トラブルシューティング

この章では、システムの動作中に発生する可能性のある問題をトラブルシューティングするために、システムのコマンドラインインターフェイス（CLI）を使用する方法について説明します。

ネットワーク接続の確認

システムでサポートされているコマンドは複数あり、ネットワーク接続の確認やトラブルシュートを行うことができます。ネットワーク接続は、システムインターフェイスとポートが設定され、バインドされている場合にのみテストできることに注意してください。

このセクションで指定するコマンドは、コンテキストごとに発行する必要があります。コンテキストは、他のコンテキストとは独立して動作するバーチャル プライベート ネットワーク（VPN）のように機能します。あるコンテキストで設定されたポート、インターフェイス、およびルートは、追加の設定なしで別のコンテキストからテストすることはできません。

コンテキストを切り替えるには、Exec モードのルートプロンプトで次のコマンドを入力します。

```
[local]host_name# context context_name
```

context_name は、切り替え先のコンテキストの名前です。次のプロンプトが表示されます。

```
[context_name]host_name#
```

ping コマンド または ping6 コマンドの使用

ping または **ping6** コマンドは、応答間でデータパケットを受け渡し、測定することによって、ネットワーク内のリモートノードとシステムが通信できることを確認します。このコマンド

は、ネットワークルーティングを確認したり、リモートノードが IP レイヤで応答できるかどうかを確認するのに役立ちます。

VPC-DI の展開では、Exec モードコマンドの **system ping** を使用して、VPC-DI VM 間の内部ネットワーク（DI ネットワーク）接続を確認します。詳細については、『*Command Line Interface Reference*』を参照してください。

構文

ping のコマンドシンタックスは、次のとおりです。

ping *host_ipv4_address* **[count** *num_packets* **] [flood] [pattern** *packet_pattern* **] [size** *octet_count* **] [src {** *src_host_name* **|** *src_host_ipv4_address* **}] [vrf** *vrf_nam* **]**

ping6 *host_ipv6_address* **[count** *num_packets* **] [flood][pattern** *packet_pattern* **] [size** *octet_count* **] [src {** *src_host_name* **|** *src_host_ipv6_address* **}] [vrf** *vrf_nam* **]**

上記のコマンドの詳細については、『*Command Line Interface Reference*』の「*Exec Mode Commands*」の章を参照してください。

次に、成功した **ping**（IPV4）の応答の例を示します。

```
PING 209.165.200.227 (209.165.200.227): 56 data bytes
64 bytes from 209.165.200.227: icmp_seq=0 ttl=255 time=0.4 ms
64 bytes from 209.165.200.227: icmp_seq=1 ttl=255 time=0.2 ms
64 bytes from 209.165.200.227: icmp_seq=2 ttl=255 time=0.2 ms
64 bytes from 209.165.200.227: icmp_seq=3 ttl=255 time=0.2 ms
64 bytes from 209.165.200.227: icmp_seq=4 ttl=255 time=0.2 ms
--- 209.165.200.227 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 0.2/0.2/0.4 ms
```

トラブルシューティング

ターゲットから応答を受信しない場合は、次のトラブルシューティング手順に従ってください。

- 正しい IP アドレスが入力されていることを確認します。
- 同じネットワーク上の別のデバイスに ping を試行します。ping が成功した場合は、システム設定が正しいと思われます。pingを実行しようとしているデバイスの電源が入っており、正常に機能していることを確認します。
- ポートが動作していることを確認します。
- コンテキスト内のポートとインターフェイスの設定が正しいことを確認します。
- 設定が正しく、pingを試行しているデバイスにアクセスできる場合は、そのデバイスからシステムに ping を実行します。
- まだ応答がない場合は、パケットがネットワークデバイスによって破棄されている可能性があります。この章で説明されている **traceroute** コマンドまたは **traceroute6** コマンドおよび **show ip static-route** コマンドを使用して、この問題のトラブルシュートを行います。

traceroute または traceroute6 コマンドの使用

traceroute コマンドまたは **traceroute6** コマンドは、指定されたホストに送信されるルートデータに関する情報を収集します。これは、ネットワーク上の重大なパケット遅延またはパケット損失の原因を特定するために使用できる、便利なトラブルシューティング コマンドです。また、このコマンドは、ネットワークを介したデータのルーティングのボトルボトルネックを識別するためにも使用できます。

traceroute：IPv4

次に、**traceroute** コマンドシンタックスを示します。

traceroute { *host_name* **|** *host_ipv4_address* **} [count** *packets* **] [df] [maxttl** *max_ttl* **] [minttl** *min_ttl* **] [port** *port_number* **] [size** *octet_count* **] [src {** *src_host_name* **|** *src_host_ipv4_address* **}] [timeout** *seconds* **] [vrf** *vrf_nam* **]**

