

ゲートキーパーの TTL とエージング アウト処理の説明とトラブルシューティング

目次

[概要](#)

[前提条件](#)

[要件](#)

[使用するコンポーネント](#)

[表記法](#)

[存続可能時間に関する情報](#)

[Cisco Gatekeeper のデバッグ](#)

[関連情報](#)

概要

このドキュメントでは、複数の例で、Cisco ゲートキーパーが、存続可能時間 (TTL) 値を使用してエンドポイントをエージングアウトする仕組みを説明します。debug コマンドと show コマンドは、さまざまな方法で TTL がどのように動作するかを表示するために使用されます。

前提条件

要件

このドキュメントの読者には、次の項目に関する知識が必要です。

- Cisco ゲートキーパーを含む Cisco H.323 実装。H.323 ゲートキーパーの基礎知識については、『[H.323 ゲートキーパーについて](#)』を参照してください。

使用するコンポーネント

このドキュメントの情報は、次のソフトウェアとハードウェアのバージョンに基づくものです。

- このドキュメントの目的のため、Cisco IOS® ソフトウェア リリース 12.3(9) を使用して情報を収集します。H.323 ゲートキーパー機能を含む Cisco IOS 使用していることを確認してください。これは、Cisco IOS イメージ名に x が付加されることで示されます。たとえば、Cisco 3640 をゲートキーパーとして機能させるために有効な Cisco IOS は、c3640-ix-mz.123-9.bin です。
- Cisco ゲートキーパー (すべてのプラットフォーム)。注: このドキュメントの例では、NetMeeting が TTL 値を提供しないため、H.323 エンドポイントとして使用されていました。Cisco IOS ゲートウェイで NetMeeting を設定する方法については、『[Cisco IOS ゲートウェイで Microsoft NetMeeting を設定する方法](#)』を参照してください。

このドキュメントの情報は、特定のラボ環境にあるデバイスに基づいて作成されたものです。このドキュメントで使用するすべてのデバイスは、クリアな（デフォルト）設定で作業を開始しています。対象のネットワークが実稼働中である場合には、どのような作業についても、その潜在的な影響について確実に理解しておく必要があります。

表記法

ドキュメント表記の詳細は、『[シスコテクニカルティップスの表記法](#)』を参照してください。

存続可能時間に関する情報

Cisco ゲートキーパーで TTL がどのように動作し、さまざまなケースでエンドポイントのエージングプロセスがどのように動作するかについて、いくつかのヒントを次に示します。

- Cisco ゲートキーパーは、登録要求 (RRQ) メッセージ (ライトウェイトまたはフル) をエンドポイントから定期的に受信することを期待します。
- エンドポイントのタイムアウトについて、Cisco ゲートキーパーはエンドポイントが RRQ メッセージで指定した TTL 値をチェックします。エンドポイントのタイムアウトは、デフォルトで 1800 秒 (30 分) にハードコードされています。この値は、Cisco ゲートキーパーの CLI コマンド `endpoint ttl <time_value>` を使用して変更できます。これにより、すべての H.323 v1 エンドポイント、または RRQ メッセージに TTL 値が含まれない H.323 v2 以降のエンドポイントの動作が変更されます。
- Cisco ゲートキーパーは「エンドポイントのエージングプロセス」を定期的に行います。このプロセスの実行間隔は、現在の CPU 負荷に応じて異なります (1 ~ 5 分間隔)。エージングの間隔は、CPU 使用率の 20 % ごとに 1 分ずつ、最大 5 分まで増加します。エンドポイントの数が多いときに CPU の過負荷を防ぐため、エージングプロセスは 1 パスあたり 50 個のエンドポイントでのみ実行されます。これを超える場合、残りは次のタイマー起動まで保留されます。これは 1 ~ 5 分です。
- RRQ RAS (Registration, Admission and Status) メッセージに `timeToLive` フィールドが含まれている場合、ゲートキーパーはそのフィールドの値を使用してシステム デフォルトまたは `endpoint ttl <time_value>` ゲートキーパー CLI コマンドを使用して設定された値を上書きします。エンドポイントから応答を受信せずにその期間が経過すると、次のタイマー起動時にそのエンドポイントのクリーンアッププロセスが実行されます。ワースト ケースは、エンドポイントによって送信された TTL から 5 分が経過した場合 (Cisco ゲートキーパーの CPU 負荷が一貫して高い場合) です。より可能性の高いワースト ケース シナリオは、TTL タイムアウトから 1 分が経過した場合です。
- エンドポイントの RRQ メッセージに `timeToLive` フィールドが含まれていない場合、Cisco ゲートキーパーはエンドポイントが TTL をサポートしないものとして扱います。この場合、ゲートキーパーはそのエンドポイントから RRQ を受信しなくなったときに TTL タイムアウト (デフォルトの 1800 秒または `endpoint ttl` コマンドで指定された値) を実行します。その後、3 つの情報要求 (IRQ) をそれぞれ (ゲートキーパーの CPU 負荷に応じて) 1 ~ 5 分のいずれかの間隔で送信します。3 つの IRQ を送信した後で応答を受信しなかった場合、Cisco ゲートキーパーは最終的にそのエンドポイントを削除します。
- エンドポイントがアクティブ コール中である場合、そのエンドポイントはコールが終了するまでエージングアウトされません。
- Cisco ゲートキーパーは、コール中のエンドポイントから情報要求応答 (IRR) メッセージを受信することを期待します。Cisco ゲートキーパーは、コールの「guid」への参照を含む定期的な IRR メッセージを受信しなかった場合、4 分待ってから、そのコールの対象となる工

