カーネル スタック

- カーネル スタックについて （1 ページ）
- Guidelines and Limitations, on page 1
- ポート範囲の変更 （2 ページ）
- kstack を使用した VXLAN について （3 ページ）
- ネットデバイスのプロパティの変更 （4 ページ）

カーネル スタックについて

カーネル スタック（kstack）は、既知の Linux API を使用してルートとフロント パネル ポートを管理します。

オープン コンテナは、ゲスト シェルと同様に、ホスト ソフトウェアから分離された Linux 環境です。ホスト ソフトウェア パッケージに影響を与えることなく、その環境内でソフトウェアをインストールまたは変更できます。

Guidelines and Limitations

- Guest shell, Docker containers, and the host Bash Shell use Kernel Stack (kstack).
- The Guest Shell and the host Bash Shell start in the default network namespace. Docker containers start in the management network namespace by default.
  - Other network namespaces may be accessed by using the **setns** system call
  - The **nsenter** and **ip netns exec** utilities can be used to execute within the context of a different network namespace.
- The interface state may be read from `/proc/net/dev` or retrieved using other typical Linux utilities such as **ip**, **ifconfig**, or **netstat**. The counters are for packets that have initiated or terminated on the switch.
- **ethtool –S** may be used to get extended statistics from the net devices, which includes packets that are switched through the interface.

- Packet capture applications like **tcpdump** may be run to capture packets that are initiated from or terminated on the switch.
- There is no support for networking state changes (interface creation or deletion, IP address configuration, MTU change, and so on) from the Guest Shell.
- IPv4 and IPv6 are supported.
- Raw PF_PACKET is supported.
- Only on stack (Netstack or kstack) at a time can use well-known ports (0-15000), regardless of the network namespace.
- There is no IP connectivity between applications using Nestack and applications running kstack on the same switch. This limitation holds true regardless of whether the kstack applications are being run from the host Bash Shell or within a container.
- Applications within the Guest Shell are not allowed to send packets directly over an Ethernet out-of-band channel (EOBC) interface to communicate with the line cards or standby Sup.
- The management interface (mgmt0) is represented as eth1 in the kernel netdevices.
- Use of the VXLAN overlay interface (NVE x) is not supported for applications utilizing the kernel stack. NX-OS features, including CLI commands, are able to use this interface via netstack.

  For more information about the NVE interface, see the Cisco Nexus 9000 Series NX-OS VXLAN Configuration Guide.

ポート範囲の変更

Netstack と kstack は、それらの間のポート範囲を分割します。デフォルトのポート範囲は次のとおりです：

- Kstack：15001 ～ 58000
- Netstack：58001 ～ 65535

（注） この範囲内で、63536 ～ 65535 は NAT 用に予約されています。

（注） **nxapi use-vrf management** で構成されたポートは kstack を使用し、アクセス可能です。

手順の概要

1. [**no**] **sockets local-port-range** *start-port end-port*

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	[**no**] **sockets local-port-range** *start-port end-port*	このコマンドは、kstackのポート範囲を変更します。このコマンドは、Netstack の範囲を変更しません。

例

次に、kstack ポート範囲を設定する例を示します：

```
switch# sockets local-port-range 15001 25000
```

次のタスク

コマンドを入力した後は、次の点に注意してください：

- コマンドを入力した後には、スイッチをリロードします。
- Netstack で使用される 7000 以上のポートを、未割り当てのままにします。
- ポート範囲に抜けが生じるのを回避するには、*start-port* を 15001 に指定するか、*end-port* を 65535 に指定します。

kstack を使用した VXLAN について

NX-OS 9.2(1) 以降、VXLAN EVPN は kstack でサポートされ、サードパーティ製アプリケーションで活用できます。この機能は、Cisco Nexus 9000 ToR スイッチでサポートされています。

kstack のための VXLAN のセットアップ

VXLAN EVPN のインターフェイスまたはネットワーク名前空間にサードパーティ アプリケーションからアクセスできるようにするために、追加の構成は必要ありません。VXLAN EVPN ルートは、NX-OS VXLAN EVPN 構成に基づいてカーネルで自動的にプログラムされます。詳細については、『*Cisco Nexus 9000* シリーズ *NX-OS VXLAN* 構成ガイド』の「VXLAN BGP EVPN の構成」の章を参照してください。

kstack での VXLAN のトラブルシューティング

VXLAN の問題をトラブルシューティングする際には、次のコマンドを入力して、収集するべきいくつかの重要な情報を一覧表示してください。

```
switch(config)# show tech-support kstack
```

- **ip route show** コマンドを実行します：

```
root@switch(config)# run bash sudo su-
root@switch# ip netns exec evpn-tenant-kk1 ip route show
```

