CHAPTER 21

RMON の設定

この章では、ONS 15454 SONET/SDH の ML シリーズ カード上で Remote Network Monitoring（RMON）を設定する方法について説明します。

RMON は、RMON 準拠のコンソール システムとネットワーク プローブ間で交換可能な一連の統計情報と機能を定義した標準モニタリング仕様です。RMON は総合的なネットワーク障害診断、プランニング、パフォーマンス調整に関する情報を提供します。ML シリーズ カードは RMON を特徴としており、NMS（Network Management System; ネットワーク管理システム）と連動するよう設計されています。

（注） この章で使用されるコマンドの構文および使用方法の詳細については、『*Cisco IOS Configuration Fundamentals Command Reference*』Release 12.2 の「System Management Commands」を参照してください。

（注） Cisco IOS を使用して RMON を管理する場合の詳細については、『*Cisco IOS Configuration Fundamentals Configuration Guide*』の「Configuring RMON Support」の章を参照してください。

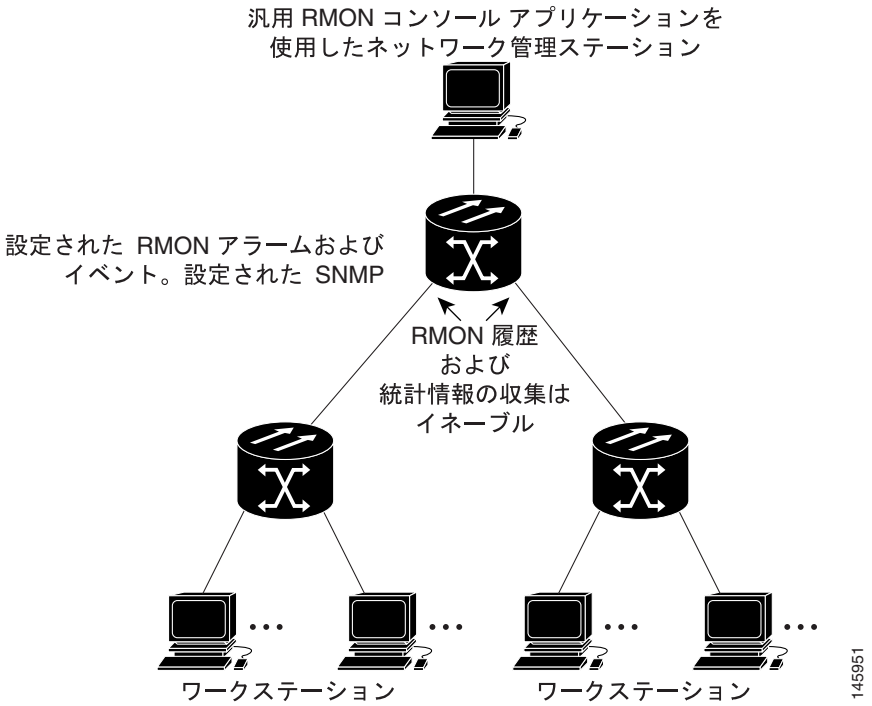
この章の内容は次のとおりです。

RMON の概要

RMON は、Internet Engineering Task Force（IETF; インターネット技術特別調査委員会）標準モニタ仕様で、各種ネットワーク エージェントおよびコンソール システムがネットワーク モニタリング データを交換できるようにします。RMON 機能を SNMP（簡易ネットワーク管理プロトコル）エージェントとともに使用すると、接続されたすべての LAN セグメント上の ML シリーズ カードと他のスイッチ間を流れるトラフィックをモニタリングできます。

ML シリーズ カードによってサポートされる MIB（管理情報ベース）の詳細については、「サポート対象の MIB」（p.22-5）を参照してください。

図 21-1　RMON の例



RMON の設定

ここでは、ML シリーズ カードで RMON を設定する方法について説明します。

RMON のデフォルト設定

RMON はデフォルトでディセーブルに設定されています。アラームやイベントは設定されていません。

RMON アラームおよびイベントの設定

CLI（コマンドライン インターフェイス）または SNMP 互換 Network Management Station（NMS; ネットワーク管理ステーション）を使用することにより、ML シリーズ カードを RMON 用に設定できます。NMS 上で汎用 RMON コンソール アプリケーションを使用し、RMON のネットワーク管理機能を利用することを推奨します。また、RMON MIB オブジェクトにアクセスするため、ML シリーズの SNMP を設定する必要があります。SNMP の設定方法の詳細については、第 22 章「SNMP の設定」を参照してください。

RMON アラームおよびイベントをイネーブルにするには、特権 EXEC モードを開始して、次の手順を実行します。この手順は必須です。

	コマンドの説明	目的
ステップ 1	`configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**rmon event** *number* [**description** *string*] [**log**] [**owner** *string*] [**trap** *community*]	RMON イベント テーブルに RMON イベント番号に対応付けられたイベントを追加します。 • *number* には、イベント番号を指定します。指定できる範囲は 1 ～ 65,535 です。 • （任意）**description** *string* には、イベントの説明を指定します。 • （任意）イベントがトリガーされたときに RMON ログ エントリを生成するには、**log** キーワードを使用します。 • （任意）**owner** *string* には、このイベントの所有者を指定します。 • （任意）**trap** *community* には、このトラップに使用される SNMP コミュニティ ストリングを入力します。

	コマンドの説明	目的
ステップ 3	**rmon alarm** *number variable interval* {**absolute** | **delta**} **rising-threshold** *value* [*event-number*] **falling-threshold** *value* [*event-number*] [**owner** *string*]	MIB オブジェクトに対してアラームを設定します。 • *number* には、アラーム番号を指定します。指定できる範囲は 1 ～ 65,535 です。 • *variable* には、モニタリングする MIB オブジェクトを指定します。 • *interval* には、アラームが MIB 変数をモニタする時間を秒単位で指定します。範囲は、1 ～ 2,147,483,647 秒です。 • 各 MIB 変数を直接テストするには、**absolute** キーワードを指定します。MIB 変数のサンプル間の変動をテストする場合は、**delta** キーワードを指定します。 • *value* には、アラームをトリガーする数値とアラームをリセットする数値を指定します。上昇しきい値および下限しきい値の範囲は、–2,147,483,648 ～ 2,147,483,647 です。 • （任意）*event-number* には、上昇または下限しきい値の限度を超過したときにトリガーされるイベント番号を指定します。 • （任意）**owner** *string* には、このアラームの所有者を指定します。
ステップ 4	**end**	特権 EXEC モードに戻ります。
ステップ 5	**show running-config**	エントリを確認します。
ステップ 6	**copy running-config startup-config**	（任意）コンフィギュレーション ファイルにエントリを保存します。

アラームをディセーブルにするには、設定したアラームごとに **no rmon alarm** *number* グローバル コンフィギュレーション コマンドを実行します。特定の番号を指定しないで、設定したアラームをすべてディセーブルにすることはできません。アラームは個別にディセーブルにする必要があります。イベントをディセーブルにするには、**no rmon event** *number* グローバル コンフィギュレーション コマンドを使用します。アラームおよびイベントとその相互作用の詳細については、Request For Comments（RFC; コメント要求）1757 を参照してください。

