Catalyst 9800メッシュWifiの問題のトラブルシューティング

内容

はじめに

このドキュメントでは、9800メッシュ環境をトラブルシューティングするさまざまな方法について説明します。

前提条件

要件

メッシュの導入に関する知識に加えて、ワイヤレスコントローラに関する知識があることが推奨されます。

1. 範囲と適用性

適用対象：当該問題は、海港および鉱業環境に関して発生したものとする。

* Catalyst 9800-L/9800-CL/9800-40ワイヤレスLANコントローラ

* 屋外メッシュ導入(RAP-MAP)

* デュアルバンド(2.4 GHz/5 GHz)WLAN

* 環境：

  * 長距離メッシュリンク

  * 高いRFノイズ/工業地帯（ポート、ターミナル、ヤード）

2. お客様から報告される一般的な症状

メッシュ/APの症状

1. メッシュAPにWLCでの加入が表示されるが、クライアントが接続しない

  * クライアントまたはアップストリームトラフィックなし

  * APがリブートされるまでpingは失敗します。

2. RAP-MAPリンク

* 間欠的にフラップする。

* MAPは予期せず別のRAP/MAPにローミングします。

* メッシュAPはWLCから切断され、手動でのリブートが必要です。

3. クライアント接続の症状

* クライアントが無期限にAuthenticating状態のままになる。

* クライアントはAPを経由してローミングするが、認証されない。

* クライアントは次の後にのみ接続します。

* WLCまたはAPのリブートからの強制的な削除

* 2.4 GHzでのクライアントの頻繁なドロップ

3. 高確率の根本原因バケット

[Category]	よくある問題
RF/設計	チャネルのオーバーラップ、チャネル幅が広い、アンテナの位置合わせ不良
メッシュ制御	親の選択が不安定、バックホールSNRが弱い
コンフィギュレーション	データレートの混在、複数のBGN、静的電力
[ソフトウェア（Software）]	wncdプロセスが停止し、クライアントの状態が古くなる
スケール/ロード	超過認証コール、EAPOLタイマーの不一致

4. 設計と設定の検証が必須

4.1メッシュバックホール（重要）

ルートAP(RAP)

- チャネル幅：20 MHzのみ
- RAP全体でチャネルがオーバーラップしない
- 同じブリッジグループ名(BGN)
- 静的チャネル割り当て
- MAPへのラインオブサイト

回避

- RAPでの20/40 MHzの混合
- すべてのRAPで同じチャネル
- 同じエリアに複数のBGN

4.2アンテナと取り付け

- 5 GHz全方向性アンテナ：
- 地面に垂直に取り付け
- メッシュバックホール専用5 GHz無線
- 長距離MAPに適した指向性アンテナ
- 障害物（金属、クレーン、コンテナ）の排除

5. RFおよびWLANのベストプラクティス

5.1データレート（強く推奨）

2.4 GHz

必須：12 Mbps

ディセーブル：6、9 Mbps

その他：サポート

5 GHz

必須：12 Mbps

ディセーブル：6、9 Mbps

その他：サポート

影響：

- スティッキクライアントの削減
- ローミングと認証の安定性が向上

5.2電源およびRRM

- APレベルのスタティックTX電力の回避
- グローバルRRMの使用
- 最小TX電力：
- 2.4 GHz:≥ 12 dBm

実稼働時間でのDCAの大幅な変更を回避

クライアント接続の問題のトラブルシューティング

問題の説明

メッシュ接続された領域：

- クライアントはMAPに正常に関連付けられます。
- 認証は開始されるが、完了しない。
- クライアントはWLC上でAuthenticating状態のままになります。
- クライアントは、認証を維持しながらAP間をローミングできます。
- 認証が成功するのは、クライアントがWLCから手動で削除されるか、MAPがリブートされた後だけです。

この動作は断続的で、オンデマンドでの再現は困難で、通常の認証フローの一部ではありません。

確認された症状

- Show wireless client summary:Authenticatingでスタックしているクライアントを表示します。
- クライアントが繰り返し認証を試みます。
- 明示的な認証の失敗や拒否は見られません。
- クライアントは、複数のローミングイベントの後もスタックしたままになります。
- この問題は主に、クライアントがMAP経由で接続されている場合に発生します。
- 運用負荷時の発行頻度の増加