ンドポイントに IRQ を送信します。さらに 8 分が経過しても Cisco ゲートキーパーがそのコールに関する情報を受信しなかった場合、そのコールはクリーンアップされ、ゲートキーパーはエンドポイントに解除要求 (DRQ) を送信します。「ダングリグ」コールがクリーンアップされる (および帯域幅が解放される) までに、合計で約 12 分が経過します。このコール タイマーは設定可能ではありません。

- 代替の Cisco ゲートキーパーが所有するエンドポイントは、(このゲートキーパーが実際にそのエンドポイントを「所有」していないので) 直接エージングアウトされません。
- ([alias static xxxxx](#) コンフィギュレーション コマンドで作成された) スタティック エンドポイントは、エージングアウトされません。

Cisco Gatekeeper のデバッグ

TTL がどのように動作するかをさまざまな方法で確認するために使用できる `show` および `debug` コマンドの例を次に示します。

- [show gatekeeper endpoints](#)
- [debug ras](#)
- [debug h225 asn1](#)

次の 2 つの項では、Cisco ゲートキーパーが異なる TTL 値を使用してエンドポイントをエージングアウトする 2 つのケースについて説明します。

ケース 1

次の出力は、Cisco ゲートキーパーから取得した [debug ras](#) および [debug h225 asn1](#) コマンドの出力です。デバッグでは、ゲートウェイの TTL 値は 60 秒になっています。Cisco ゲートキーパーは、デフォルトや設定されたエンドポイントの TTL 値に関係なく、登録確認 (RCF) メッセージでこれを確認し、受け入れます。これは、エンドポイントに TTL 値が含まれているためです。

```
Mar 2 23:52:50.797: RAS INCOMING ENCODE BUFFER ::= 0E 400FD206 0008914A
00030000 0100AC10 0D2AE26A 00040067 006B0062 0
02D0032 00B50000 12128F00 02003B01 80211E00 36003100 36004600 32004400
43004300 30003000 30003000 30003000 30003101 00
0180
Mar 2 23:52:50.797:
Mar 2 23:52:50.797: RAS INCOMING PDU ::=
```

```
value RasMessage ::= registrationRequest : { requestSeqNum 4051 protocolIdentifier { 0 0 8 2250
0 3 } discoveryComplete FALSE callSignalAddress { } rasAddress { ipAddress : { ip 'AC100D2A'H
port 57962 } } terminalType { mc FALSE undefinedNode FALSE } gatekeeperIdentifier {"gkb-2"}
endpointVendor { vendor { t35CountryCode 181 t35Extension 0 manufacturerCode 18 } } timeToLive
60 !--- TTL value. keepAlive TRUE endpointIdentifier {"616F2DCC00000001"} willSupplyUIEs FALSE
maintainConnection TRUE } Mar 2 23:52:50.805: RRQ (seq# 4051) rcvd Mar 2 23:52:50.805: RAS
OUTGOING PDU ::= value RasMessage ::= registrationConfirm : { requestSeqNum 4051
protocolIdentifier { 0 0 8 2250 0 3 } callSignalAddress { } gatekeeperIdentifier {"gkb-2"}
endpointIdentifier {"616F2DCC00000001"} alternateGatekeeper { { rasAddress ipAddress : { ip
'AC100D29'H port 1719 } gatekeeperIdentifier {"gkb-1"} needToRegister TRUE priority 0 } }
timeToLive 60 willRespondToIRR FALSE maintainConnection TRUE } Mar 2 23:52:50.813: RAS OUTGOING
ENCODE BUFFER ::= 12 400FD206 0008914A 00030008 0067006B 0062002D 00321E00 36003100 3 6004600
32004400 43004300 30003000 30003000 30003000 3000310F 8A140140 AC100D29 06B70800 67006B00
62002D00 31800200 3B 010001 80 Mar 2 23:52:50.813: Mar 2 23:52:50.817: IPSOCK_RAS_sendto: msg
length 86 from 172.16.13.16:1719 to 172.16.13.42: 57962 Mar 2 23:52:50.817: RASLib::RASsendRCF:
RCF (seq# 4051) sent to 172.16.13.42
```