アラームはいずれの MIB オブジェクトに対しても設定できます。次の例では、**rmon alarm** コマンドを使用して RMON アラーム番号 10 を設定します。このアラームは、MIB 変数 *ifEntry.20.1* を 20 秒ごとに 1 回モニタし、アラームがディセーブルになるまで、変数の上昇または下降の変化をチェックします。*ifEntry.20.1* 値が 15 以上の MIB カウンタの増加を示した場合（たとえば 100000 から 100015）、アラームがトリガーされます。次にアラームは、イベント番号 1 をトリガーします。これは、**rmon event** コマンドで設定されます。設定可能なイベントにはログ エントリまたは SNMP トラップを含めることができます。*ifEntry.20.1* 値が 0 に変化した場合、アラームはリセットされ、再度トリガーできます。

```
ML_Series(config)# rmon alarm 10 ifInErrors.65539 20 delta rising 15 1 fall 0
```

（注） この例では、下限しきい値が 0 の場合、任意のイベントをトリガーしません。

65539 がインターフェイス POS 0 の SNMP ifIndex である場合、SNMP get の付いた特定のポート用に SNMP ifIndex を取得できます。この出力例では、POS0 の SNMP ifIndex は 65539 です。

```
tuvoks-view:128> getmany -v2c 10.92.56.97 tcc@1 ifDescr
ifDescr.65536 = GigabitEthernet0
ifDescr.65537 = GigabitEthernet1
ifDescr.65538 = Null0
ifDescr.65539 = POS0
ifDescr.65540 = POS1
ifDescr.65541 = SPR1
tuvoks-view:129>
```

次の例では、**rmon event** コマンドを使用して RMON イベント番号 1 を作成します。イベントは *High ifOutErrors* として定義され、アラームによってイベントがトリガーされるときにログ エントリが生成されます。ユーザ *jjones* は、このコマンドによってイベント テーブルに作成された行を所有します。この例では、イベントがトリガーされたときに SNMP トラップも生成されます。

```
ML_Series(config)# rmon event 1 log trap eventtrap description "High ifOutErrors"
owner jjones
```

インターフェイスでのグループ履歴統計情報の収集

収集情報を表示するには、最初に RMON アラームおよびイベントを設定する必要があります。

インターフェイス上でグループ履歴統計情報を収集するには、特権 EXEC モードで次の手順を実行します。この手順は任意です。

	コマンドの説明	目的
ステップ 1	**configure terminal**	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**interface** *interface-id*	履歴を収集するインターフェイスを指定し、インターフェイス コンフィギュレーション モードを開始します。 （注）　グループ履歴統計情報は、Packet-over-SONET/SDH（POS）（POS_ interface）では動作せず、イーサネット インターフェイスでのみ動作します。
ステップ 3	**rmon collection history** *index* [**buckets** *bucket-number*] [**interval** *seconds*] [**owner** *ownername*]	指定されたバケット数および時間に関する履歴収集をイネーブルにします。 • *index* には、統計情報の RMON グループを指定します。指定できる範囲は 1 ～ 65,535 です。 • （任意）**buckets** *bucket-number* には、統計情報の RMON 収集履歴グループに対する、最大バケット数を指定します。指定できる範囲は 1 ～ 65,535 です。デフォルトは 50 バケットです。 • （任意）**interval** *seconds* には、各ポーリング サイクルの秒数を指定します。指定できる範囲は 1 ～ 3600 です。デフォルトは 1800 秒です。 • （任意）**owner** *ownername* には、統計情報の RMON グループの所有者名を入力します。
ステップ 4	**end**	特権 EXEC モードに戻ります。
ステップ 5	**show running-config**	エントリを確認します。
ステップ 6	**show rmon history**	ML シリーズ カード履歴テーブルの内容を表示します。

	コマンドの説明	目的
ステップ 7	**copy running-config startup-config**	（任意）コンフィギュレーション ファイルにエントリを保存します。

履歴収集をディセーブルにするには、**no rmon collection history** *index* インターフェイス コンフィギュレーション コマンドを使用します。

次に、所有者 *root* の RMON 履歴を収集および表示する例を示します。

```
ML_Series(config)# interface gigabitethernet1
ML_Series(config-if)# rmon collection history 2 owner root
ML_Series(config-if)# end
ML_Series# show rmon history
Entry 2 is active, and owned by root
Monitors ifIndex.393217 every 1800 second(s)
Requested # of time intervals, ie buckets, is 50,
```

インターフェイスでのグループ イーサネット統計情報の収集

インターフェイス上でグループ イーサネット統計情報を収集するには、特権 EXEC モードで次の手順を実行します。この手順は任意です。

	コマンドの説明	目的
ステップ 1	**configure terminal**	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**interface** *interface-id*	統計情報を収集するインターフェイスを指定し、インターフェイス コンフィギュレーション モードを開始します。
ステップ 3	**rmon collection stats** *index* [**owner** *ownername*]	インターフェイスでの RMON 統計情報収集をイネーブルにします。 • *index* には、統計情報の RMON グループを指定します。指定できる範囲は 1 ～ 65,535 です。 • （任意）**owner** *ownername* には、統計情報の RMON グループの所有者名を入力します。
ステップ 4	**end**	特権 EXEC モードに戻ります。
ステップ 5	**show running-config**	エントリを確認します。
ステップ 6	**show rmon statistics**	ML シリーズ カード統計情報テーブルの内容を表示します。
ステップ 7	**copy running-config startup-config**	（任意）コンフィギュレーション ファイルにエントリを保存します。

グループ イーサネット統計情報の収集をディセーブルにするには、**no rmon collection stats** *index* インターフェイス コンフィギュレーション コマンドを使用します。

次に、所有者 *root* の RMON 統計情報を収集する例を示します。

```
ML_Series(config)# interface gigabitethernet1
ML_Series(config-if)# rmon collection stats 2 owner root
```

ML シリーズ カードの CRC エラーしきい値の概要

ML シリーズ カードの POS ポートは、SF（信号障害）アラームおよび SD（信号劣化）アラームを含めた、SONET/SDH 障害および GFP 障害のアラームを報告します。多くの場合、これらのアラームはユーザに対し、POS ポート上で超過 Cyclic Redundancy Check（CRC; 巡回冗長チェック）エラーを発生させる原因ともなる問題を警告します。ただし、超過 CRC エラーが POS ポートで発生しても、リンクには報告されるべき SONET 障害または GFP 障害が存在しないことがあります。このような状況には、リンクの他端の ML シリーズ カードから送信されるパケットの CRC エラーまたはビット エラー レートが SF または SD 障害をトリガーするには低すぎるが、重大な CRC パケット エラー レートを引き起こすには十分に高いといった例が含まれます。