次のような出力が表示されます。

```
10.160.1.0/24 dev Vlan1601 proto kernel scope link src 10.160.1.254
10.160.1.1 dev veth1-3 proto static scope link metric 51
10.160.2.0/24 dev Vlan1602 proto kernel scope link src 10.160.2.253
127.250.250.1 dev veth1-3 proto static scope link metric 51
```

対応する VRF のすべての EVPN ルートがカーネルに存在することを確認します。

- **ip neigh show** コマンドを実行します：

```
root@switch(config)# run bash sudo su-
root@switch# ip netns exec evpn-tenant-kk1 ip neigh show
```

次のような出力が表示されます。

```
10.160.1.1 dev veth1-3 lladdr 0c:75:bd:07:b4:33 PERMANENT
127.250.250.1 devveth1-3 lladdr0c:75:bd:07:b4:33 PERMANENT
```

ネットデバイスのプロパティの変更

NX-OS 9.2(2) リリース以降、フロント チャネル ポート インターフェイスを表すネットデバイスは常に ADMIN UP 状態です。最終的に有効な状態は、リンク キャリアの状態によって決まります。

次に、NX-OS の以下のインターフェイスの例を示します。eth1/17 は **up** として表示され、eth1/1 は **down** として表示されます。

```
root@kstack-switch# sh int ethernet 1/17 brief
Eth1/17       --      eth  routed up      none                       1000(D) -


root@kstack-switch# sh int ethernet 1/1 brief
Eth1/1        --      eth  routed down    Link not connected         auto(D) -
```

次の例は、これらと同じインターフェイスを示していますが、今回は **ip link show** コマンドを使用して Bash シェルに表示しています。

```
bash-4.3# ip link show Eth1-17
49: Eth1-17: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP mode
DEFAULT group default qlen 100
    link/ether 00:42:68:58:f8:eb brd ff:ff:ff:ff:ff:ff


bash-4.3# ip link show Eth1-1
33: Eth1-1: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc pfifo_fast state DOWN
mode DEFAULT group default qlen 100
    link/ether 00:42:68:58:f8:eb brd ff:ff:ff:ff:ff:ff
```

この例では、Eth1-1 は **UP** として表示されていますが、**NO-CARRIER** および **state DOWN** として表示されています。

次の例は、これらと同じインターフェイスを示していますが、今回は **ifconfig** コマンドを使用して Bash シェルに表示しています。

```
bash-4.3# ifconfig Eth1-17
Eth1-17   Link encap:Ethernet  HWaddr 00:42:68:58:f8:eb
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:7388 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          RX bytes:0 (0.0 B)  TX bytes:1869164 (1.7 MiB)


bash-4.3# ifconfig Eth1-1
Eth1-1    Link encap:Ethernet  HWaddr 00:42:68:58:f8:eb
          inet addr:99.1.1.1  Bcast:99.1.1.255  Mask:255.255.255.0
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
```

ifconfig コマンドの出力には、さまざまな情報が表示されますが、**RUNNING** キーワードを使用して最終的な状態を示すことができます。デフォルトでは、すべてのネットデバイスにキーワード **UP** が表示されます。これは、カーネル内のネットデバイスの ADMIN 状態を表しています。

NX-OS 9.2(2) リリースの変更点の一部に、次のものがあります。

- **ネットデバイスの IPv4 アドレス**：NX-OS 9.2(2) リリースより前は、NX-OS の対応するインターフェイスが **DOWN** 状態であっても、IPv4 アドレスはカーネルのネットデバイスに組み込まれていました。NX-OS 9.2(2) リリース以降、IPv4 アドレスは、インターフェイスが **UP** 状態の場合にのみカーネル空間に組み込まれます。いったん組み込まれると、インターフェイスが **DOWN** になっても、IPv4 アドレスはカーネル内のネットデバイスに残ります。次の CLI コマンドを入力して、NX-OS インターフェイスから IP アドレスを明示的に削除した後にのみ削除されます。

  ```
  Interface Eth1/1
              no ip address IP-address
  ```

- **ネットデバイスの IPv6 アドレス**：NX-OS 9.2(2) リリースより前は、インターフェイスが **DOWN** になると、IPv6 アドレスはカーネルのネットデバイスからフラッシュされていました。NX-OS 9.2(2) リリース以降、ネットデバイスは常に管理 **UP** 状態であるため、インターフェイスがダウンしても、IPv6 アドレスはカーネルからフラッシュされません。

翻訳について
このドキュメントは、米国シスコ発行ドキュメントの参考和訳です。リンク情報につきましては、日本語版掲載時点で、英語版にアップデートがあり、リンク先のページが移動/変更されている場合がありますことをご了承ください。あくまでも参考和訳となりますので、正式な内容については米国サイトのドキュメントを参照ください。