クライアント接続の問題に関するメッシュ導入の主な寄与要因

1. メッシュバックホールの不安定性

- RAPとMAP間のRSSI/SNRの変動
- 認証中のMAP再選択の親。
- EAPタイムアウトまたは再送信を引き起こすメッシュ遅延。
- MAPは一時的にトラフィックを転送するが、一貫していない

影響：

- 認証状態マシンが完了しません。
- クライアントがAuthenticatingでスタックしている。

2. 認証中のローミング

- クライアントはMAP間、またはMAPとRAP間をローミングします。
- 認証コンテキストは完全には転送されません。
- クライアントがAuthenticating状態のままローミングを続行する

影響：

- 認証が繰り返し再起動する。
- クライアントがRUN状態になることはありません。

3. 無線を提供するクライアントの低データレート(2.4 GHz)

- 6 Mbpsまたは9 Mbpsの必須が有効。
- 過剰な再試行と通信時間の消費。
- 認証フレームの遅延またはドロップ。

影響：

- EAP交換はメッシュ上で信頼性が低くなります。
- 認証が明示的な障害なしでハングしているように見えます。

4. 同じRF制約を共有するメッシュバックホールとクライアントトラフィック

- メッシュリンクの使用率が高い
- クライアント認証トラフィックは次と競合：
- データ トラフィック
- 制御トラフィック
- 認証パケットは小さいが、時間の影響を受けやすい。

影響：

- 認証は再試行またはリセット後にのみ完了します

ヒットした問題を特定する方法（メッシュ認証スタック）

この問題は、メッシュ展開で上記のすべての条件が同時に見られた場合に発生したと見なされます。

クライアント動作インジケータ

- クライアントが60 ~ 120秒を超えてAuthenticating状態のままになっている。
- クライアントが自動的にRUN状態に移行しない。
- クライアントは次の後にのみ正常に接続します：
- WLCからのクライアントの強制削除
- メッシュAPのリブート
- クライアントは、Authenticating状態を維持したまま、MAPまたはRAP間でローミングできます。

WLCインジケータ

コマンド：

show wireless client summary（ワイヤレスクライアントの概要を表示）

インジケータ:

- 同じクライアントMACがAuthenticatingの下に永続的にリストされます。
- クライアントエントリは自然にエージングアウトしません。

クライアントが接続されてから10分を超えた場合は、次のコマンドを確認します。

show wireless client mac <クライアントMAC>

メッシュ固有のインジケータ

コマンド：

APメッシュペアレントの表示

APメッシュリンクの表示

インジケータ:

- クライアント認証中の親の変更または不安定さ
- RSSI/SNR値の変動
- メッシュバックホールでの再試行またはパケット損失の増加

必須のログ収集（障害が発生した時間帯）

クライアントがAuthenticating状態に留まっている間にログを収集する必要があります。
リブートまたはクライアントの削除後に収集されたログは、根本原因には有用ではありません。

1. コントローラのベースラインログ

show tech wirelessコマンド

show clock

目的:

- WLC全体の状態のキャプチャ
- ログ間のタイムスタンプの関連付け

2. クライアント状態検証ログ

show wireless client summary（ワイヤレスクライアントの概要を表示）

show wireless client summary | include Authenticating（ワイヤレスクライアントの概要の表示）

show wireless client mac <クライアントMAC>

3. WNCD内部ログ（重要）

詳細トレースを有効にする：

set platform software trace wncd chassis active r0 all verbose（プラットフォームのソフトウェアトレースをwncdシャーシでアクティブにするr0のすべての詳細を設定する）

ログの収集（過去30分間）:

show logging process wncd internal last 30分

クライアント固有のフィルタログ：

show logging process wncd start last 30 minutes filter mac <client-mac> to-file bootflash:wncd_client.log

4. 無線アクティブ(RA)トレース – クライアントごと

GUI から：

- Monitor > Wireless > Client > Troubleshooting
- 影響を受けるクライアントMACを追加します。
- RAトレースを開始します。
- 問題を再現します。

5. メッシュバックホール検証ログ

APメッシュリンクの表示

APメッシュペアレントの表示

APメッシュ統計情報の表示

6. オプション（使用可能な場合）– 認証サーバー・ログ

- 該当するクライアントのRADIUS認証ログ
- 認証の遅延と再送信

MAP-RAP接続解除問題のトラブルシューティング

問題の説明

複数のIW9167 MAP間でメッシュバックホール接続が断続的かつ予測不能に失われ、APの分離、メッシュ認証障害、到達不能なAP、およびクライアントトラフィックのブラックホールが発生する。回復にはAPのリブートまたはWLCの介入が必要になることが多かった。