デフォルトの ML シリーズ カードにシスコ固有の RPR が実装され、SONET/SDH 障害または GFP 障害が報告されていない状況では、POS インターフェイスは Shared Packet Ring（SPR; 共有パケット リング）インターフェイスのメンバーとしてアップ ステートのままになります。トラフィックはゆっくり失われ、アラームまたはアクションをトリガーしません。

FCS しきい値設定および検出機能はこの問題を解決します。CRC エラーによるパケット損失のパーセンテージが、設定可能なしきい値を超過する場合、アラームを発生するよう ML シリーズ カードを設定できます。発生したアラームは、CRC Threshold Crossing Alarm（CRC-ALARM）で、アラーム重大度が Major（MA）で、サービスに影響する（SA）SONET/SDH アラームです。報告された SONET/SDH アラームは、CTC の Alarms タブの下に表示されます。

また、ポートのリンク ダウン状態をトリガーし、シスコ固有の RPR をラップするよう CRC-ALARM を設定することもできます。デフォルトでは、CRC-ALARM はディセーブルです。アラームが設定されているとき、デフォルトではリンク ダウン アクションおよびラップ アクションはディセーブルのままです。この機能も ML シリーズ カードのイーサネット ポートでサポートされます。

しきい値およびトリガーされたアクション

設定可能なしきい値は、可変フレーム長およびさまざまな帯域幅のパーセンテージによって、その有用性が損なわれるので、BER は設定されていません。代わりに、CRC エラー レートをトラフィックのパーセンテージとして使用して、より適切な測度を設定します。トリガーするしきい値は次のとおりです。

- 10e-2 または 1% トラフィック（100 パケットで 1 個の CRC エラー）
- 10e-3 または 0.1% トラフィック（1000 パケットで 1 個の CRC エラー）（デフォルト）
- 10e-4 または 0.01% トラフィック（10000 パケットで 1 個の CRC エラー）

デフォルトのしきい値は、トラフィックの CRC エラー レート 0.1% です。音声およびビデオ トラフィックの場合、エラー レート 1% は通常、クリティカルな問題で、0.1% はメジャーな問題です。エラー レートが 0.1%（1000 パケットごとに 1 エラー）を超えると、音声およびビデオはラップをトリガーする必要があります。通常のデータ トラフィックの場合、エラー レート 10% のトラフィックはクリティカルな問題で、ただちに解決する必要があります。1% のトラフィックはマイナーな問題です。

超過 CRC エラーが検出されたあと、次のアクションが発生します。

1. このオプションが設定されている場合、シスコ固有の RPR がラップします。
2. このオプションが設定されている場合、リンクがシャットダウンします。
3. リンクがシャットダウンすると、Path Defect Indication（PDI; パス障害表示）が遠端の ML シリーズ カード ポートに送信されます。これにより、リモート エンドがラップします。
4. ローカル エンドの ML シリーズ カードに対して、CRC-ALARM が発生します（リモート エンドも超過 CRC エラーを受信している場合、CRC-ALARM が遠端の ML シリーズ カード ポートに対して発生します）。

CRC-ALARM の SONET/GFP 抑制

この超過 CRC エラー検出は、SONET/GFP 障害とは別のものです。1 つの問題により、SONET/GFP 障害と CRC-ALARM の両方が引き起こされる可能性があります。この事例では、CRC エラーしきい値検出は遅いプロセスなので、CRC-WRAP アラームの前に SONET/GFP 障害が発生します。SONET/GFP 障害によりリンクがダウンする場合、このリンクダウンは CRC-ALARM が検出される前に発生して CRC-ALARM を抑制します。CRC-ALARM を発生させる SONET/GFP 障害がリンクダウン トリガーではなく、CRC-ALARM によってリンクダウンするよう設定されている場合、CRC-ALARM はリンクダウンを報告してトリガーします。

CRC-ALARM のクリア

トリガー アクションがディセーブルの場合（デフォルト）、一定の時間、エラー レートがしきい値を下回ると CRC-ALARM は自動的にクリアされます。

トリガー アクションがイネーブルの場合、ユーザが手動で CRC-ALARM をクリアする必要があります。アラームによるラップまたはリンクダウンは、トラフィックとポートからのトラフィック内の CRC エラー両方をブロックするので、これが必要になります。CRC エラーがない場合、ファイバの汚れや障害の発生した ML シリーズ カードなどの根本的な問題が存続しても、自動的にクリアされます。この状況では、インターフェイスのフラッピングが発生します。

手動でクリアする前に、CRC-ALARM の根本原因を判別し、解決する必要があります。解決したら、アラームを手動でクリアする方法がいくつかあります。

- Cisco IOS CLI を通じて、EXEC レベルで **clear crc alarm interface** *interface-type interface-number* コマンドを入力します。
- Cisco IOS CLI を通じて、リンクされたポートで管理上の **shutdown** を行ってから、**no shutdown** を行ってポートをイネーブルにします。
- CTC または TL-1 を通じて、回線をディセーブルにしてから再度イネーブルにします。
- CTC または TL-1 を通じて、SONET/SDH 回線を削除し、同じ送信元と宛先を持った回線を作成します。

同期化のラップ解除

ML シリーズ カードのソフトウェアは、エラー フレームを監視する POS インターフェイスで CRC-ALARM アラームを発生させます。単一方向の FCS エラーの場合、ユーザは、CRC-ALARM アラームが発生したスパンの一方の端の POS ポートで unwrap コマンドを実行するだけです。双方向の障害の場合、スパンの両端で CRC-ALARM アラームが発生するので、スパンの各端で一度にコマンドを実行する必要があります。

リンクの各端の POS ポートはラップされるので、CRC-ALARM がクリアされたときにラップを削除する（ラップ解除）には、調整が必要です。また、ソフトウェアはラッピングを発生させる他のエラーがないことを確認する必要があります。次の例では、単一方向と双方向両方の障害のこの処理の方法を示します。簡単にするために、この例では、超過 CRC エラーはラッピングを引き起こす唯一の条件であることを前提とします。

単一方向エラー

図 21-2 に、ノード E の POS ポート 0 での単一方向の超過 CRC エラーによってラップされたシスコ固有の RPR を示します。これは、CRC-ALARM も報告します。これにより、ノード E の POS ポート 1 とノード D の POS ポート 0 がラップします。図のキャプションでは、プロセスをさらに説明します。