ケース2

もう1つの例を次に示します。この例では、RRQ メッセージで TTL 値を送信しなかったエンドポイントが、120 秒 (ゲートキーパーに設定されている値) が経過する前にライトウェイト RRQ を送信するように通知されています。Cisco ゲートキーパーは、登録解除要求 (URQ) メッセージを受信した場合でも、3 つの IRQ メッセージが未応答になるまでエンドポイントを削除しないことが、この出力で確認できます。IRQ 間の間隔は 1 ~ 5 分です。

```
gka-1#show logging Syslog logging: enabled (0 messages dropped, 0 messages rate-limited, 0
flushes, 0 overruns) Console logging: disabled Monitor logging: level debugging, 1076 messages
logged Buffer logging: level debugging, 4257 messages logged Logging Exception size (4096 bytes)
Trap logging: level informational, 60 message lines logged Log Buffer (9999999 bytes): Mar 14
06:28:31.771: RAS INCOMING ENCODE BUFFER::= 0C 80000006 0008914A 00020001 00AB4555 BF06B801
00AB4555 BF05C502 00014007 006B0065 00740070 00610074 0065006C 60B50053 4C164D69 63726F73
6F6674AE 204E6574 4D656574 696E67AE 0003332E 3000 Mar 14 06:28:31.783: Mar 14 06:28:31.787: RAS
INCOMING PDU ::= value RasMessage ::= registrationRequest : { requestSeqNum 1 protocolIdentifier
{ 0 0 8 2250 0 2 } discoveryComplete FALSE callSignalAddress { ipAddress : { ip 'AB4555BF'H port
1720 } } rasAddress { ipAddress : { ip 'AB4555BF'H port 1477 } } terminalType { terminal { } mc
FALSE undefinedNode FALSE } terminalAlias { h323-ID : {"ketpatel"} } endpointVendor { vendor {
t35CountryCode 181 t35Extension 0 manufacturerCode 21324 } productId
'4D6963726F736F6674AE204E65744D656574696E...'H versionId '332E3000'H } } Mar 14 06:28:31.811:
RAS OUTGOING PDU ::= value RasMessage ::= registrationConfirm : { requestSeqNum 1
protocolIdentifier { 0 0 8 2250 0 3 } callSignalAddress { } terminalAlias { h323-ID :
{"ketpatel"} } gatekeeperIdentifier {"gka-1"} endpointIdentifier {"81F6A89800000001"}
alternateGatekeeper { } timeToLive 120 willRespondToIRR FALSE maintainConnection FALSE } Mar 14
06:28:31.823: RAS OUTGOING ENCODE BUFFER::= 12 C0000006 0008914A 00030001 4007006B 00650074
00700061 00740065 006C0800 67006B00 61002D00 31E0038 00310046 00360041 00380039 00380030
00300030 00300030 00300030 00310F8A 01000200 77010001 00 gka-1#show gatekeeper endpoints
GATEKEEPER ENDPOINT REGISTRATION ===== CallSignalAddr Port
RASSignalAddr Port Zone Name Type Flags -----
-- ----- 171.69.85.191 1720 171.69.85.191 1477 gka-1 TERM H323-ID: ketpatel Total number of
active registrations = 1 Mar 14 06:28:31.835: Mar 14 06:28:31.835: RAS OUTGOING PDU ::= value
RasMessage ::= infoRequest : { requestSeqNum 70 callReferenceValue 0 callIdentifier { guid
'00000000000000000000000000000000'H } } Mar 14 06:28:31.839: RAS OUTGOING ENCODE BUFFER::= 56
00004500 000B0011 00000000 00000000 00000000 00000000 00 Mar 14 06:28:31.843: Mar 14
06:28:31.847: RAS INCOMING ENCODE BUFFER::= 58 80004502 03C00038 00310046 00360041 00380039
00380030 00300030 00300030 00300030 003100AB 4555BF05 C50100AB 4555BF06 B8024007 006B0065
00740070 00610074 0065006C 4007006B 00650074 00700061 00740065 006C Mar 14 06:28:31.859: Mar 14
06:28:31.859: RAS INCOMING PDU ::= value RasMessage ::= infoRequestResponse : { requestSeqNum 70
endpointType { terminal { } mc FALSE undefinedNode FALSE } endpointIdentifier
{"81F6A89800000001"} rasAddress ipAddress : { ip 'AB4555BF'H port 1477 } callSignalAddress {
ipAddress : { ip 'AB4555BF'H port 1720 } } endpointAlias { h323-ID : {"ketpatel"}, h323-ID :