症状

- MAPは親RAPとの関連付けを解除します。
- MAPは関連付けられているが、トラフィックを渡せない
- WLC、RAP、およびゲートウェイからMAPに到達できない
- 関連付けられているが、アップストリーム到達可能性がないクライアント
- 親MAPまたはRAPのローミング時のカスケード停止

エラーメッセージ/インジケータ

ERROR-MeshSecurity：タイマーが切れました

CRIT-MeshSecurity：メッシュセキュリティが親との認証に失敗しました

CRIT-MeshAwppAdj：親として削除

mlme_ext_vap_down:VAP(mon1)がダウンしている

ieee80211_ucfg_mesh_add_client()：ノードが見つかりません

DTLS終了アラート

CAPWAPハートビートタイムアウト

ヒットした問題を特定する方法（RAP-MAP接続の問題）

1. メッシュコントロールプレーンが正常に見える

上記のコマンドは正常に表示される可能性があり、トラフィック転送の検証に単独で使用することはできません。

show ap summary

show wireless mesh apツリー

show capwap client rcb

これらのコマンドは、コントロールプレーンの状態だけを確認します。

メッシュデータプレーンの障害の特定

マップ：メッシュステータスを表示

これは、メッシュ転送の状態を示す主要なインジケータです。

正常な出力

親AP MAC:24:D7:9C:04:79:B1

メッシュリンクの状態：UP

フォワーディングステート：ENABLED

トラフィックのブラックホール出力

親AP MAC:24:D7:9C:04:79:B1

メッシュリンクの状態：UP

フォワーディングステート：無効

解釈：
メッシュ隣接関係は存在するが、APがトラフィックを転送していない。

2. MAP：メッシュ履歴を表示

APのリロードなしで親の遷移が繰り返される場合は、フォワーディング状態が不安定であることを示しています。

CRIT-MeshAwppAdj：親として削除

CRIT-MeshAwppAdj：親として設定

CRIT-MeshAwppAdj：親として削除

このパターンでは、APが非転送状態になることがよくあります。

3. MAP syslogの症状

トラフィックのブラックホール化中に観察される一般的なsyslogメッセージ：

ieee80211_ucfg_mesh_add_client()：ノードが見つかりません

CLSM：ヌルキーのためキープログラミングをスキップします

これは、メッシュセキュリティコンテキストが不完全で、暗号化されたトラフィック転送を妨げていることを示します。

4. WLCのshow ap name <AP> mesh path

このコマンドは、データパスに対するコントローラのビューを確認します。

正常

パスステータス：アクティブ

データパス：完了

トラフィックブラックホール化

パスステータス：アクティブ

データパス：不完全

解釈：
メッシュパスは存在しますが、データ転送は確立されません。

5. ARP関連指標

VLAN SVIがWLC上に存在する環境：

- ARPエントリは、クライアントとAP用に存在します。
- クライアントトラフィックが失敗する
- ARPをクリアすると、接続がただちに復元されます。

この動作により、RFやCAPWAPの不安定性ではなく、データプレーンの転送障害が確認されます。

必須の口グ収集（障害が発生した時間帯）

フェーズ0：必須の準備（問題発生前）

重要：再起動後に収集された口グは、メッシュRCAには不十分です。

RAPおよびMAPでの永続的なデバッグの有効化

RAP上

端子の長さ0

メッシュイベントのデバッグ

debug mesh adjacency child（メッシュ隣接関係の子をデバッグ）

debug mesh adjacency packet

debug mesh adjacency channel

メッシュセキュリティのデバッグ

debug mesh forwardingパケット

debug capwap client events

debug capwap client error

端末モニタ

地図上

端子の長さ0

メッシュイベントのデバッグ

debug mesh adjacency parent

debug mesh adjacency packet

debug mesh adjacency channel

メッシュセキュリティのデバッグ

debug capwap client events

debug capwap client error

端末モニタ

問題が再発するまで、デバッグを有効のままにします。

フェーズ1 – 問題発生時の口グ収集（重要）

口グを収集する前にAPをリブートしないでください

影響を受けるMAPからの口グ（問題発生直後）

show mesh status

メッシュ履歴を最も古く表示

メッシュ履歴を表示

show flash syslog（隠しコマンド）

詳細なsyslog <日付>

RAP（前の親と新しい親）からの口グ

メッシュ履歴を最も古く表示

show mesh status

WLCからの口グ（障害発生時）

show wireless mesh apツリー

show wireless mesh neighbor（ワイヤレスメッシュネイバーを表示）

show ap name <AP-NAME> mesh path

show ap name <AP-NAME> config general

show tech-support wireless（登録ユーザ専用）

オプション（上限値）:

show logging process wncd start last 2 days level verbose（過去2日間のログプロセスの詳細を表示）

クライアントとトラフィックの関連付け（推奨）

障害が発生している時間帯に連続pingを実行します。

ping -t <ゲートウェイIP>

フェーズ2:RFと設定の検証（キャプチャ後）

RF検証(WLC)

show ap dot11 5ghzの概要

show ap dot11 24ghz summary

show ap name <AP> config dot11 5ghz

show ap name <AP> config dot11 24ghz

ARP/転送検証（トラフィックのブラックホールの場合）

SVIがWLCでホストされている場合：

clear arp-cache（ARPキャッシュのクリア）

トラフィックが回復→ると、ARP処理が原因になります。

フェーズ3：安定化アクション（検証済み）

メッシュトポロジ制御

- 必要に応じて、MAPで子をブロックを有効にします。
- MAPを最も近いRAPに強制的に接続します。
- メッシュホップカウントを削減する。

RF最適化

- RAPの送信電力を削減します。
- 5 GHzバックホールチャネルをロックします。
- 2.4 GHzチャネルの標準化(1/6/11)

前述の問題はすべて、メッシュ導入では非常に断続的で取得が困難なため、ログをキャプチャするクイックスクリプトを導入すると、より迅速に解決できます。

次に、クライアント認証の問題のためにWLC上で実行できるEEMスクリプトの例を示します。

完全なEEMスクリプト（WLC CLI経由で適用）

```
::cisco::eem::event_register_timerウォッチドッグタイム900 maxrun 240
名前空間のインポート：:cisco::eem::*
名前空間のインポート：:cisco::lib:*
# ----------------------------
#処理：WLCの時間文字列を秒に変換します。
#サポート： "X days Xh:Xm:Xs", "Xh:Xm:Xs", "Xm:Xs"
# ----------------------------
proc time_to_seconds {time_str} {
セット合計0
{[regexp {([0-9]+)\s+days?\s+([0-9]+)\s+h:([0-9]+)\s+m:([0-9]+)\s+s} $time_str -> d h m s]} {
セット合計[式{$d*86400 + $h*3600 + $m*60 + $s}]
} elseif {[regexp {([0-9]+)\s+h:([0-9]+)\s+m:([0-9]+)\s+s} $time_str -> h m s]} {
セット合計[式{$h*3600 + $m*60 + $s}]
} elseif {[regexp {([0-9]+)\s+m:([0-9]+)\s+s} $time_str -> m s]} {
セット合計[式{$m*60 + $s}]
} elseif {[regexp {([0-9]+)\s+s} $time_str -> s]} {
合計の設定($s)
}
合計を返す($t)
}
# ----------------------------
# Proc：ログ収集インスタンスの総数を追跡します（最大2）
# ----------------------------
proc get_log_count {} {
{[ファイルが/bootflash/auth_log_count.txtに存在する]} {
set fd [open /bootflash/auth_log_count.txt r]
set count [読み込み$fd]
閉じる($f)
return $count
}else {
リターン0
}
}
proc set_log_count {count} {
```

```
set fd [open /bootflash/auth_log_count.txt w]
$fd $countを配置します。
閉じる($f)
}
# ----------------------------
#主なEEM実行
# ----------------------------
if {[catch {cli_open} result]} {
出口1
}
配列セットcli $result
set fd $cli(fd)
cli_exec $fd "有効"
cli_exec $fd "ターミナル長0"
cli_exec $fd "ターミナル幅0"
#現在のログ収集数を取得する
set log_count [get_log_count]
max_log_instances 2を設定します
#すべてのクライアントをAuthenticating状態でプル
set summary [cli_exec $fd "show wireless client summary | include Authenticating"]
行を設定する[split $summary "\n"]
foreach line $lines {
# MACフォーマットxxxx.xxxx.xxxxに一致
if {[regexp {([0-9a-fA-F]{4}\.[0-9a-fA-F]{4}\.[0-9a-fA-F]{4})} $line -> mac]} {
set detail [cli_exec $fd "show wireless client mac-address $mac detail"]

# 「Connected For」時間文字列を抽出
if {[regexp {Connected For[[:space:]]*:[:space:]]*(.+)} $detail -> conn_time]} {