図 21-2　単一方向の超過 CRC エラーでラップされたシスコ固有の RPR

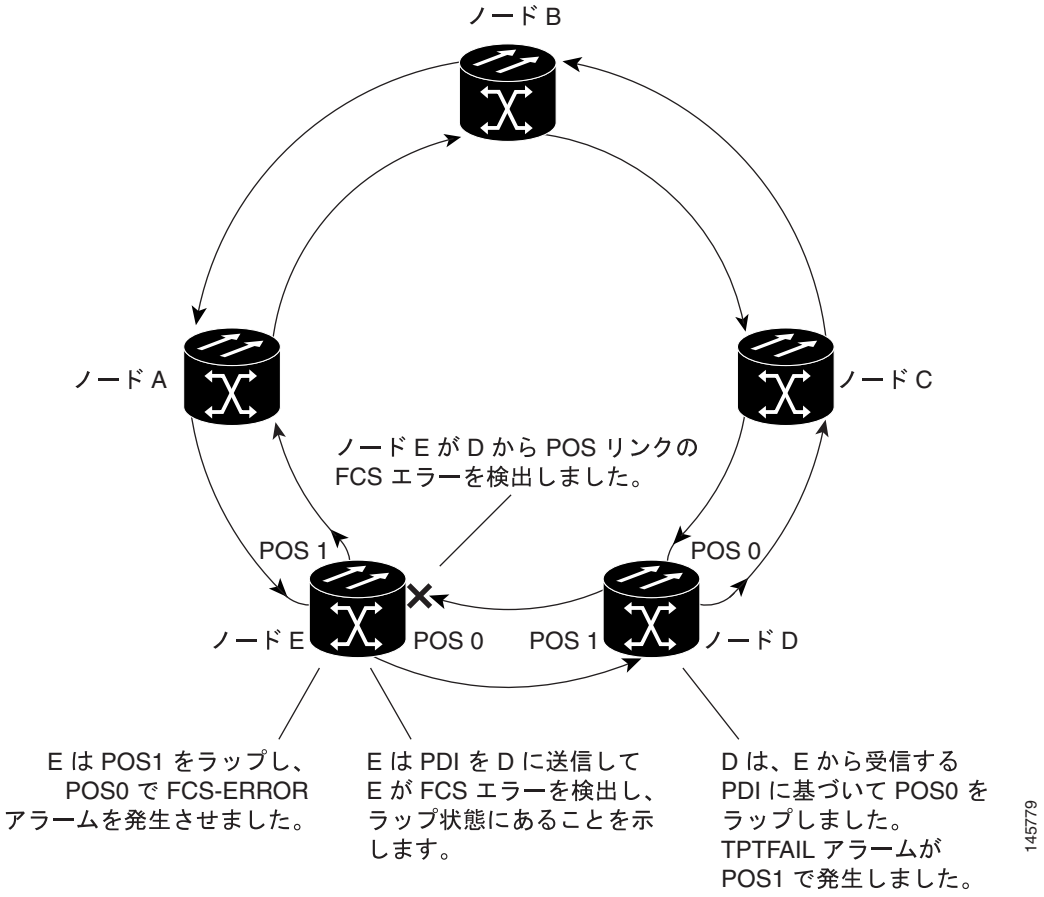
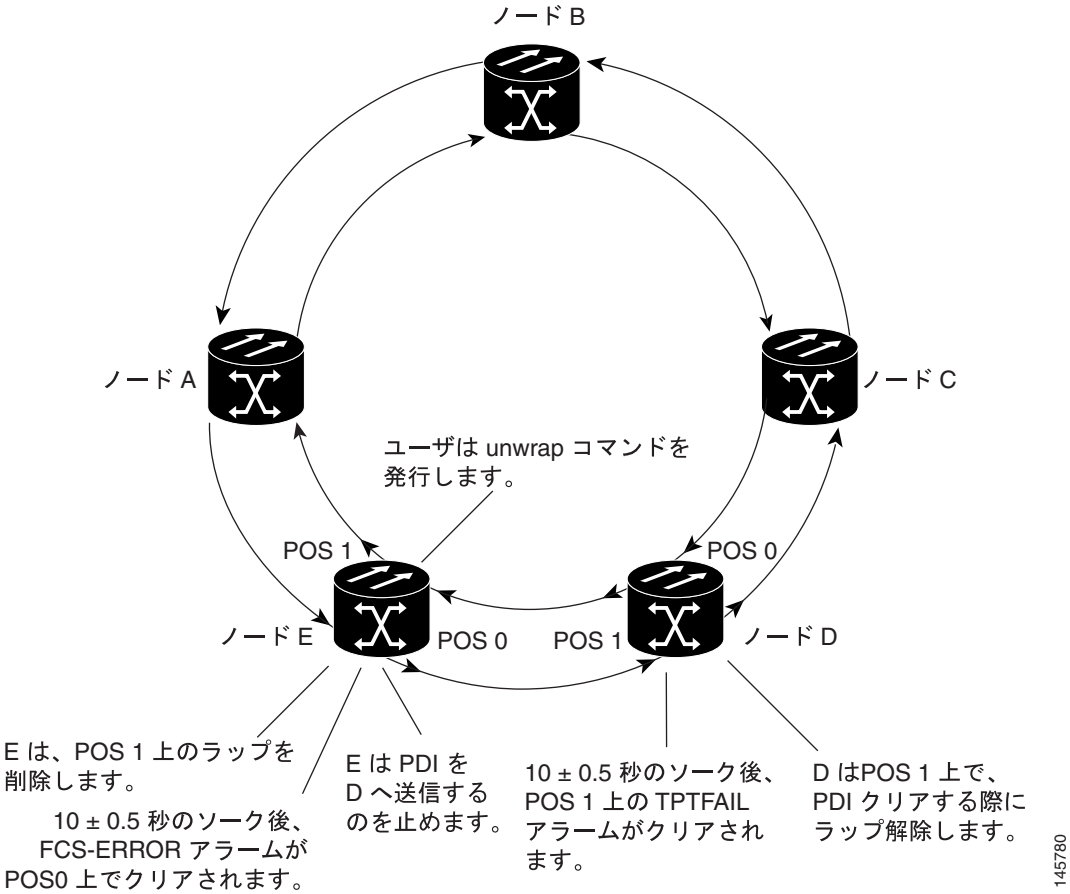


図 21-3 に、図 21-2 のラップ解除シーケンスを示します。ラップ解除のためのトラフィック ヒットは、ノード D でクリアされた PDI を宣言するのに必要なソーク時間に依存します。

図 21-3　単一方向の超過 CRC エラーでラップ解除されたシスコ固有の RPR



双方向エラー

図 21-4 に、双方向の超過 CRC エラーでラップされたシスコ固有の RPR を示します。両方のポートが CRC-ALARM を報告します。図のキャプションでは、プロセスをさらに説明します。

