

Gatekeeper TTL-verwerking en probleemoplossing

Inhoud

[Inleiding](#)

[Voorwaarden](#)

[Vereisten](#)

[Gebruikte componenten](#)

[Conventies](#)

[Tijd om tips te leven](#)

[Cisco Gatekeeper debugs](#)

[Gerelateerde informatie](#)

[Inleiding](#)

Dit document beschrijft hoe de Cisco Gatekeeper de eindpunten uitsteekt met de Time to Live (TTL) waarde in verschillende gevallen. Opdrachten en opdrachten worden gebruikt om aan te tonen hoe de TTL-serie op verschillende manieren werkt.

[Voorwaarden](#)

[Vereisten](#)

Lezers van dit document zouden op de hoogte moeten zijn van deze onderwerpen:

- Cisco H.323-implementatie, inclusief Cisco Gatekeeper. Voor een basisbegrip van H.323-gatekeeper, zie [begrip H.323-gatekeeper](#).

[Gebruikte componenten](#)

De informatie in dit document is gebaseerd op deze software- en hardwareversies.

- Voor dit document wordt Cisco IOS® softwarerelease 12.3(9) gebruikt om de informatie te verzamelen. Zorg ervoor dat u Cisco IOS gebruikt met de functie H.323 Gatekeeper functionaliteit. Dit wordt aangegeven met een x in de Cisco IOS-beeldnaam. Bijvoorbeeld, een geldig Cisco IOS voor Cisco 3640 om als gatekeeper op te treden is c3640-ix-mz.123-9.bin.
- Cisco Gatekeeper (alle platforms). **Opmerking:** NetMeeting is in het voorbeeld in dit document gebruikt als een H.323-eindpunt omdat het geen TTL-waarde biedt. Raadpleeg voor informatie over het configureren van NetMeeting met Cisco IOS gateways [hoe u Microsoft NetMeeting met Cisco IOS gateways kunt configureren](#).

De informatie in dit document is gebaseerd op apparaten in een specifieke laboratoriumomgeving.

Alle apparaten die in dit document worden beschreven, hadden een opgeschoonde (standaard)configuratie. Als u in een levend netwerk werkt, zorg er dan voor dat u de potentiële impact van om het even welke opdracht begrijpt alvorens het te gebruiken.

[Conventies](#)

Zie de [Cisco Technical Tips Convention](#) voor meer informatie over documentconventies.

[Tijd om tips te leven](#)

Dit zijn enkele tips over hoe TTL werkt op een Cisco Gatekeeper en hoe het verouderingsproces in verschillende gevallen voor de eindpunten werkt.

- De Cisco Gatekeeper verwacht van het eindpunt periodiek te horen via de berichten van het Registratieverzoek (RRQ) (of lichtgewicht of volledig).
- Voor uiteinden van eindpunt, besteedt de Cisco Gatekeeper aandacht aan de TTL waarde die door het eindpunt in het RQ bericht wordt geleverd. Er is een hard codec standaard van 1800 seconden (dertig minuten) voor de eindtijd. Deze waarde kan worden gewijzigd met de Cisco Gatekeeper CLI opdracht [endpointgebeurtenissen](#) `<tijd_waarde>`. Dit wijzigt het gedrag voor alle H.323 v1-eindpunten, of H.323 v2 en latere eindpunten die de TTL-waarde niet in het RQ-bericht opnemen.
- De Cisco Gatekeeper voert periodiek een "verouderingsproces van eindpunt" uit. Dit proces varieert afhankelijk van de huidige CPU-lading van elke minuut tot elke vijf minuten. Voor elke 20 procent van het CPU-gebruik wordt het verouderingsinterval met één minuut verlengd, tot maximaal vijf minuten. Om de CPU's niet te overladen wanneer er veel endpoints zijn, wordt het verouderingsproces alleen uitgevoerd door vijftig eindpunten per doorloop. Als er meer zijn, worden die uitgesteld tot de volgende timer pop. Dit kan van één tot vijf minuten zijn.
- Als het veld TimeToLive in het bericht RQ Registration, Admission and Status (RAS) is opgenomen, gebruikt de poortwachter de waarde van dat veld om de systeemstandaardinstelling te omzeilen of de waarde die is geconfigureerd met behulp van de **opdracht** `tl<time_waarde>` **gatekeeper CLI-opdracht**. Zodra die tijd is verstreken zonder te worden gehoord van het eindpunt, gaat de volgende timer door het schoonmaakproces voor dat eindpunt. Het ergste geval is de TTL die door het eindpunt plus vijf minuten wordt verzonden (als de Cisco Gatekeeper constant onder zware CPU-lading is). Een waarschijnlijker scenario is dat de TTL-tijd plus één minuut wordt verlengd.
- Wanneer het eindpunt niet het veld TimeToLive in het RQ-bericht omvat, behandelt de Cisco Gatekeeper het eindpunt alsof het TTL niet ondersteunt. In dit geval, wanneer de poortwachter geen RQs meer van dat eindpunt ontvangt, doet hij de TTL-tijd (ofwel de standaard van 1800 seconden, of de waarde die in de [opdracht endpointgebeurtenissen](#) is gespecificeerd). Vervolgens wordt er drie informatieaanvragen (IRQ's) verzonden met intervallen van elk één tot vijf minuten (gebaseerd op de CPU-belasting van de poortwachter). Nadat het drie IRQ's heeft verstuurd en geen respons heeft ontvangen, verwijdert de Cisco Gatekeeper eindelijk het eindpunt.
- Als het eindpunt in een actieve vraag is, wordt het niet verouderd tot de vraag wordt beëindigd.
- De Cisco Gatekeeper verwacht te horen van het (de) endpoints die betrokken zijn bij een oproep met het IRR-bericht (Information Response). Als de Cisco Gatekeeper geen periodieke IRR-berichten ontvangt die verwijzingen naar de "hulp" bevatten voor een gesprek,

