

Présentation et dépannage de la durée de vie et du processus de vieillissement du contrôleur d'accès

Contenu

[Introduction](#)

[Conditions préalables](#)

[Conditions requises](#)

[Composants utilisés](#)

[Conventions](#)

[Conseils de Time to Live](#)

[Debugs de Cisco Gatekeeper](#)

[Informations connexes](#)

Introduction

Ce document décrit comment Cisco Gatekeeper vieillit les points finaux utilisant la valeur du Time to Live (TTL) dans différents cas. Des debugs et les **commandes show** sont utilisés d'afficher comment le TTL fonctionne dans diverses manières.

Conditions préalables

Conditions requises

Les lecteurs de ce document doivent prendre connaissance des rubriques suivantes :

- De Cisco implémentation H.323, y compris Cisco Gatekeeper. Pour une compréhension de base des Contrôleurs d'accès H.323, référez-vous [compréhension derrière des Contrôleurs d'accès H.323](#).

Composants utilisés

Les informations dans ce document sont basées sur les versions de logiciel et matériel suivantes :

- Afin de ce document, la version de logiciel 12.3(9) de Cisco IOS® est utilisée pour collecter les informations. Assurez-vous que vous utilisez le Cisco IOS avec la configuration de fonctionnalité de contrôleur d'accès H.323. Ceci est dénoté avec un x dans le nom d'image de Cisco IOS. Par exemple, le Cisco IOS valide pour que le Cisco 3640 agisse en tant que garde-porte est c3640-ix-mz.123-9.bin.
- Cisco Gatekeeper (toutes les Plateformes). **Remarque:** NetMeeting a été utilisé comme H.323

point final dans l'exemple dans ce document parce qu'il ne fournit pas une valeur de TTL. Pour les informations sur la façon dont configurer NetMeeting avec des passerelles de Cisco IOS, référez-vous [Comment-à configurer la Microsoft NetMeeting avec des passerelles de Cisco IOS](#).

Les informations présentées dans ce document ont été créées à partir de périphériques dans un environnement de laboratoire spécifique. Tous les périphériques utilisés dans ce document ont démarré avec une configuration effacée (par défaut). Si vous travaillez dans un réseau opérationnel, assurez-vous de bien comprendre l'impact potentiel de toute commande avant de l'utiliser.

[Conventions](#)

Pour plus d'informations sur les conventions des documents, référez-vous aux [Conventions utilisées pour les conseils techniques de Cisco](#).

[Conseils de Time to Live](#)

Ce sont quelques conseils sur la façon dont le TTL travaille à Cisco Gatekeeper et la façon dont les travaux par processus vieillissants pour les points finaux dans différents cas.

- Cisco Gatekeeper compte entendre du point final périodiquement par l'intermédiaire des messages de la demande d'enregistrement (RRQ) (léger ou plein).
- Pour des délais d'attente de point final, Cisco Gatekeeper prête l'attention à la valeur de TTL fournie par le point final dans le message RRQ. Il y a un par défaut dur codé de 1800 secondes (trente minutes) pour le délai d'attente de point final. Cette valeur peut être changée avec le `<time_value>` d'[endpoint ttl](#) de commande de Cisco Gatekeeper CLI. Ceci change le comportement pour tous H.323 les points finaux v1, ou H.323 v2 et points finaux postérieurs qui n'incluent pas la valeur de TTL dans le message RRQ.
- Cisco Gatekeeper exécute un « processus vieillissant de point final » périodiquement. Ce processus varie basé sur le chargement en cours CPU de chacun minute à toutes les cinq minutes. Pour tous les vingt pour cent d'utilisation du processeur, l'intervalle vieillissant augmente d'une minute, jusqu'à un maximum de cinq minutes. Pour ne pas surcharger la CPU quand il y a beaucoup de points finaux, le processus vieillissant fonctionne seulement par cinquante points finaux par passage. S'il y a plus, alors ceux sont reportés jusqu'au prochain bruit de temporisateur. Ceci peut être d'une à cinq minutes.
- Si le champ de Time to Live est inclus dans le message d'enregistrement, d'admission et d'état RRQ (RAS), les utilisations de garde-porte que la valeur du champ pour ignorer les paramètres systèmes par défaut ou la valeur a configurées utilisant la commande CLI de garde-porte de `<time_value>` d'[endpoint ttl](#). Une fois que ce délai prévu s'écoule sans entendre du point final, le prochain bruit de temporisateur passe par le processus de nettoyage pour ce point final. Le pire des cas est le TTL envoyé par le point final plus cinq minutes (si Cisco Gatekeeper est uniformément sous le chargement lourd CPU). Un scénario plus probable de le pire des cas est le délai d'attente TTL plus une minute.
- Quand le point final n'inclut pas le champ de Time to Live dans le message RRQ, Cisco Gatekeeper traite le point final comme si il ne prend en charge pas le TTL. Dans ce cas, une fois que le garde-porte ne reçoit plus RRQs de ce point final, il fait le délai d'attente TTL (le par défaut de 1800 secondes, ou la valeur spécifique dans la commande d'[endpoint ttl](#)). Alors il envoie trois demandes de l'information (IRQs) avec des intervalles n'importe où d'un à cinq