上記のコマンドの詳細については、『*Command Line Interface Reference*』の「*Exec Mode Commands*」の章を参照してください。

次に、出力例を示します。

```
traceroute to 209.165.200.227 (209.165.200.227), 30 hops max, 40 byte packets
 1  209.165.200.227 (209.165.200.227)  0.446 ms  0.235 ms  0.178 ms
```

traceroute6：IPv6

次に、**traceroute6** コマンドのシンタックスを示します。

traceroute6 { *host_name* **|** *host_ipv6_address* **} [count** *packets* **] [maxttl** *max_ttl* **] [port** *port_number* **] [size** *octet_count* **] [src {** *src_host_name* **|** *src_host_ipv6_address* **}] [timeout** *seconds* **] [vrf** *vrf_nam* **]**

上記のコマンドの詳細については、『*Command Line Interface Reference*』の「*Exec Mode Commands*」の章を参照してください。

次に、出力例を示します。

```
traceroute6 to 2001:4A2B::1f3F (2001:4A2B::1f3F), 30 hops max, 40 byte packets
 1  2001:4A2B::1f3F (2001:4A2B::1f3F)  0.446 ms  0.235 ms 0.178 ms
```

IP ルートの表示

このシステムには、特定のノードへのルート情報またはコンテキスト全体を表示するメカニズムが備わっています。この情報を使用して、ネットワーク接続を確認し、ネットワーク接続の効率を高めることができます。コマンドの構文は、次のとおりです。

show ip route [*route_ip_address* **]**
show ipv6 route [*route_ipv6_address* **]]**

上記のコマンドの詳細については、『*Command Line Interface Reference*』の「*Exec Mode show Commands*」の章を参照してください。

キーワードを指定しなかった場合は、コンテキストのルーティングテーブル内のすべての IP ルートが表示されます。

次に、コンテキスト IPv4 ルーティングテーブルが示されているこのコマンドの出力例を示します。

```
"*" indicates the Best or Used route.
  Destination     Nexthop     Protocol    Prec    Cost    Interface
*0.0.0.0/0        10.0.4.1    static      0       0       SPIO1
*10.0.4.0/24      0.0.0.0     kernel      0       0       SPIO1
*10.0.4.0/32      0.0.0.0     kernel      0       0       SPIO1
*10.0.4.3/32      0.0.0.0     kernel      0       0       SPIO1
*10.0.4.255/32    0.0.0.0     kernel      0       0       SPIO1
```

アドレス解決プロトコルテーブルの表示

システムは、特定のノードまたはコンテキスト全体に対して、Address Resolution Protocol（ARP）のテーブル情報を表示するメカニズムを提供します。この情報は、システムが ARP パケットを送信したときに、他のネットワークノードから有効な応答を受信したことを確認するために使用できます。

`[local]host_name#` **show ip arp [** *arp_ip_address* **]**

arp_ip_address は、ARP 情報を表示する特定のネットワークノードを指定します。このアドレスは、IPv4 ドット付き 10 進表記または IPv6 コロン区切り 16 進表記で入力できます。このキーワードが指定されていない場合は、コンテキストの ARP テーブル内のすべてのエントリが表示されます。

重要 VPN マネージャを再起動すると、カーネルからすべてのインターフェイスが削除されます。これにより、すべての ARP エントリが削除されます。ただし、NPU では、トラフィックが中断されないように、すべての ARP エントリが引き続き保持されます。ユーザーの観点から、このコマンドは NPU ではなくカーネルから情報を収集するため、**show ip arp** が破損しています。

次に、コンテキストの ARP テーブルを表示するこのコマンドの出力例を示します。

```
Flags codes:
C - Completed, M - Permanent, P - Published, ! - Not answered
T - has requested trailers
Address         Link Type    Link Address          Flags    Mask Interface
10.0.4.240      ether        00:05:47:02:20:20     C        MIO1
10.0.4.7        ether        00:05:47:02:03:36     C        MIO1
10.0.4.1        ether        00:01:30:F2:7F:00     C        MIO1
```

システム診断ユーティリティの使用

システムには、設定のトラブルシューティングや確認の際に役立つプロトコルモニターとテストユーティリティが備わっています。これらのユーティリティによって生成される情報は、ソフトウェアやネットワーク設定の問題の根本原因を特定するのに便利です。