wacht de poortwachter vier minuten op en stuurt dan een IRQ naar het (de) eindpunt(en) waarvoor de oproep is vereist. Als, na nog acht minuten, de Cisco Gatekeeper nog steeds niets over die oproep heeft gehoord, wordt de oproep schoongemaakt en de poortwachter Vertalingsaanvragen (DRQ's) naar de eindpunten stuurt. Een totaal van ongeveer twaalf minuten vervalft voordat een "gevaarlijke" vraag wordt schoongemaakt (en de bandbreedte wordt bevrijd). Deze aanroep-timer is niet configureerbaar.

- Endpoints die eigendom zijn van een alternatieve Cisco Gatekeeper worden niet direct verouderd (omdat deze gatekeeper het eindpunt niet feitelijk "eigen" heeft).
- Statische eindpunten (gemaakt met de configuratieopdracht ook [bekend als statische xxxxx](#)) zijn niet verouderd.

Cisco Gatekeeper debugs

Dit zijn enkele van de **tonen** en **debug** opdrachten die u kunt gebruiken om te controleren hoe de TTL op verschillende manieren werkt:

- [Toon gatekeeper endpoints](#)
- [Debug ras](#)
- [h225 debug 1](#)

Deze twee secties bespreken twee gevallen waarin Cisco Gatekeeper de eindpunten uitkomt met verschillende TTL-waarden.

Zaak 1

Deze uitvoer is afkomstig van de **opdrachten** [debug ras](#) en [debug h225](#) n1 en is afkomstig van een **Cisco Gatekeeper**. In het debug heeft de gateway een TTL-waarde van 60 seconden. De Cisco Gatekeeper bevestigt en accepteert dit in zijn bericht van de Bevestiging van de Registratie (RCF), ongeacht wat zijn standaard- of geconfigureerde TTL-waarde is. Dit komt doordat het eindpunt een TTL-waarde bevat.

```
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

```

[Zaak 2](#)

Dit is een ander voorbeeld waarbij een eindpunt dat geen TTL-waarde in zijn RQ-bericht heeft verzonden is gecommuniceerd om een lichtgewicht RQ vóór 120 seconden te verzenden, wat de waarde is die op de poortwachter is ingesteld. U kunt in deze uitvoer zien hoe Cisco Gatekeeper het eindpunt niet verwijdert tot drie onbeantwoorde IRQ berichten, alhoewel een Unregistration Application (URQ) bericht werd ontvangen. De tijd tussen de IRQ is tussen één en vijf minuten.

```
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

[Gerelateerde informatie](#)

- [Cisco hoogwaardige Gatekeeper](#)
- [Ondersteuning van H.323, versie 2](#)
- [Problemen oplossen door registratie van Gatekeeper](#)
- [Ondersteuning voor spraaktechnologie](#)
- [Productondersteuning voor spraak en Unified Communications](#)
- [Probleemoplossing voor Cisco IP-telefonie](#)
- [Technische ondersteuning - Cisco-systemen](#)