

# MidoNet トラブルシューティング ガイド

2015.06-SNAPSHOT (2015-10-15 05:20 UTC)

DRAFT



## 目次

はじめに .....	iv
表記規則 .....	iv
1. 全体アプローチ .....	1
アンダーレイネットワーク .....	1
オーバーレイネットワーク .....	3
トポロジースミュレーション .....	5
仮想トポロジー .....	5
2. 共通トピック .....	6
MidoNet エージェント .....	6
MidoNet API .....	6
ボーダーゲートウェイプロトコル (BGP) .....	6
ZooKeeper .....	8
VM の相互接続 .....	8
3. ツールとコマンド .....	10
midonet-cli .....	10
mm-dpctl .....	10
mm-trace .....	10
ip .....	11
4. ディレクトリーとファイル .....	12
Cassandra .....	12
MidoNet Agent .....	12
MidoNet API .....	12
Quagga (BGPD) .....	12
ZooKeeper .....	12
5. プロセス .....	14

# はじめに

## 表記規則

MidoNet のドキュメントは、いくつかの植字の表記方法を採用しています。

### 注意

注意には以下の種類があります。



#### 注記

簡単なヒントや備忘録です。



#### 重要

続行する前に注意する必要があるものです。



#### 警告

データ損失やセキュリティ問題のリスクに関する致命的な情報です。

## コマンドプロンプト

### \$ プロンプト

root ユーザーを含むすべてのユーザーが、\$ プロンプトから始まるコマンドを実行できます。

### # プロンプト

root ユーザーは、# プロンプトから始まるコマンドを実行する必要があります。利用可能ならば、これらを実行するために、sudo コマンドを使用できます。







```
Datapath name : midonet
Datapath index : 11
Datapath Stats:
  Flows :1340066
  Hits :1111802509
  Lost :0
  Misses:17302163
Port #0 "midonet" Internal Stats{rxPackets=0, txPackets=0, rxBytes=0, txBytes=0,
rxErrors=0, txErrors=0, rxDropped=0, txDropped=0}
Port #1 "tngre-overlay" Gre Stats{rxPackets=508157678, txPackets=398704120,
rxBytes=291245619484, txBytes=318474308439, rxErrors=0, txErrors=0, rxDropped=0,
txDropped=0}
Port #2 "tnvxlan-overlay" VXLAN Stats{rxPackets=0, txPackets=0, rxBytes=0, txBytes=0,
rxErrors=0, txErrors=0, rxDropped=0, txDropped=0}
Port #3 "tnvxlan-vtep" VXLAN Stats{rxPackets=0, txPackets=0, rxBytes=0, txBytes=0,
rxErrors=0, txErrors=0, rxDropped=0, txDropped=0}
Port #4 "tapa0164c42-dd" NetDev Stats{rxPackets=389426272, txPackets=342761506,
rxBytes=1128206548338, txBytes=241007949600, rxErrors=0, txErrors=0, rxDropped=0,
txDropped=0}
Port #5 "tap19ccc069-f1" NetDev Stats{rxPackets=0, txPackets=54640, rxBytes=0,
txBytes=2347034, rxErrors=0, txErrors=0, rxDropped=0, txDropped=0}
Port #6 "tape3055fc6-cc" NetDev Stats{rxPackets=21375, txPackets=42911, rxBytes=3573207,
txBytes=4607633, rxErrors=0, txErrors=0, rxDropped=0, txDropped=0}
```

```
# mm-dpctl --dump-dp midonet
1340149 flows
  Flow:
    match keys:
      Tunnel{tun_id=4360, ipv4_src=10.11.0.16, ipv4_dst=10.11.0.15, tun_flag=0,
      ipv4_tos=0, ipv4_ttl=-3}
      InPort{1}
      Ethernet{src=02:13:38:97:08:f3, dst=fa:16:3f:92:53:60}
      EtherType{0x800}
      KeyIPv4{src=8.8.8.8, dst=10.17.3.14, proto=17, tos=0, ttl=55, frag=0}
      UDP{src=53, dst=56975}
    actions:
      Output{port=21}
[...]
```

```
# mm-ctl --list-hosts
Host: id=17ef018f-de8b-431b-89f0-b5472f176769
  name=hostname
  isAlive=true
  addresses:
  vport-host-if-bindings:
    VirtualPortMapping{virtualPortId=ac0c2557-9fa0-4009-9e18-dc62ea65052a,
localDeviceName=' tapac0c2557-9f'}
    VirtualPortMapping{virtualPortId=c37d8bf2-d008-464e-a688-0627f2da342f,
localDeviceName=' f58b0880_MN_dp'}
    VirtualPortMapping{virtualPortId=7aa08012-d06c-4c78-ae8-1fff7c063fed,
localDeviceName=' tap7aa08012-d0'}
    VirtualPortMapping{virtualPortId=5aa6a752-57f2-4749-b160-9e632e0a16bb,
localDeviceName=' f58b0880_MN_dp'}
[...]
```

## MTU

アンダーレイネットワークのフラグメンテーションを避けるために、VMインスタンスのMTUは、トンネルプロトコルのオーバーヘッドに対応する必要があります。

この調整されたMTUは、DHCPを通じて、MidoNetに酔って自動的にアドバタイズされます。しかし、これは、VMで使われているオペレーティングシステムによっては、適応されない場合があります。