この項では、これらのユーティリティの使用方法について説明します。

重要 この章で説明する診断ユーティリティは、オペレータ以上の権限を持つ管理者のみが実行できます。

モニターユーティリティの使用

システムにはトラブルシューティングを目的としたプロトコル モニタリング ユーティリティが用意されています。このツールは、特定のサブスクライバセッションか、または処理中のすべてのセッションのプロトコル情報を表示します。

注意 モニターツールによって、セッションの処理遅延やデータ損失が発生する場合があります。したがって、トラブルシューティングを行う場合にのみ使用してください。

プロトコルモニターの使用

プロトコルモニターには、現在処理中のすべてのセッションの情報が表示されます。モニター対象のプロトコルの数と進行中のセッション数に応じて、大量のデータが生成されます。生成されたすべての情報をキャプチャするには、端末クライアントでロギングを有効にすることを強くお勧めします。

monitor protocol コマンドおよび **monitor subscriber** コマンドの PCAP 機能を有効にするには、パケットキャプチャ（PCAP）トレース も参照してください。

プロトコル モニタリング ツールを起動して設定するには、次の手順に従います。

ステップ 1 **monitor protocol** コマンドを入力して、Exec モードでプロトコルモニターを起動します。

```
[local]host_name# monitor protocol
```

現在使用可能なすべてのプロトコル（それぞれに割り当てられた番号を持つ）が一覧表示された出力が表示されます。

ステップ 2 *Select*: プロンプトで関連付けられた番号を入力して、モニターするプロトコルを選択します。選択したプロトコルの横に右矢印（>）が表示されます。

ステップ 3 必要に応じてステップ *2* を繰り返して、複数のプロトコルを選択します。

ステップ 4 **B** を押して、プロトコルモニターを開始します。

```
WARNING!!! You have selected options that can DISRUPT USER SERVICE
Existing CALLS MAY BE DROPPED and/or new CALLS MAY FAIL!!!
(Under heavy call load, some debugging output may not be displayed)
Proceed? - Select (Y)es or (N)o
```

ステップ 5 **Y** を入力してモニターを続行するか、**N** を入力して前のメニューに戻ります。

```
C - Control Events    (ON )
D - Data Events               (ON )
```

```
E - EventID Info              (ON )
H - Display ethernet          (ON )
I - Inbound Events            (ON )
O - Outbound Events           (ON )
S - Sender Info               (OFF)
T - Timestamps                (ON )
X - PDU Hexdump               (OFF)
A - PDU Hex/Ascii             (OFF)
+/- Verbosity Level           (    1)
L - Limit Context             (OFF)
M - Match Newcalls            (ON )
R - RADIUS Dict               (no-override)
G - GTPP Dict                 (no-override)
Y - Multi-Call Trace          ((OFF))
      (Q)uit,      <ESC> Prev Menu,      <SPACE> Pause,      <ENTER> Re-Display Options
```

ステップ 6 モニターによって表示される情報の量を設定します。オプションを有効化または無効化するには、そのオプションに関連付けられている文字（C、D、E など）を入力します。冗長性を向上または低下させるには、プラス（+）またはマイナス（-）キーを使用します。

各オプションの右側には、[ON (enabled)] または [OFF (disabled)] の現在の状態が表示されます。

ステップ 7 **[Enter]** キーを押して画面を更新し、モニタリングを開始します。

モニターは、無効になるまでアクティブのままになります。プロトコルモニターを終了してプロンプトに戻るには、**q** を押します。

特定サブスクライバのプロトコルモニターの使用

プロトコルモニターは、現在処理中の特定のサブスクライバセッションの情報を表示するために使用できます。モニター対象のプロトコルの数と進行中のセッション数に応じて、大量のデータが生成されます。生成されたすべての情報をキャプチャするには、端末クライアントでロギングを有効にすることを強くお勧めします。

特定のサブスクライバセッションのプロトコルモニタリング ツールを起動して設定するには、この項の手順に従います。

ステップ 1 Exec モードからセッション固有のプロトコルモニターを起動するには、**monitor subscriber** コマンドを入力します。

```
[local]host_name# monitor subscriber { callid | imei | imsi | ipaddr | ipv6addr |
msid | msisdn | next-call | pcf | peer-fa | peer-lac | sgsn-address | type |
username }
```

