

Introducción al control de acceso TTL y resolución de problemas en el proceso de desactualización

Contenido

[Introducción](#)

[prerrequisitos](#)

[Requisitos](#)

[Componentes Utilizados](#)

[Convenciones](#)

[Consejos](#)

[Depuraciones de Cisco Gatekeeper](#)

[Información Relacionada](#)

[Introducción](#)

Este documento describe cómo el gatekeeper de Cisco envejece hacia fuera los puntos finales usando el valor del Time to Live (TTL) en diversos casos. Se utilizan los comandos de debugs y show de mostrar cómo TTL trabaja en las distintas maneras.

[prerrequisitos](#)

[Requisitos](#)

Quienes lean este documento deben tener conocimiento de los siguientes temas:

- Implementación de Cisco H.323, incluyendo el gatekeeper de Cisco. Para una comprensión básica de Gatekeepers H.323, refiera [comprensión de Gatekeepers H.323](#).

[Componentes Utilizados](#)

La información que contiene este documento se basa en estas versiones de software y hardware.

- Con el fin de este documento, el Software Release 12.3(9) de Cisco IOS® se utiliza para recoger la información. Asegúrese de que usted utilice el Cisco IOS con la característica de la funcionalidad de gatekeeper de H.323. Esto se denota con un x en el nombre de la imagen del Cisco IOS. Por ejemplo, el Cisco IOS válido para que el Cisco 3640 actúe como portero es c3640-ix-mz.123-9.bin.
- Gatekeeper de Cisco (todas las Plataformas). **Nota:** El NetMeeting fue utilizado como punto final de H.323 en el ejemplo en este documento porque no proporciona un valor de TTL. Para

la información sobre cómo configurar el NetMeeting con los gateways del Cisco IOS, refiera al [NetMeeting de Microsoft de la configuración del Cómo con los gateways del Cisco IOS](#).

La información que se presenta en este documento se originó a partir de dispositivos dentro de un ambiente de laboratorio específico. Todos los dispositivos que se utilizan en este documento se pusieron en funcionamiento con una configuración verificada (predeterminada). Si la red está funcionando, asegúrese de haber comprendido el impacto que puede tener un comando antes de ejecutarlo.

[Convenciones](#)

Para obtener más información sobre las convenciones del documento, consulte [Convenciones de Consejos Técnicos de Cisco](#).

[Consejos](#)

Éstas son algunas extremidades en cómo TTL trabaja en un gatekeeper de Cisco y cómo los trabajos de proceso de desactualización para los puntos finales en diversos casos.

- El gatekeeper de Cisco espera oír del punto final periódicamente vía los mensajes del Solicitud de Inscripción (RRQ) (ligero o lleno).
- Para los tiempos de espera del punto final, el gatekeeper de Cisco presta la atención al valor de TTL suministrado por el punto final en el mensaje RPQ. Hay un valor por defecto difícilmente cifrado de 1800 segundos (treinta minutos) para el tiempo de espera del punto final. Este valor se puede cambiar con el gatekeeper de Cisco. Esto cambia el comportamiento para todos los puntos finales del v1 de H.323, o los puntos finales del H.323v2 y posterior que no incluyen el valor de TTL en el mensaje RPQ.
- El gatekeeper de Cisco funciona con un “proceso de desactualización del punto final” periódicamente. Este proceso varía basado en la corriente carga de la CPU todos minutos a cada cinco minutos. Para el cada veinte por ciento de utilización de la CPU, el intervalo del envejecimiento aumenta en un minuto, hasta un máximo de cinco minutos. Para no sobrecargar el CPU cuando hay muchos puntos finales, el proceso de desactualización se ejecuta solamente a través de cincuenta puntos finales por el paso. Si hay más, después éstos se difieren hasta el estallido siguiente del temporizador. Éste puede ser a partir un a cinco minutos.
- Si el campo timeToLive se incluye en el mensaje del registro de RRQ, de la admisión y del estatus (RAS), las aplicaciones del portero que el valor del campo para reemplazar el valor predeterminado del sistema o el valor configuró usando el comando gatekeeper cli del `<time_value>` del [punto final TTL](#). Una vez que ese período de tiempo transcurre sin la audición del punto final, el estallido siguiente del temporizador pasa con el proceso de la limpieza para ese punto final. El peor caso es TTL enviado por el punto final más cinco minutos (si el gatekeeper de Cisco está constantemente bajo pesado carga de la CPU). Un peor de los casos más probable es el tiempo de espera TTL más un minuto.
- Cuando el punto final no incluye el campo timeToLive en el mensaje RPQ, el gatekeeper de Cisco trata el punto final como si no soporte TTL. En este caso, una vez que el portero recibe no más los RRQ de ese punto final, hace el tiempo de espera TTL (el valor por defecto de 1800 segundos, o el valor especificado en el [comando endpoint ttl](#)). Entonces envía tres peticiones de la información (IRQ) con los intervalos de dondequiera a partir el uno a cinco anota cada uno (basado en carga de la CPU del portero). Después de que envíe tres IRQ, y

