, B - BGP Derived
O - OSPF Intra, OI - OSPF Inter, OE1 - OSPF Ext 1, OE2 - OSPF Ext 2
ON1 - OSPF NSSA Ext Type 1, ON2 - OSPF NSSA Ext Type 2, K - kernel
C 2001::/64 [0/0]
via ::, 0/25
K 5555::/64 [1/0]
via 2001::2, 00h:00m:14s, 0/25

Industry-standard CLI Interface

- ICOS OS provides a comprehensive Industry Standard-Command Line Interface (IS-CLI) to allow the administrator to enable and configure the various features supported by the switch. To access the ICOS OS CLI, use the following Linux command:
  icos-cli

Example:

icos-cli [-f<session-id>] [-s] [-h]
Command Line Options:
-f<Session-id>
Detect and close if particular session already open. The command option also creates a new session.
-s Displays existing sessions.
-h Help. Prints this text.

- There is an always-on CLI session which the user can use to login and execute commands similar to *icos-cli*. The standard output of the ICOS OS service is displayed on the **icos-console** session. The *icos-console* command is used when it is necessary to see the ICOS OS application console output that was generated before an *icos-console* session was started. Use the logout command or the **<Ctrl> + z** escape sequence to end the *icos-console* session.

At any given time, only one user can start an *icos-console* session. If a session is busy, a force option (-f) is provided to kill the existing session and start a new session.

Example:

icos-console [-r] [-h]
Command Line Options:
-r Retry to connect to pseudo terminal in the event ICOS restarts or user tries to exit. User will have to press ^z twice to completely exit.
-f Detect and close if a session is already open
-h Help. Prints this text.

ZTP and DevOps Tools

- **Zero-Touch Provisioning**

ZTP uses DHCP option 239 to specify a provisioning script URL. The provisioning script is downloaded from the web server using this URL and is executed on the switch. The provisioning script can be used to execute Linux commands and modify Linux application configuration files.

The provisioning script can be a simple Shell script. The script must be placed on a web server, and the URL of the script is used as the *provision-url* (DHCP option 239) in DHCP server configuration.

- **Automated Software and Configuration Management**

Non-disruptive Configuration
The automated software, such as Chef or Puppet, can be configured for periodic synchronization of ICOS OS configuration. The Chef/Puppet configuration file defines the interval at which the tool fetches the ICOS OS configuration file. When the software pulls the configuration for the first time, the configuration file is applied and then saved to a defined target location. On subsequent fetches, before updating the configuration, the newly downloaded configuration file is compared against the file in the target location. If the files are same, no action is taken. If the files are different, the new configuration file is applied and is saved in the target location for future comparison.

Disruptive Configuration
If the user wants the configuration to be nonvolatile, the user can configure the automated software to save the configuration as *startup-config*. To achieve this behavior, the following configurations should be made to the manifest file.

• The configuration file is marked as ‘disruptive’.
• The target is mentioned as ‘/mnt/fastpath/startup-config’

In this case, the behavior is similar to periodic configuration. The new configuration file is honored only if it is different than the current configuration. In this case, the configuration is not applied using *icos-cfg*, but saved to the ‘startup-config’, and the ICOS OS application is restarted.

Incremental Software Upgrades

To upgrade the ICOS package:

1. From the Linux shell, download the image into the switch.
2. After the image is downloaded, install it into the system.
3. Delete the download file and reboot the system.
4. Starting from ICOS release 2.1.9, a dedicated config partition is supported to store switch configuration through ONIE re-installation.

Example:

Connecting to 172.19.96.169:22... Connection established.
To escape to local shell, press 'Ctrl+Alt+]'.
Welcome to Ubuntu 14.04 LTS (GNU/Linux 3.16.0-29-generic x86_64)
** Documentation: https://help.ubuntu.com/*
Last login: Tue May 31 08:43:05 2016 from 172.19.96.126 admin@localhost:~$
admin@localhost:~$ sudo -i
[sudo] password for admin:
root@localhost:~#
root@localhost:~# cd /mnt/download/
root@localhost:/mnt/download# tftp 172.19.96.126
tftp> bin
tftp> g nba720-2.0.6-rc.deb
tftp> quit
root@localhost:/mnt/download# dpkg –i nba720-2.0.6-rc.deb
(Reading database ... 52155 files and directories currently installed.)
Preparing to unpack nba720-2.0.6-rc.deb …
icos stop/waiting
Unpacking icos-x86 (1.2.0.5.0) over (1.2.0.6.0) ...
Setting up icos-x86 (1.2.0.6.0) ...
icos start/running, process 32199
Processing triggers for ureadahead (0.100.0-16) ...
Processing triggers for libc-bin (2.19-0ubuntu6) ...
root@localhost:/mnt/download# rm nba720-2.0.6-rc.deb
root@localhost:/mnt/download# reboot

Software Features

Layer 2 features

L2 MAC address table: 288K

Link aggregation:
* 802.3ad with LACP
* Cisco EtherChannel
* Max number of group: 8
* Unicast/Multicast traffic balance
* Virtual Port Channel (MLAG)