ステップ 2 適切なキーワードを入力して、モニターが使用するメソッドを指定します。

ステップ 3 その他のオプションを選択したり、選択したキーワードに適切な情報を入力したりします。

モニターの起動時に、指定された基準に一致するセッションが処理されなかった場合は、使用可能なモニタリングオプションの画面が表示されます。

ステップ 4 モニターによって表示される情報の量を設定します。オプションを有効または無効にするには、そのオプションに関連付けられている文字または 2 桁の数字（C、D、E、11、12 など）を入力します。冗長性を向上または低下させるには、プラス（+）またはマイナス（-）キーを使用します。

各オプションの右側には、[ON (enabled)] または [OFF (disabled)] の現在の状態が表示されます。

マルチコールトレースを実行するためのオプション **Y** は、GGSN での使用に対してのみサポートされています。

ステップ 5 必要に応じてステップ *6* を繰り返して、複数のプロトコルを有効または無効にします。

ステップ 6 **Enter** を押して画面を更新し、モニタリングを開始します

次に、*user2@aaa* という名前のサブスクライバに対するモニターの出力例の一部を示します。デフォルトのプロトコルがモニターされました。

```
------------------------------------------------------------------------------
Incoming Call:
------------------------------------------------------------------------------
    MSID: 0000012345 Callid: 002dc6c2
    Username: user2@aaa SessionType: unknown
    Status: Active Service Name: xxx1
    Src Context: source Dest Context:
------------------------------------------------------------------------------

<<<<OUTBOUND 10:02:35:415 Eventid:25001(0)
PPP Tx PDU (9)
PAP 9: Auth-Ack(1), Msg=

<<<<OUTBOUND 10:02:35:416 Eventid:25001(0)
PPP Tx PDU (14)
IPCP 14: Conf-Req(1), IP-Addr=192.168.250.70

<<<<OUTBOUND 10:02:35:416 Eventid:25001(0)
PPP Tx PDU (27)
CCP 27: Conf-Req(1), MPPC, Stac-LZS, Deflate, MVRCA

INBOUND>>>>> 10:02:35:517 Eventid:25000(0)
PPP Rx PDU (30)
IPCP 30: Conf-Req(1), IP-Comp VJ-Comp, IP-Addr=0.0.0.0, Pri-DNS=0.0.0.0,
Sec-DNS=0.0.0.0

<<<<OUTBOUND 10:02:35:517 Eventid:25001(0)
PPP Tx PDU (26)
IPCP 26: Conf-Rej(1), IP-Comp VJ-Comp, Pri-DNS=0.0.0.0, Sec-DNS=0.0.0.0

INBOUND>>>>> 10:02:35:517 Eventid:25000(0)
PPP Rx PDU (12)
IPCP 12: Conf-Ack(1), IP-Addr=192.168.250.70

INBOUND>>>>> 10:02:35:518 Eventid:25000(0)
PPP Rx PDU (31)
LCP 31: Prot-Rej(1), Rejected-Protocol=CCP (0x80fd)

INBOUND>>>>> 10:02:35:518 Eventid:25000(0)
PPP Rx PDU (12)
IPCP 12: Conf-Req(2), IP-Addr=0.0.0.0

<<<<OUTBOUND 10:02:35:518 Eventid:25001(0)
PPP Tx PDU (14)
IPCP 14: Conf-Nak(2), IP-Addr=192.168.250.87

INBOUND>>>>> 10:02:35:519 Eventid:25000(0)
PPP Rx PDU (12)
IPCP 12: Conf-Req(3), IP-Addr=192.168.250.87
```

モニターは、無効になるまでアクティブのままになります。プロトコルモニターを終了してプロンプトに戻るには、**q** を押します。

SSD の生成

SSD は、Exec モードの **show support details** コマンドが実行されたときの出力のインスタンスです。トラブルシューティングのために役立つシステム情報の包括的なリストが表示されます。ほとんどの場合、このコマンドの出力はテクニカルアシスタンスセンター（TAC）によって要求されます。

SSD 出力の .tar ファイルは、ローカルまたはリモートの場所（URL）にリダイレクトできます。

.tar ファイルには次のものが含まれます。

- **support_summary**：サポートの詳細情報を含む ASCII テキストファイル。
- **information.minicores.tar**：システム上ので検出された minicore ファイルを含む .tar ファイル。Minicoreファイルには、一部のイベント中にキャプチャされるメモリコアダンプが含まれています。これらのコアダンプは、イベントに関する特定のメモリの場所とその他の情報を提供します。この情報はテクニカルサポートチームにとって、推定原因とともにイベントが発生した場所とタイミングを特定するために役立ちます。

show support details コマンドには、他の方法ではユーザーがアクセスできない情報が含まれていますが、TAC による問題の迅速な解決に役立ちます。