図 21-4　双方向の超過 CRC エラーでラップされたシスコ固有の RPR

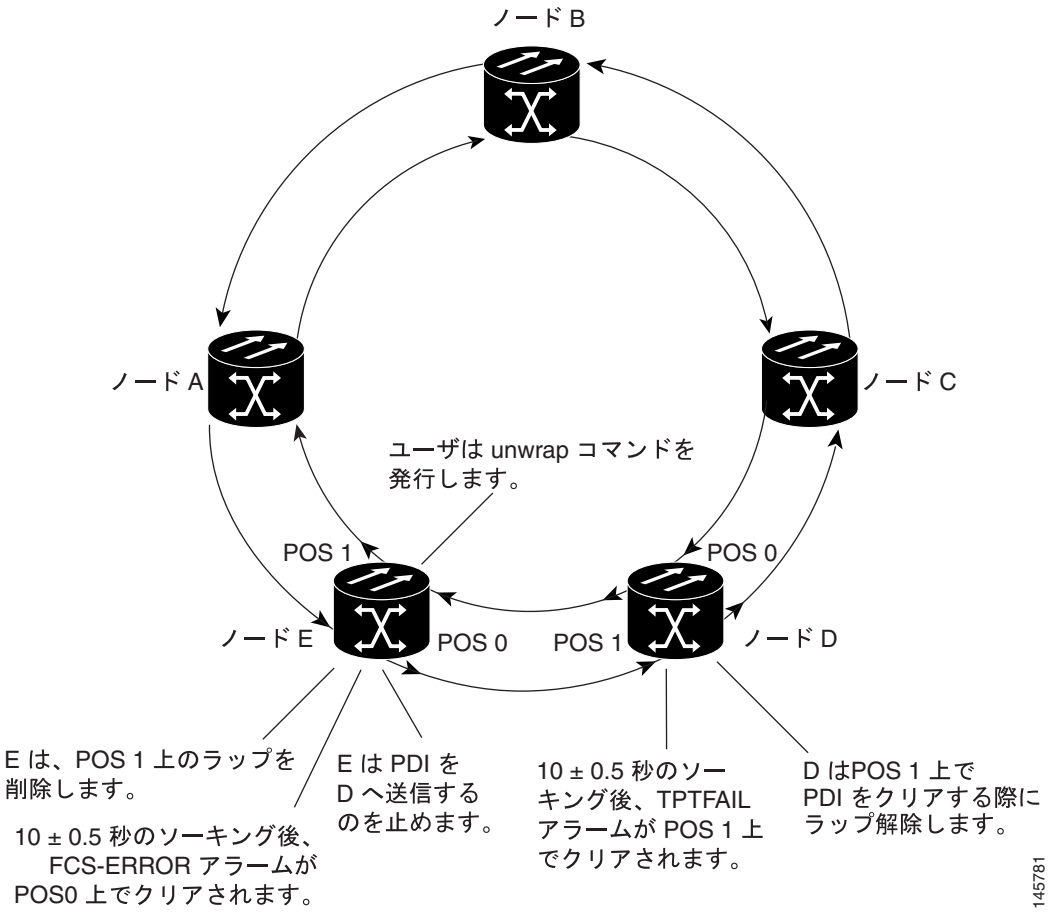
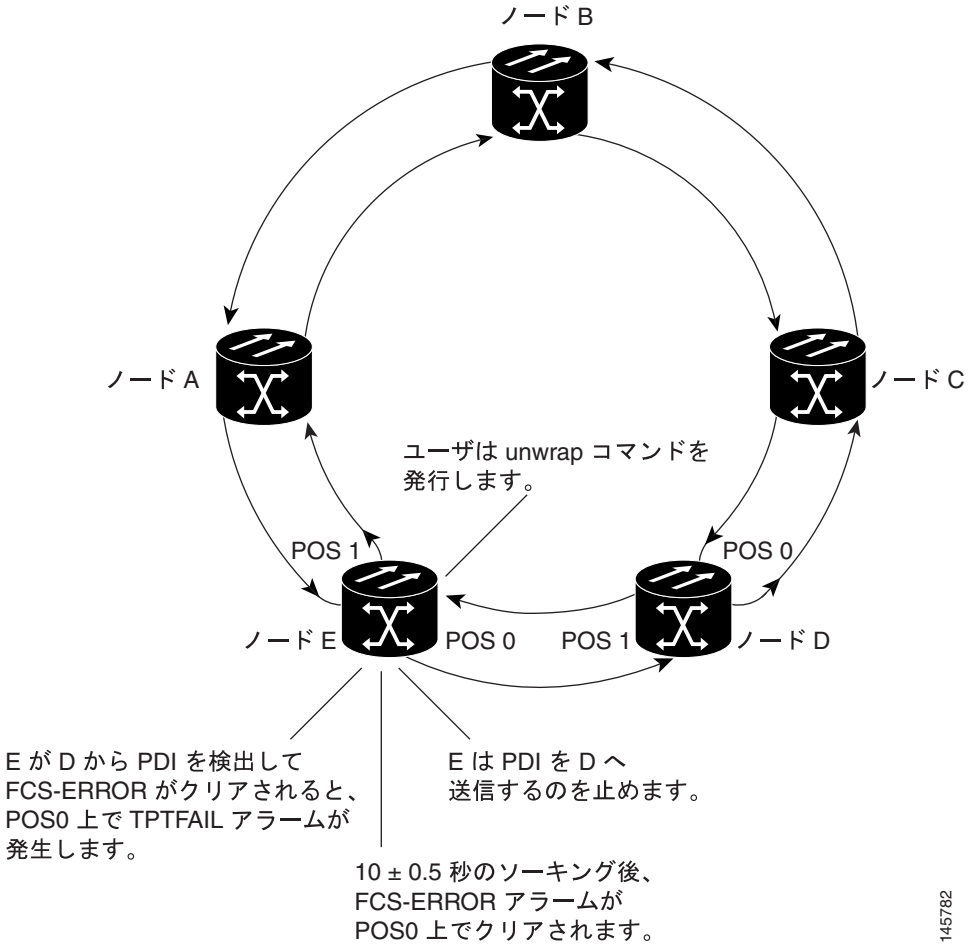