set seconds [time_to_seconds $conn_time]

# 15分（900秒）以上スタックしているかどうかを確認します。
{$seconds > 900} {
action_syslog msg "EEM：クライアント$macが$conn_time (>$seconds)の間、認証中にスタックしました"

#インスタンスの最大数より少ない場合にのみログを収集する
if {$log_count < $max_log_instances} {
action_syslog msg "EEM: Collecting WLC + client logs (Instance [expr {$log_count +
1}]/$max_log_instances)"
set log_file "/bootflash/auth_stuck_eem.log"


set fd_log [open $log_file a]

#クライアントごとのログ
puts $fd_log "\n=== [clock format [clock seconds]] | Client $mac | Stuck $conn_time ==="
```

```
$fd_log "\n – クライアントの詳細 – "を置く
$fd_log $detailを置きます
$fd_log "\n – クライアントの概要 – "を置く
$fd_logを追加[cli_exec $fd "show wireless client summary | include $mac"]

# WLC全体のログ
$fd_log "\n— WLC WNCDログ(30m) —"を配置
puts $fd_log [cli_exec $fd "show logging process wncd start last 30 minutes"]
$fd_log "\n— WLC Show Tech Wireless —"を出力します。
puts $fd_log [cli_exec $fd "show tech wireless"]

$fd_logを閉じる
set log_count [式{$log_count + 1}]
set_log_count $log_count
}else {
action_syslog msg "EEM: Max log instances ($max_log_instances)に達しました。ログの収集をスキップしています。
}

#常にスタックしているクライアントを認証解除する
cli_exec $fd "ワイヤレスクライアントmacアドレス$mac認証解除"
action_syslog msg "EEM: Deauthenticated client $mac"
}
}
}
}
cli_close $fd
出口0
—
####スクリプトの主な機能
1. **15分間隔**：要求に応じてウォッチドッグタイマーを900秒（15分）に設定
2. **Stuck threshold**:15分（900秒）を超えてスタックしたクライアントでのみトリガーします。
3. **ログ制限**:WLC + 1クライアントあたりのログを収集して**最大2つの合計インスタンス数を取得し**ログ収集をスキップする（クライアントの認証を解除する）
4. **WLCログ収集**：次の情報が含まれます。
– クライアントごとの詳細/サマリー
- WNCDプロセスログ（30分ウィンドウ）
– 完全な「show tech wireless」
5. **Persistent counter**:EEMスクリプトの実行全体で、「/bootflash/auth_log_count.txt」を介してログインスタンスを追跡

導入と検証
1. WLCにスクリプトを適用します。
WLC# configure terminal
WLC(config)# event manager applet AuthStuckHandler
```

WLC(config-applet)# event timer watchdog time 900
WLC(config-applet)# action 1 cliコマンド「sh bootflash:auth_stuck_eem.tcl」
WLC(config-applet)#終了
（または、完全なTclスクリプトをWLC EEM設定に直接貼り付けます）。

2. EEM登録を確認します。
WLC# show event manager policy registered（イベント管理ポリシーが登録されている場合）

3. 収集されたログの取得：
WLC# copy bootflash:auth_stuck_eem.log ftp:
WLC# copy bootflash:auth_log_count.txt ftp:

4. ログ・カウンタをリセットして、収集を再度有効にします（必要な場合）。
WLC# delete bootflash:auth_log_count.txt

結論

このドキュメントでは、検証されたTACの手法と実際の導入事例を統合し、最も一般的なCatalyst 9800メッシュWiFiの問題（不安定なバックホール、クライアントが認証状態のままになる、トラフィックが送信されない）を解決します。

報告されたメッシュ障害の90 %は孤立したハードウェアやクライアントの障害ではなく、コントロールプレーンとデータプレーンの状態の不一致、メッシュトポロジの不安定さ、最適でないRF設計などの症状が報告されていることが重要なポイントです。

翻訳について
シスコは世界中のユーザにそれぞれの言語でサポート コンテンツを提供するために、機械と人による翻訳を組み合わせて、本ドキュメントを翻訳しています。ただし、最高度の機械翻訳であっても、専門家による翻訳のような正確性は確保されません。シスコは、これら翻訳の正確性について法的責任を負いません。原典である英語版（リンクからアクセス可能）もあわせて参照することを推奨します。