重要 大規模なコンフィギュレーション ファイルを持つプラットフォームでは、SSD を完了するまでに最大で30分かかる場合があります。**show support details** コマンドを実行すると、システムリソースが消費され、トラフィックのスループットが低下する可能性があります。

オペレータが **show support details** コマンドを入力したときに SSD が進行中である場合、StarOS は SSD がすでに進行中であることを示す警告メッセージで応答します。ユーザーは後で再試行する必要があります。オペレータは、一度に 1 つの SSD インスタンスだけを実行するように制限されています。

show support details コマンドには、特定のタイプの情報だけを報告するように SSD をターゲットにできるオプションのキーワードがあります。これらのキーワードにより、SSD の生成に必要な時間を短縮できます。

show support details コマンドの詳細については、『*Command Line Interface Reference*』の「*Exec Mode Show Commands (Q-S)*」の章を参照してください。

サポートデータコレクターの設定と使用

サポートデータを収集するタスクは、record collector と呼ばれるバックグラウンド CLI タスクによって実行されます。管理者は、CLI を介して Support Data Collector（SDC）を設定し、コマンドを定期的に実行します。レコードコレクタは常にバックグラウンドで実行され、収集レコードがあるかどうかを確認します。

サポートデータを収集する時間になると、スケジューラは設定された CLI コマンドのシーケンスを実行し、その結果をハードディスク上の gunzipped（gz）ファイルに保存します。このファイルは SDR（サポートデータレコード）と呼ばれ、その時点でのシステム全体の状態のスナップショットを表します。

テクニカル アシスタンス センター（TAC）担当者およびローカル管理者は、SDR をオンラインで、またはシステムから転送して確認することができます。また、コレクタの状態の情報を調査する場合もあります。

SDC 機能の詳細については、「サポートデータコレクター」の章を参照してください。

ハイパーバイザの強制再起動

ハイパーバイザは、仮想ウォッチドッグデバイスを提供するために必要です。基盤となるホスト OS の障害が原因で、StarOS がこのウォッチドッグデバイスのサービスを停止した場合、ハイパーバイザは VM を強制的に再起動する必要があります。

VPC-DI インスタンスである仮想化されたシャーシでは、ホストの障害に起因する別の VM の再起動を CF がリモートで行う方法はありません。正常な状態では、CF は VM 上で実行されている StarOS にメッセージを送信することで、別の VM をリモートで再起動できます。ただし、ホストが障害状態の場合、これは機能しない可能性があり、ハイパーバイザのウォッチドッグはフォールバックメカニズムとして機能する必要があります。

KVM では、「--watchdog i6300esb」コマンドライン引数を使用して、仮想ウォッチドッグデバイスを指定できます。

VMware は、独自のウォッチドッグメカニズムを提供します。詳細については、VMware のドキュメントを参照してください。

表 *1* : 障害状況および再起動方法

障害状況	再起動方法	リカバリ	注意
card reboot *x*	CF は、リモート StarOS にローカルリブートを実行するように指示します。		
クリティカルタスクの失敗	ハイパーバイザ ウォッチドッグ	ハイパーバイザによる VM の再起動	StarOS ドッグ

障害状況	再起動方法	リカバリ	注意
カーネルのハング/クラッシュ	カーネルの再起動またはハイパーバイザのウォッチドッグ	ハイパーバイザによる VM の再起動	
ホストの障害	ハイパーバイザ HA	CF は、リモート StarOS にローカルリブートを実行するように指示します。	参照先：V

手動によるスタンバイ CF へのスイッチング

手動によりアクティブ CF「カード」を強制的に冗長ペアのスタンバイ CF に切り替えることができます。

次の Exec モード CLI は、アクティブ CF をスタンバイ CF に切り替えます。

card switch to *slot*

注：

- このコマンドは、VPC-DI インスタンス内の冗長 CF VM に対してのみ機能します。
- スイッチオーバーを開始するには、アクティブ CF にログインする必要があります。
- スロット には 1 または 2 を指定できます。

スイッチオーバープロセスの流れは、次のとおりです。

1. アクティブ CF はスタンバイモードに移行します。
2. スタンバイ CF はアクティブモードに移行します。
3. リロードは、以前のアクティブ CF（現在はスタンバイ CF）で開始されます。

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。