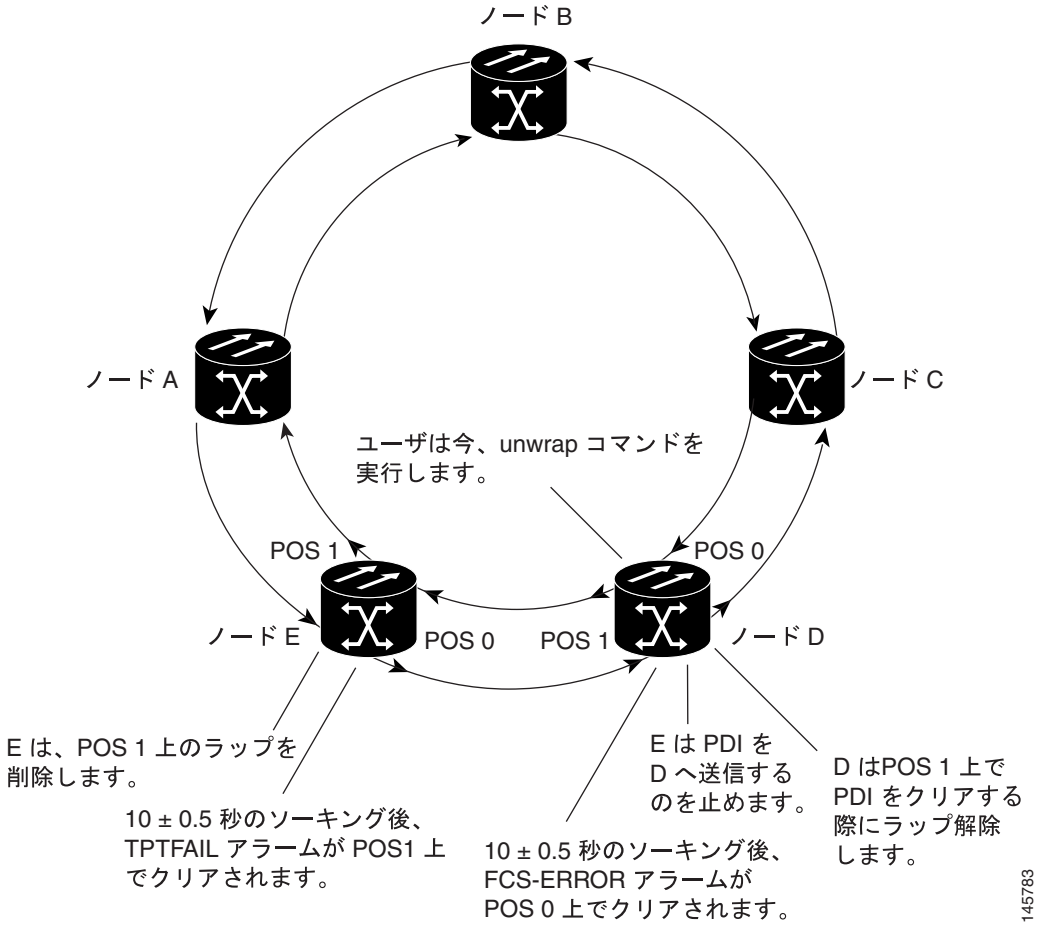
図 21-5 に、図 21-4 のラップ解除シーケンスの最初の部分を示します。ノード E で unwrap コマンドが設定されたあと、ラップ解除が行われます。この双方向事例でのラップ解除の場合、リンクの両端の POS ポートにコマンドを設定する必要があります。

図 21-5　双方向の超過 CRC エラーでラップ解除されたシスコ固有の RPR の最初の段階



最初の CRC-ALARM クリア コマンドのあと、ノード E は POS ポート 1 をラップ解除しません。ノード D は PDI をノード E に送信し続けるので、CRC-ALARM がクリアされるとノード E は TPTFAIL アラームを発生します。この時点では、シスコ固有の RPR は単一方向障害と同様のステートにいます。図 21-5 に示すように、ユーザが 2 回めの unwrap コマンドを実行したらラップ解除は完了です。

図 21-6　双方向の超過 CRC エラーでラップ解除されたシスコ固有の RPR の第 2 段階



ML シリーズ カードの CRC エラーしきい値の設定

ML シリーズ カードの CRC エラーしきい値を設定するには、特権 EXEC モードで次の手順を実行します。

	コマンドの説明	目的
ステップ 1	`ML_Series# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	`ML_Series (config)# interface` *interface-type interface-number*	インターフェイス コンフィギュレーション モードを開始します。
ステップ 3	`ML_Series (config-if)# [no] trigger crc threshold` [*threshold-value*]	fcs エラー レベルを帯域幅のパーセンテージとして設定し、SONET/SDH CRC-ALARM をトリップします。有効な値は次のとおりです。 • **2 —** 10e-2 または 1% トラフィック（100 パケットで 1 CRC エラー） • **3 —** 10e-3 または 0.1% トラフィック（1000 パケットで 1 CRC エラー）（デフォルト） • **4 —** 10e-4 または 0.01% トラフィック（10000 パケットで 1 CRC エラー） このコマンドの **no** 形式では、レベルの設定をデフォルトのしきい値 **3** に戻します。
ステップ 4	`ML_Series (config-if)# [no] trigger crc action`	（任意）報告するポートに対して、リンクダウンを発生させるよう CRC-ALARM を設定します。シスコ専用 RPR POS ポートに対して設定します。これもシスコ固有の RPR をラップします。 このコマンドの **no** 形式では、トリガーの設定をデフォルトのオフに戻します。
ステップ 5	`ML_Series(config-if)# [no] trigger crc delay` soak-time	（任意）超過 CRC エラー検出のソーク時間（分）を設定します。有効な値は 3 ～ 10 分です。 このコマンドの **no** 形式では、遅延の設定をデフォルトの 1 分に戻します。
ステップ 6	`ML_Series# end`	特権 EXEC モードに戻ります。
ステップ 7	`ML_Series# show running-config`	エントリを確認します。
ステップ 8	`ML_Series# copy running-config startup-config`	（任意）コンフィギュレーション ファイルにエントリを保存します。

クリア CRC エラー コマンドを使用した CRC-ALARM ラップの解除

Cisco IOS CLI の **clear crc alarm interface** *interface-type interface-number* コマンドは、対応する SONET/SDH エラーに対応するエラーがなく、FCS エラーによってシスコ固有の RPR ラップが発生したときにこれを解除します。SONET/SDH 障害またはキープ アライブ（KA）障害などの他の原因によるラップの解除は行いません。FCS エラーがなくても SONET/SDH または KA 障害が存在する場合、ソフトウェアはエラー メッセージを出してコマンドを拒否します。FCS エラーが存在し、SONET/SDH または KA 障害が存在する場合、コマンドはソフトウェアによって受け入れられますが、ノードは障害が解決してからのみラップ解除します。この場合、SONET/SDH または KA 障害がクリアされたあと、コマンドを再度実行する必要はありません。

（注） ラップ解除はただちに行われませんが、条件が満たされれば行われます。

ML シリーズ カードの CRC-ALARM をクリアするには、特権 EXEC モードで次の手順を実行します。

	コマンドの説明	目的
ステップ 1	`ML_Series # clear crc alarm interface interface-type interface-number`	SONET/SDH CRC-ALARM をクリアし、条件が満たされればシスコ固有の RPR がラップ解除できるようにします。

CRC エラーの ML シリーズ カードの RMON の設定

ML シリーズ カードは、CRC エラーのモニタリングを含め、NMS を使用して SNMP Performance Monitoring（PM; パフォーマンス モニタリング）を行うことをサポートします。NMS が、ML シリーズ カード インターフェイスすべてのインターフェイス インデックス エラー（ifInErrors）をモニタするために定期的なポーリングとプログラムされたしきい値をサポートする場合、NMS を信頼して CRC エラーを管理およびモニタリングできます。

NMS がポーリングをサポートしない場合、または望ましいポーリング周波数によって使用される帯域幅が多すぎる場合、Cisco IOS CLI を通じて SNMP トラップを ML シリーズ カードに設定できます。この方法は、ONS 15454 SONET/SDH の ML シリーズ カード専用です。ONS 15310-CL および ONS 15310-MA の ML シリーズ カードの RMON 機能は、Cisco Transport Controller（CTC）、Transaction Language One（TL1）、または Cisco Transport Manager（CTM）を介した、ノードを管理する標準的な方法で管理するのが一番です。

