

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

Contenido

[Introducción](#)

[Prerequisites](#)

[Requirements](#)

[Componentes Utilizados](#)

[Convenciones](#)

[Consejos](#)

[Depuraciones de Cisco Gatekeeper](#)

[Información Relacionada](#)

Introducción

Este documento describe cómo Cisco Gatekeeper desactualiza los terminales utilizando el valor de Tiempo de vida (TTL) en casos diferentes. Los comandos Debugs y **show** se utilizan para mostrar cómo funciona el TTL de varias maneras.

Prerequisites

Requirements

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

- Implementación de Cisco H.323, incluido Cisco Gatekeeper. Para obtener una comprensión básica de los Gatekeepers H.323, refiérase a [Comprensión de los Gatekeepers H.323](#).

Componentes Utilizados

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

- A los efectos de este documento, se utiliza Cisco IOS® Software Release 12.3(9) para recopilar la información. Asegúrese de utilizar Cisco IOS con la función H.323 Gatekeeper. Esto se denota con una **x** en el nombre de la imagen de Cisco IOS. Por ejemplo, un Cisco IOS válido para que el Cisco 3640 actúe como gatekeeper es c3640-**ix**-mz.123-9.bin.
- Cisco Gatekeeper (todas las plataformas). **Nota:** NetMeeting se utilizó como punto final H.323 en el ejemplo de este documento porque no proporciona un valor TTL. Para obtener información sobre cómo configurar NetMeeting con Gateways de Cisco IOS, refiérase a

[Cómo Configurar NetMeeting de Microsoft con Gateways de Cisco IOS.](#)

La información que se presenta en este documento se originó a partir de dispositivos dentro de un ambiente de laboratorio específico. All of the devices used in this document started with a cleared (default) configuration. 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](#)

Estos son algunos consejos sobre cómo funciona TTL en un gatekeeper de Cisco y cómo funciona el proceso de envejecimiento para los terminales en diferentes casos.

- El gatekeeper de Cisco espera escuchar periódicamente desde el terminal a través de los mensajes de solicitud de registro (RRQ) (ligero o completo).
- Para los tiempos de espera de los terminales, el gatekeeper de Cisco presta atención al valor TTL proporcionado por el punto final en el mensaje RRQ. Hay un valor predeterminado codificado de 1800 segundos (treinta minutos) para el tiempo de espera del terminal. Este valor se puede cambiar con el comando Cisco Gatekeeper CLI [endpoint ttl <time_value>](#). Esto cambia el comportamiento de todos los terminales H.323 v1, o los terminales H.323 v2 y posteriores que no incluyen el valor TTL en el mensaje RRQ.
- Cisco Gatekeeper ejecuta periódicamente un "proceso de envejecimiento de terminales". Este proceso varía según la carga actual de la CPU de cada minuto a cada cinco minutos. Por cada veinte por ciento del uso de la CPU, el intervalo de caducidad aumenta en un minuto, hasta un máximo de cinco minutos. Para no sobrecargar la CPU cuando hay muchos puntos finales, el proceso de envejecimiento sólo se ejecuta a través de cincuenta puntos finales por paso. Si hay más, se posponen hasta la siguiente ventana emergente del temporizador. Puede ser de uno a cinco minutos.
- Si el campo timeToLive se incluye en el mensaje RRQ Registration, Admission and Status (RAS), el control de acceso utiliza el valor de ese campo para reemplazar el valor predeterminado del sistema o el valor configurado usando el comando de CLI [terminal ttl <time_value>](#) del gatekeeper. Una vez que ese período de tiempo transcurre sin que se oiga del terminal, la siguiente ventana emergente del temporizador pasa por el proceso de limpieza para ese terminal. El peor caso es el TTL enviado por el terminal más cinco minutos (si el gatekeeper de Cisco se encuentra sistemáticamente bajo una carga pesada de CPU). Un escenario más probable en el peor de los casos es el tiempo de espera TTL más un minuto.
- Cuando el punto final no incluye el campo timeToLive en el mensaje RRQ, el gatekeeper de Cisco trata al punto final como si no admitiera TTL. En este caso, una vez que el gatekeeper ya no recibe RRQ de ese punto final, realiza el tiempo de espera TTL (el valor predeterminado de 1800 segundos, o el valor especificado en el comando [endpoint ttl](#)). A continuación, envía tres solicitudes de información (IRQ) con intervalos de entre uno y cinco minutos cada una (en función de la carga de CPU del control de acceso). Después de enviar tres IRQ y de no recibir ninguna respuesta, Cisco Gatekeeper finalmente elimina el terminal.
- Si el terminal se encuentra en una llamada activa, no caducará hasta que se finalice la

llamada.

- El gatekeeper de Cisco espera oír hablar de los terminales involucrados en una llamada con el mensaje de respuesta de información (IRR). Si el gatekeeper de Cisco no recibe mensajes IRR periódicos que contienen referencias al "guid" para una llamada, el gatekeeper espera cuatro minutos y luego envía una IRQ a los extremos para los que está llamada. Si, después de ocho minutos más, el gatekeeper de Cisco todavía no ha oído nada acerca de esa llamada, la llamada se limpia y el gatekeeper envía solicitudes de desconexión (DRQ) a los terminales. Pasan aproximadamente doce minutos antes de que se elimine una llamada "colgante" (y se libere el ancho de banda). Este temporizador de llamadas no se puede configurar.
- Los terminales que son propiedad de un gatekeeper de Cisco alternativo no se desactualizan directamente (ya que este gatekeeper en realidad no "posee" el terminal).
- Los extremos estáticos (creados con el comando de configuración [alias static xxxxx](#)) no están desactualizados.

Depuraciones de Cisco Gatekeeper

Estos son algunos de los comandos **show** y **debug** que puede utilizar para verificar cómo funciona el TTL de varias maneras:

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

Estas dos secciones tratan de dos casos en los que Cisco Gatekeeper desactualiza los terminales utilizando diferentes valores TTL.

Caso 1

Este resultado proviene de los comandos [debug ras](#) y [debug h225 asn1](#) y se toma de un Cisco Gatekeeper. En la depuración, el gateway tiene un valor TTL de 60 segundos. El gatekeeper de Cisco lo confirma y acepta en su mensaje de confirmación de registro (RCF), sin importar cuál sea su valor TTL de terminal predeterminado o configurado. Esto se debe a que el terminal incluye un valor 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 40FD206 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

Este es otro ejemplo donde se ha notificado a un punto final que no envió un valor TTL en su mensaje RRQ que envíe un RRQ ligero antes de 120 segundos, que es el valor configurado en el control de acceso. Puede ver en este resultado cómo Cisco Gatekeeper no elimina el punto final hasta que se recibieron tres mensajes IRQ sin respuesta, aunque se recibió un mensaje de solicitud de desregistro (URQ). Las horas entre la IRQ son de uno a 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	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

[Información Relacionada](#)

- [Gatekeeper de alto rendimiento de Cisco](#)
- [Soporte de la versión 2 de H.323](#)
- [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](#)