{"ketpatel"} } } Mar 14 06:30:42.208: RAS OUTGOING PDU ::= value RasMessage ::= infoRequest : {
requestSeqNum 71 callReferenceValue 0 callIdentifier { guid '00000000000000000000000000000000'H
} } Mar 14 06:30:42.212: RAS OUTGOING ENCODE BUFFER::= 56 00004600 000B0011 00000000 00000000
00000000 00000000 00 Mar 14 06:30:42.216: Mar 14 06:30:42.216: RAS INCOMING ENCODE BUFFER::= 58
80004602 03C00038 00310046 00360041 00380039 00380030 00300030 00300030 00300030 003100AB
4555BF05 C50100AB 4555BF06 B8024007 006B0065 00740070 00610074 0065006C 4007006B 00650074
00700061 00740065 006C Mar 14 06:30:42.228: Mar 14 06:30:42.232: RAS INCOMING PDU ::= value
RasMessage ::= infoRequestResponse : { requestSeqNum 71 endpointType { terminal { } mc FALSE
undefinedNode FALSE } endpointIdentifier {"81F6A89800000001"} rasAddress ipAddress : { ip
'AB4555BF'H port 1477 } callSignalAddress { ipAddress : { ip 'AB4555BF'H port 1720 } }
endpointAlias { h323-ID : {"ketpatel"}, h323-ID : {"ketpatel"} } } Mar 14 06:32:05.938: RAS
INCOMING ENCODE BUFFER::= 19 40000101 00AB4555 BF06B802 4007006B 00650074 00700061 00740065
006C4007 006B0065 00740070 00610074 0065006C 1E003800 31004600 36004100 38003900 38003000
30003000 30003000 30003000 31 Mar 14 06:32:05.950: Mar 14 06:32:05.950: RAS INCOMING PDU ::=
value RasMessage ::= unregistrationRequest : { requestSeqNum 2 callSignalAddress { ipAddress : {
ip 'AB4555BF'H port 1720 } } endpointAlias { h323-ID : {"ketpatel"}}, h323-ID : {"ketpatel"} } }
endpointIdentifier {"81F6A89800000001"} } Mar 14 06:32:05.962: RAS OUTGOING PDU ::= value
RasMessage ::= unregistrationConfirm : { requestSeqNum 2 } Mar 14 06:32:05.962: RAS OUTGOING
ENCODE BUFFER::= 1C 0001 Mar 14 06:32:05.966: gka-1#show gatekeeper endpoints GATEKEEPER
ENDPOINT REGISTRATION ===== CallSignalAddr Port RASSignalAddr Port
```

```

Zone Name Type Flags -----
171.69.85.191 1720 171.69.85.191 1477 gka-1 TERM Total number of active registrations = 1 Mar 14
06:33:42.223: RAS OUTGOING PDU ::= value RasMessage ::= infoRequest : { requestSeqNum 72
callReferenceValue 0 callIdentifier { guid '00000000000000000000000000000000'H } } Mar 14
06:33:42.227: RAS OUTGOING ENCODE BUFFER::= 56 00004700 000B0011 00000000 00000000 00000000
00000000 00 Mar 14 06:33:42.231: Mar 14 06:34:42.234: RAS OUTGOING PDU ::= value RasMessage ::=
infoRequest : { requestSeqNum 73 callReferenceValue 0 callIdentifier { guid
'00000000000000000000000000000000'H } } Mar 14 06:34:42.238: RAS OUTGOING ENCODE BUFFER::= 56
00004800 000B0011 00000000 00000000 00000000 00000000 00 Mar 14 06:34:42.242: Mar 14
06:35:42.244: RAS OUTGOING PDU ::= value RasMessage ::= infoRequest : { requestSeqNum 74
callReferenceValue 0 callIdentifier { guid '00000000000000000000000000000000'H } } Mar 14
06:35:42.248: RAS OUTGOING ENCODE BUFFER::= 56 00004900 000B0011 00000000 00000000 00000000
00000000 00 Mar 14 06:35:42.252: gka-1# gka-1#show gatekeeper endpoints GATEKEEPER ENDPOINT
REGISTRATION ===== CallSignalAddr Port RASignalAddr Port Zone Name
Type Flags ----- Total number of
active registrations = 0

```

関連情報

- [Cisco ハイパフォーマンス ゲートキーパー](#)
- [H.323 バージョン 2 のサポート](#)
- [ゲートキーパー登録問題のトラブルシューティング](#)
- [音声に関する技術サポート](#)
- [音声とユニファイド コミュニケーションに関する製品サポート](#)
- [Cisco IP Telephony のトラブルシューティング](#)
- [テクニカルサポート - Cisco Systems](#)