ML シリーズ カードの CRC しきい値の設定の注意事項

NMS PM アラートを生成するインターフェイス CRC エラー（ifInErrors）のしきい値を決定するための注意事項です。

- SONET/SDH ビット エラーは POS CRC エラーも作成します。SONET/SDH エラーと POS エラー間にはアラーム抑制階層がないので、各エラー セットは個別のアラートを作成します。
- インターフェイスの実際のパケット レートは、予測不可能です。高帯域幅のインターフェイスでは、低いデータ トラフィックが続く特定の時間に、分単位でわずかなパケットのみを転送する可能性があります。これは、比較的少ない CRC エラー数でも 100% の損失であることを意味します。低帯域幅のインターフェイスは、特定の時間に、分単位で高パケット カウント（100 万単位）を転送します。したがって、比較的少ない CRC エラー数の場合には、エラー レート 10^{-9} を意味します。この状況により、非パケットベースの PM にしばしば使用される最大 BER を単純に判別できなくなります。
- モニタリング問題または主要な問題の兆候を示す ML シリーズ カードの CRC エラーのモニタリングも設定できます。マイナーな問題のモニタリングの場合、60 秒間に 10 個のエラーなど、比較的速くて影響されやすいエラー レートのトリガーを設定します。この方法は、インターフェイスがアップ状態またはダウン状態になるか、ファイバ エラーが発生するか、あるいは SONET/SDH 保護イベントが発生する（保護が 50 ms 以内に発生しても）たびに NMS アラートを発生します。主要な問題のみをモニタリングし、アラート数を減らすには、300 秒間に 1000 個のエラーなど比較的高いしきい値を設定します。

SNMP を通じた CRC エラーへのアクセス

各インターフェイスの CRC エラーは、IF-MIB オブジェクトの ifInErrors（OID 1.3.6.1.2.1.2.2.1.14）で報告されます。SNMP get 要求により、ifInErrors の現在値を確認できます。各 ML シリーズ カードは、SNMP のインスタンスを個別に実行します。SNMP 要求は、コミュニティ ストリングに基づいて各 ML シリーズ カードにリレーされます。コミュニティ ストリングは次の形式を使用します。

com_str_configured_from_CTC@ml_slot_number

Cisco IOS を使用した CRC エラーしきい値の SNMP トラップの設定

ML シリーズ カードは、Cisco IOS の RMON トラップ機能をサポートします。Cisco IOS CLI を使用して、ifInErrors をモニタリングし、しきい値を超過した場合に NMS に対しトラップを生成するよう RMON を設定する必要があります。ONS 15454 SONET/SDH の ML シリーズ カードは、SNMP set 要求による RMON トラップの設定をサポートしません。この要求は、一般にネットワーク装置上のアクションを開始します。

ifInErrors をモニタリングし、しきい値を超過した場合に NMS に対しトラップを生成するよう RMON を設定するには、特権 EXEC モードで次の手順を実行します。

	コマンドの説明	目的
ステップ 1	`configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**rmon event** *number* [**log**] [**trap** *community*] [**description** *string*] [**owner** *string*]	RMON イベント テーブルに RMON イベント番号に対応付けられたイベントを追加します。 • *number* には、イベント番号を指定します。指定できる範囲は 1 ～ 65,535 です。 • （任意）イベントがトリガーされたときに RMON ログ エントリを生成するには、**log** キーワードを使用します。 • （任意）**trap** *community* には、このトラップに使用される SNMP コミュニティ ストリングを入力します。 • （任意）**description** *string* には、イベントの説明を指定します。 • （任意）**owner** *string* には、このイベントの所有者を指定します。
ステップ 3	**rmon alarm** *number* **ifInErrors**.*ifIndex-number interval* {**absolute** | **delta**} **rising-threshold** *value* [*event-number*] **falling-threshold** *value* [*event-number*] [**owner** *string*]	MIB オブジェクトに対してアラームを設定します。 • *number* には、イベント番号を指定します。指定できる範囲は 1 ～ 65,535 です。 • *ifIndex-number* 変数は、10 進表記の ML シリーズ カード インターフェイスの ifIndex 番号です（この番号の決定に関する詳細は、「ML シリーズ カードの ifIndex 番号の判別」[p.21-18] を参照）。 • *interval* には、MIB 変数をアラームがモニタする時間を秒単位で指定します。範囲は、1 ～ 4,294,967,295 秒です。 • 各 MIB 変数を直接テストするには、**absolute** キーワードを指定します。MIB 変数のサンプル間の変動をテストする場合は、**delta** キーワードを指定します。 • *value* には、アラームをトリガーする数値とアラームをリセットする数値を指定します。上昇しきい値および下限しきい値の範囲は、–2,147,483,648 ～ 2,147,483,647 です。 • （任意）*event-number* には、上昇または下限しきい値の限度を超過したときにトリガーされるイベント番号を指定します。 • （任意）**owner** *string* には、このアラームの所有者を指定します。
ステップ 4	`end`	特権 EXEC モードに戻ります。
ステップ 5	`show running-config`	エントリを確認します。
ステップ 6	`copy running-config startup-config`	（任意）コンフィギュレーション ファイルにエントリを保存します。

次に、CRC エラーしきい値の SNMP トラップを設定する例を示します。

```
ML_Series # configure terminal
ML_Series(config)# rmon event 10 log trap slot15 owner config
ML_Series(config)# rmon alarm 9 ifInErrors.983043 300 delta rising-threshold 1000 10
 falling-threshold 1000 10 owner config
ML_Series(config)# end
ML_Series # show running-config
ML_Series # copy running-config startup-config
```

この例で、**rmon alarm** コマンドに使用される 10 進表記の ML シリーズ カード インターフェイスの ifIndex 番号は、**ifInErrors.983043** です。この変数は、ML シリーズ カード インターフェイスの ifIndex 番号と組み合わせてモニタリングする MIB オブジェクトです。ML シリーズ カード インターフェイスの ifIndex 番号の決定に関する詳細は、「ML シリーズ カードの ifIndex 番号の判別」（p.21-18）を参照してください。

次に、5 分間にしきい値 1000 を超過する 1002 ifInErrors によって生成された上昇しきい値トラップの例を示します。

```
2005-03-22 16:25:38 ptlm9-454e56-97.cisco.com [10.92.56.97]:
SNMPv2-MIB:sysUpTime.0 = Wrong Type (should be Timeticks): 43026500
SNMPv2-MIB:snmpTrapOID.0 = OID: RMON-MIB:risingAlarm
RFC1271-MIB:alarmIndex.9 = 9
RFC1271-MIB:alarmVariable.9 = OID: IF-MIB:ifInErrors.983043
RFC1271-MIB:alarmSampleType.9 = deltaValue(2)
RFC1271-MIB:alarmValue.9 = 1002
RFC1271-MIB:alarmRisingThreshold.9 = 1000
SNMPv2-SMI:snmpModules.18.1.3.0 = IpAddress: 10.92.56.97
```

ML シリーズ カードの ifIndex 番号の判別

NMS がパフォーマンス データについて ML シリーズ カードをポーリングする場合、NMS は内部で ifIndex 番号を使用して、複数の MIB からのインターフェイス データを統合し、このデータとインターフェイス名を関連付けます。インターフェイス名は信頼できるので、実際の ifIndex 番号を知る必要はありません。