- no reciba ninguna respuesta, el gatekeeper de Cisco finalmente quita el punto final.
- Si el punto final está en una llamada activa, no se envejece hacia fuera hasta que se termine la llamada.
- El gatekeeper de Cisco espera oír de los puntos finales implicados en una llamada con el mensaje del Information Response (IRR). Si el gatekeeper de Cisco no recibe los mensajes IRR periódicos que contienen las referencias al “guid” para una llamada, el portero espera cuatro minutos, y después envía un IRQ a los puntos finales que la llamada está para. Si, después de ocho más minutos, el gatekeeper de Cisco todavía no ha oído cualquier cosa sobre esa llamada, se limpia la llamada y el portero envía los pedidos de desconexión (DRQ) a los puntos finales. Un total de aproximadamente doce anota los pasajes antes de una llamada “que cuelga” se limpia (y el ancho de banda liberado). Este temporizador de la llamada no es configurable.
- Los puntos finales que son poseídos por un gatekeeper de Cisco alterno no se envejecen hacia fuera directamente (puesto que este portero “no posee realmente” el punto final).
- Los puntos finales estáticos (creados con el **xxxxx [estático del](#)** comando configuration [alias](#)) no se envejecen hacia fuera.

Depuraciones de Cisco Gatekeeper

Éstos son algunos de los **comandos show and debug** que usted puede utilizar para verificar cómo TTL trabaja en las distintas maneras:

- [show gatekeeper endpoints](#)
- [haga el debug de los ras](#)
- [debug h225 asn1](#)

Estas dos secciones discuten dos casos donde el gatekeeper de Cisco envejece hacia fuera los puntos finales usando diversos valores de TTL.

Caso 1

Esta salida es de y se toma de un [gatekeeper de Cisco](#). En el debug, el gateway tiene un valor de TTL de 60 segundos. El gatekeeper de Cisco confirma y valida esto en su mensaje de Confirmación de Inscripción (RCF), no importa qué es su valor de TTL del valor por defecto o del punto final configurado. Esto es porque el punto final incluye un valor 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"} willSupplyUUIEs 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
```

Caso 2

Éste es otro ejemplo donde un punto final que no envió un valor de TTL en su mensaje RPQ se ha notificado para enviar un RRQ ligero antes de 120 segundos, que es el valor configurado en el portero. Usted puede ver en esta salida cómo el gatekeeper de Cisco no borra el punto final hasta tres mensajes por contestar IRQ, aunque un mensaje del Unregistration Request (URQ) fue recibido. Los tiempos entre el IRQ son entre un y cinco minutos.

```
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

```

[Información Relacionada](#)

- [Gatekeeper de alto rendimiento de Cisco](#)
- [Soporte H.323 versión 2](#)
- [Resolución de problemas relacionados al registro del Gatekeeper](#)
- [Soporte de tecnología de voz](#)
- [Soporte de Productos de Voice and Unified Communications](#)
- [Troubleshooting de Cisco IP Telephony](#)
- [Soporte Técnico - Cisco Systems](#)