Underlay MTU	Tunnel Protocol	Protocol Overhead	VM' s MTU
1500 bytes	VxLAN	50 bytes	1450 bytes
1500 bytes	GRE	46 bytes	1455 bytes
9000 bytes	VxLAN	50 bytes	8950 bytes
9000 bytes	GRE	46 bytes	8955 bytes

Service	Port
ZooKeeper	2181
Cassandra	9042

プロトコル (ICMP, SSH, HTTP など) と使われているポートに関する適切なルールが存在しているかをチェックしてください。





8

VM のポート UUID を特定するには、Horizon で VM のネットワークに移動し、ポートリストで VM の内部 IP を検索します。そこから、次の例のように TAP インターフェイス名を作成します。

ポート UUID: 7aa08012-d06c-4c78-ae8-1fff7c063fed

TAP インターフェイス: tap7aa08012-d0

## インターフェースでトラフィックを検証する

コンピューティングホストの VM の TAP インターフェイスで tcpdump を使用すれば、ゲストホストにログインせずに VM の仮想 NIC でトラフィックが確認されているかどうかを検証できます。

```
# tcpdump -n -i tap7aa08012-d0
```

TAP インターフェイスでパケットカウンターを監視します。

```
# watch -d ip -s link show tap7aa08012-d0
```

## 第3章 ツールとコマンド

### 目次

midonet-cli .....	10
mm-dpctl .....	10
mm-trace .....	10
ip .....	11

このセクションは、便利なツールとコマンドに関する概要を提供します。

### midonet-cli

midonet-cli コマンドは、MidoNet APIに接続し、python-midonet-client パッケージをインストールしているホスト上で走らせることができます。

### mm-dpctl

mm-dpctl コマンドは、どのMidoNetエージェントでも走らせることができ、現在のフローなどデータパス情報を表示することができます。

利用可能なオプション:

```
$ mm-dpctl
usage: mm-dpctl
  --add-dp <arg>      新しいデータパスを追加
  --delete-dp <arg>   データパスを削除
  --dump-dp <arg>     与えられたデータパスにインストールされている全てのフローを表示
  --list-dps           インストールされているデータパスをリスト化
  --show-dp <arg>     与えられたのデータパスに関連する情報を表示
  --timeout <arg>     何秒でタイムアウトするかを特定。特定した時間内にプログラムが結果が帰ってこない場合は、停止してエラーコードを戻します。
```

例題:

```
$ mm-dpctl --show-dp midonet # shows datapath and interfaces
$ mm-dpctl --dump-dp midonet # shows current flows
```

### mm-trace

a mm-trace コマンドは、MidoManエージェントがある特定のトラフィックフローをキャプチャして、シミュレーションの各ステージのログを取ることができます。

この設定は、MidoMan再起動の際に、常に同じではありません。

アウトプットは /var/log/midolman/mm-trace.log ファイルに書き込まれます。

利用可能なオプション:

```
$ mm-trace --help
-h, --host <arg>   Host (default = localhost)
-p, --port <arg>   JMX port (default = 7200)
--help             ヘルプメッセージを表示
```

Subcommand: add - トレーシングマッチのパケットを追加



## 第4章 ディレクトリーとファイル

### 目次

Cassandra .....	12
MidoNet Agent .....	12
MidoNet API .....	12
Quagga (BGPD) .....	12
ZooKeeper .....	12

このセクションは、よく使う設定とログファイルの概要を提供します。

ファイル名とパスは、使っているオペレーティングシステムとOpenStackのディストリビューションによって、微妙に異なりますので注意してください。

### Cassandra

File	Type
/etc/cassandra/conf/cassandra.yaml	CONF
/var/log/cassandra/cassandra.log	LOG

### MidoNet Agent

File	Type
/etc/midolman/midolman-akka.conf	CONF
/etc/midolman/midolman.conf	CONF
/etc/midolman/midolman-env.sh	CONF
/var/log/midolman/midolman.event.log	LOG
/var/log/midolman/midolman.log	LOG
/var/log/midolman/mm-trace.log	LOG
/var/log/midolman/upstart-stderr.log	LOG

### MidoNet API

File	Type
/usr/share/midonet-api/WEB-INF/web.xml	CONF
/var/log/tomcat/catalina.out	LOG
/var/log/tomcat/midonet-api.log	LOG

### Quagga (BGPD)

File	Type
/var/log/quagga/bgpd.log	LOG

### ZooKeeper

File	Type
/etc/zookeeper/zoo.cfg	CONF



File	Type
/var/log/zookeeper/zookeeper.out	LOG

## 第5章 プロセス

このセクションは、共通プロセスを提供します。

使っているオペレーション・システムとOpenStackディストリビューションによって、名前とファイルパスが異なります。

プログラ	プロセス
Cassandra	java [...] org.apache.cassandra.service.CassandraDaemon
MidoNet Agent	java [...] org.midonet.midolman.Midolman
MidoNet Agent (Watchdog)	/usr/bin/python /usr/bin/wdog [...] org.midonet.midolman.Midolman
MidoNet API (Tomcat)	java [...] org.apache.catalina.startup.Bootstrap
ZooKeeper	java [...] org.apache.zookeeper.server.quorum.QuorumPeerMain