- minute chacun (basé sur le chargement CPU du garde-porte). Après qu'il envoie trois IRQs, et ne reçoive pas n'importe quelle réponse, Cisco Gatekeeper retire finalement le point final.
- Si le point final est dans un appel actif, il n'est pas vieilli jusqu'à ce que l'appel soit terminé.
 - Cisco Gatekeeper compte entendre des points finaux impliqués dans un appel du message de la réponse de l'information (IRR). Si Cisco Gatekeeper ne reçoit pas les messages périodiques IRR qui contiennent des références au « guid » pour un appel, le garde-porte attend quatre minutes, et puis envoie un IRQ aux points finaux que l'appel est pour. Si, après huit minutes supplémentaires, Cisco Gatekeeper n'a toujours entendu rien au sujet de cet appel, l'appel est nettoyé et le garde-porte envoie des demandes de désengagement (DRQs) aux points finaux. Un total d'approximativement douze minutes des passages avant un appel « balançant » est nettoyés (et la bande passante libérée). Ce temporisateur d'appel n'est pas configurable.
 - Des points finaux qui sont possédés par Cisco Gatekeeper alternatif ne sont pas vieillis directement (puisque ce garde-porte réellement « ne possède pas » le point final).
 - Des points finaux statiques (créés avec l'[alias static](#) `xxxxx` de commande de configuration) ne sont pas vieillis.

Debugs de Cisco Gatekeeper

Ce sont certaines des commandes d'**exposition** et de **débogage** que vous pouvez employer pour vérifier comment le TTL fonctionne dans diverses manières :

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

Ces deux sections discutent deux cas où Cisco Gatekeeper vieillit les points finaux utilisant différentes valeurs de TTL.

Affaire 1

Cette sortie est du [debug ras](#) et des commandes de [debug h225 asn1](#) et est prise de Cisco Gatekeeper. Dans le débogage, la passerelle a une valeur de TTL de 60 secondes. Cisco Gatekeeper confirme et reçoit ceci dans son message de la confirmation d'enregistrement (RCF), n'importe ce que sa valeur de TTL de par défaut ou de point d'extrémité configuré est. C'est parce que le point final inclut une valeur de TTL.

```
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"} willSupplyUIIEs 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
```

Affaire 2

C'est un autre exemple où un point final qui n'a pas envoyé une valeur de TTL dans son message RRQ a été annoncé pour envoyer un RRQ léger avant 120 secondes, qui est la valeur configurée sur le garde-porte. Vous pouvez voir dans cette sortie comment Cisco Gatekeeper ne supprime pas le point final jusqu'à trois messages sans réponse IRQ, quoiqu'un message de la demande d'Unregistration (URQ) ait été reçu. Les temps entre l'IRQ ont lieu entre une et cinq minutes.

```
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 311E0038 00310046 00360041 00380039 00380030
00300030 00300030 00300030 00310F8A 01000200 77010001 00 gka-1#show gatekeeper endpoints
GATEKEEPER ENDPOINT REGISTRATION ===== CallSignalAddr Port
RASignalAddr 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 RASignalAddr 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

```

[Informations connexes](#)

- [Garde-porte de haute performance de Cisco](#)
- [Support H.323 version 2](#)
- [Dépannage des problèmes liés à l'enregistrement du contrôleur d'accès](#)
- [Assistance technique concernant la technologie vocale](#)
- [Assistance concernant les produits vocaux et de communications unifiées](#)
- [Dépannage des problèmes de téléphonie IP Cisco](#)
- [Support technique - Cisco Systems](#)