VLAN:
* IEEE 802.1Q
* Port-Based
* Private VLAN
* Voice VLAN

Spanning Tree:
* IEEE 802.1D
* IEEE 802.1w
* IEEE 802.1s
* Spanning Tree Fast Forwarding
* Edge port (same as Fast Forwarding)
* Auto Edge
* BPDU Filter/Guard
* Loop Guard
* TCN Guard
* Root Guard

Storm Control:
* Broadcast
* Unknown Multicast
* DLF (Unknown Unicast)

IGMP Snooping:
* IGMP Snooping v1/v2/v3
* IGMP v1/v2 querier support
* IGMP Immediate Leave
* MLD Snooping
* Jumbo frame
* IEEE 802.3x Flow Control
* Q-in-Q

IPv6
* V4/V6 dual stack
* ICMPv6
* ICMPv6 redirect

* IPv6 Path MTU Discovery
* IPv6 Neighbor Discovery
* Stateless Autoconfiguration
* Manual Configuration
* DHCPv6
* SNMP over IPv6
* HTTP over IPv6
* SSH over IPv6
* IPv6 Telnet support
* IPv6 DNS resolver
* IPv6 RADIUS support
* IPv6 TACACS+ support
* IPv6 Syslog support
* IPv6 SNTP support
* IPv6 TFTP support
* Remote IPv6 ping

QoS features
* Number of priority queue: 8
* Scheduling:
** WRR
** Strict priority
** Hybrid (WRR+Strict priority)
* CoS:
** 802.1p-based CoS
** IP TOS Precedence based CoS
** IP DSCP based CoS
* DiffServ:
** 32 classes
** 13 rules per class
** No. class in policy: 64
** No. policy in class: 28
* Auto VoIP

Layer 3 Features
* Number of IP interfaces: 128
* Multinetting/CIDR
* /31 subnet support
* IP ARP
* Proxy ARP
* Local proxy ARP
* IRDP
* Static route
* ECMP
* OSFP v2/v3
* BGP v4/v6
** RFC4893
* Virtual routing and forwarding (VRF) awareness in BGP:
** BGP extended communities
** BGP route leaking
** BGP dynamic neighbors
* Multicast:
** Multicast groups
** IGMP v1/v2/v3
** MLD v1/v2
** DVMRP
** PIM-DM v4/v6
** PIM-SM v4/v6
** IGMP proxy
* VRRP
* Loopback

Software Features (cont'd)

* Routes:
** IPv4
** IPv6
** ARP entry
** ND entries
** IP IGMP/MLD
** PIM-SM/DM v4/v6
** DVMRP
* Source IP configuration
* Policy-based routing (PBR)
* IPv6 Tunneling
* IPv6 Loopback
* DHCPv6 relay
* DHCPv6 server

Security
* Static/Dynamic Port Security (MAC-based)
* 802.1x:
** Port based
** MAC based
** VLAN assignment
** Guest VLAN
** Unauthenticated VLAN
** QoS assignment
* ACL:
** L2: MAC SA/DA, CoS, EtherType
** L3: IPv4 SA/DA, subnet based
** L3: IPv6 SA/DA, flow-label, DSCP
** L4: TCP/UDP port
** Time-based ACL
** ACL counters
* RADIUS:
** Authentication
** Accounting
* TACACS+:
** Authentication
* HTTPS & SSL
* SSH 1.5/2.0
* User authentication:
** Local
** RADIUS/TACACS+
** AAA
* DoS control
* MAC filter
* IP Source Guard
* Dynamic ARP inspection
* DHCP snooping
* Control Plane Policy (CoPP)

Management
* Standard Linux shell tools
* Linux application integration
* Industry standard CLI
* CLI filtering
* Telnet/SSH
* Software/configuration upload/download using TFTP/XMODEM/HTTP/FTP/SCP/SFTP
* SNMP v1/v2c/v3
* RMON 1,2,3,9 groups
* BOOTP client/relay
* DHCP:
** Client
** Server
** Relay
** L2 option 82 relay
** L3 option 82 relay
* Event log
* DNS Client
* Utility: remote ping, traceroute
* SNTP v4
* LLDP: 802.1AB, 802.MED
* CDP
* UDLD
* Port mirroring:
** SPAN: one-to-one, many-to-one
** SPAN with ACL filter
** SPAN with VLAN
** RSPAN
* sFlow v5
* Cable test
* Email alerting
* Auto install
* RESTCONF interface
* NetSNMP

Data Center
* ONIE enabled bootloader
* FIP snooping
* Congestion Notification (CN)
* ETS
* PFC
* DCBX for PFC (CEE v1.0)
* DCBX for ETS (CEE v1.0)
* OpenFlow 1.3
* Open Ethernet Networking (OpEN) API
* Puppet/Chef support
* VXLAN
* NVGRE

NETBERG

Thank you!