Cisco IOS CLI を使用してトラップを直接、生成するよう ML シリーズ カードを設定する場合、この関連付けた名前は使用しません。トラップを設定している各インターフェイスの実際の ifIndex 番号を使用する必要があります。実際の ifIndex 番号を決定するには、NMS を使用して各 ML シリーズ カード インターフェイスと VLAN（仮想 LAN）サブインターフェイスの ifIndex 番号を取得するか、またはインターフェイスの ifIndex 番号を計算します。

また、MIB ブラウザ（SNMP MIB 定義の検索サービス）を使用して、適切な ifIndex 番号の ifDescr を検証できます。ifDescr からの ifIndex 番号は、希望の ifIndex 番号でなければなりません。

ML シリーズ カードでは、イーサネットおよび POS インターフェイスの ifIndex 番号は、次の 2 種類のカードの情報からコンパイルされています。

- カードのシャーシ スロット番号 — スロット番号は、ML シリーズ カードが常駐するシェルフの物理的なスペースの番号です。ONS 15454 SONET/SDH シェルフの場合、有効な範囲はスロット 1 ～ 6、またはスロット 12 ～ 17 です。この情報は、CTC のシェルフ スロットのグラフ表示、あるいは物理シェルフの前面で見つけることができます。
- カード内のローカル ポート番号 — ONS 15454 SONET/SDH の ML シリーズ カードのポート番号は、ファースト イーサネット インターフェイスおよびギガビット イーサネット インターフェイスのインターフェイス番号と一致します。POS ポート番号はインターフェイス番号と一致せず、イーサネット ポートを連続してナンバリングしません。連続値は、最後のイーサネット ポート番号と最初の POS 番号（POS ポート 0）の間ではスキップされます。インターフェイスのポート番号を 表 21-1 に示します。

表 21-1　ML シリーズ カードのインターフェイスのポート番号

ML100T-12 ファースト イーサネット インターフェイス	ML100T-12 POS インターフェイス	ML100X-8 ファースト イーサネット インターフェイス	ML100X-8 POS インターフェイス	ML1000-2 ギガビット イーサネット インターフェイス	ML1000-2 POS インターフェイス
FE 0 = ポート 0	POS 0 = ポート 13	FE 0 = ポート 0	POS 0 = ポート 9	GE 0 = ポート 0	POS 0 = ポート 3
FE 1 = ポート 1	POS 1 = ポート 14	FE 1 = ポート 1	POS 1 = ポート 10	GE 1 = ポート 1	POS 1 = ポート 4
FE 2 = ポート 2		FE 2 = ポート 2			
FE 3 = ポート 3		FE 3 = ポート 3			
FE 4 = ポート 4		FE 4 = ポート 4			
FE 5 = ポート 5		FE 5 = ポート 5			
FE 6 = ポート 6		FE 6 = ポート 6			
FE 7 = ポート 7		FE 7 = ポート 7			
FE 8 = ポート 8					
FE 9 = ポート 9					
FE 10 = ポート 10					
FE 11 = ポート 11					

次の公式を使用して、スロットおよびポートを組み合わせて ifIndex を出します。

ifIndex = (slot * 10000h) + (port)

10000h は、16 進数を示し 65536 に相当します。その結果の ifIndex は 16 進数で有意味な 2 値数字ですが、10 進数では紛らわしく、明確でない数字です。たとえば、ifIndex E0002h はポート 2 のスロット 14 です。10 進表記での同じ番号は 917506 になります。**rmon alarm** コマンドは、10 進数表記の ifindex 番号を必要とします。

rmon alarm コマンドを使用して正確な ifindex 値を算出するための参照として、表 21-1 にスロット 1 ～ 17 のベース ifindex 番号を示します。希望のポート番号をスロット ベース番号に足すと正しい ifIndex 番号をすばやく決定できます。

表 21-2　ML シリーズ カードのインターフェイスのポート番号

ML シリーズ カードのスロット番号	16 進表記のベース ifIndex 番号	10 進表記のベース ifIndex 番号
1	10000h	65536
2	20000h	131072
3	30000h	196608
4	40000h	262144
5	50000h	327680
6	60000h	393216
12	C0000h	786432
13	D0000h	851968
14	E0000h	917504
15	F0000h	983040
16	100000h	1048576
17	110000h	1114112

ML シリーズ カードでの手動による CRC エラー検証

show interface コマンドを使用して、インターフェイス上の ML シリーズ カードの現在の CRC エラー カウントも検証できます。次に、6 つの総入力エラーの例を示します。これはすべての CRC エラーで、出力の最後の行にあります。

```
ML_Series(config)# show interface pos 0

POS0 is up, line protocol is up
Hardware is Packet/Ethernet over Sonet, address is 0005.9a39.713e (bia 0005.9a39.713e)
MTU 1500 bytes, BW 48384 Kbit, DLY 100 usec,
reliability 255/255, txload 1/255, rxload 182/255
Encapsulation: Cisco-EoS-LEX, crc 32, loopback not set
Keepalive set (10 sec)
Scramble enabled
ARP type: ARPA, ARP Timeout 04:00:00
Last input never, output never, output hang never
Last clearing of "show interface" counters never
Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
Queueing strategy: fifo
Output queue: 0/40 (size/max)
5 minute input rate 34621000 bits/sec, 60083 packets/sec
5 minute output rate 0 bits/sec, 0 packets/sec
311190527 packets input, 931220183 bytes
Received 0 broadcasts (0 IP multicast)
6 runts, 0 giants, 0 throttles
0 parity
6 input errors, 6 CRC, 0 frame, 0 overrun, 0 ignored
```

RMON ステータスの表示

（注）　RMON ステータス コマンドは、POS インターフェイス用には動作しません。

RMON のステータスを表示するには、表 21-3 に示す 1 つまたは複数のイネーブル EXEC コマンドを使用します。

表 21-3　RMON ステータスの表示用コマンド

コマンドの説明	目的
show rmon	一般的な RMON 統計情報を表示します。
show rmon alarms	RMON アラーム テーブルを表示します。
show rmon events	RMON イベント テーブルを表示します。
show rmon history	RMON 履歴テーブルを表示します。
show rmon statistics	RMON 統計情報テーブルを表示します。

例 21-1 に、表 21-3 のコマンドの例を示します。

例 21-1　show rmon コマンドで表示された CRC エラー

```
ML_Series# show rmon alarms

Alarm 9 is active, owned by config
Monitors ifInErrors.983043 every 300 second(s)
Taking delta samples, last value was 0
Rising threshold is 1000, assigned to event 10
Falling threshold is 1000, assigned to event 10
On startup enable rising or falling alarm

ML_Series# show rmon events
Event 10 is active, owned by config
Description is
Event firing causes log and trap to community slot15,
last event fired at  0y3w2d,00:32:39,
Current uptime       0y3w6d,03:03:12
Current log entries:
index  uptime              description
1      0y3w2d,00:32:39
```