

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

## Contenu

[Introduction](#)

[Conditions préalables](#)

[Conditions requises](#)

[Components Used](#)

[Conventions](#)

[Conseils sur le temps de vie](#)

[Débogues du contrôleur d'accès Cisco](#)

[Informations connexes](#)

## Introduction

Ce document décrit comment le contrôleur d'accès Cisco vieillit les terminaux en utilisant la valeur TTL (Time to Live) dans différents cas. Les débogages et les commandes **show** sont utilisés pour montrer le fonctionnement du TTL de différentes manières.

## Conditions préalables

### Conditions requises

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

- Mise en oeuvre de Cisco H.323, y compris Cisco Gatekeeper. Pour une compréhension de base des contrôleurs d'accès H.323, référez-vous à [Comprendre les contrôleurs d'accès H.323](#).

### Components Used

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

- Pour les besoins de ce document, le logiciel Cisco IOS® Version 12.3(9) est utilisé pour collecter les informations. Assurez-vous d'utiliser Cisco IOS avec la fonctionnalité de contrôleur d'accès H.323. Ceci est indiqué avec un x dans le nom de l'image Cisco IOS. Par exemple, une plate-forme logicielle Cisco IOS valide pour que le Cisco 3640 agisse en tant que contrôleur d'accès est c3640-ix-mz.123-9.bin.
- Cisco Gatekeeper (toutes les plates-formes). **Remarque** : NetMeeting a été utilisé comme

point de terminaison H.323 dans l'exemple de ce document, car il ne fournit pas de valeur TTL. Pour plus d'informations sur la configuration de NetMeeting avec les passerelles Cisco IOS, référez-vous à [Configuration de Microsoft NetMeeting avec les passerelles Cisco IOS](#).

Les informations présentées dans ce document ont été créées à partir de périphériques dans un environnement de laboratoire spécifique. All of the devices used in this document started with a cleared (default) configuration. 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 sur le temps de vie

Voici quelques conseils sur le fonctionnement de TTL sur un contrôleur d'accès Cisco et sur le fonctionnement du processus de vieillissement pour les terminaux dans différents cas.

- Le contrôleur d'accès Cisco s'attend à recevoir régulièrement des messages du point de terminaison via la requête d'enregistrement (RRQ) (léger ou complet).
- Pour les délais d'attente des points d'extrémité, le contrôleur d'accès Cisco prend en compte la valeur TTL fournie par le point d'extrémité dans le message RRQ. Il existe une valeur par défaut codée en dur de 1 800 secondes (trente minutes) pour le délai d'expiration du point d'extrémité. Cette valeur peut être modifiée avec la commande CLI du contrôleur d'accès Cisco `terminal ttl <time_value>`. Cela modifie le comportement de tous les terminaux H.323 v1 ou H.323 v2 et ultérieurs qui n'incluent pas la valeur TTL dans le message RRQ.
- Le contrôleur d'accès Cisco exécute périodiquement un processus de vieillissement des terminaux. Ce processus varie en fonction de la charge actuelle du processeur de chaque minute à toutes les cinq minutes. Pour chaque vingt pour cent de l'utilisation du processeur, l'intervalle de vieillissement augmente d'une minute, jusqu'à un maximum de cinq minutes. Afin de ne pas surcharger le processeur lorsqu'il y a beaucoup de terminaux, le processus de vieillissement ne passe que par cinquante terminaux par passage. S'il y en a d'autres, ils sont reportés jusqu'à la prochaine fenêtre de temporisation. Cela peut aller de une à cinq minutes.
- Si le champ timeToLive est inclus dans le message RAS (RRQ Registration, Admission and Status), le contrôleur d'accès utilise la valeur de ce champ pour remplacer la valeur par défaut du système ou la valeur configurée à l'aide de la commande CLI du contrôleur d'accès `ttl <time_value> du point de terminaison`. Une fois que cette période s'est écoulée sans entendre le point de terminaison, la fenêtre contextuelle suivante passe par le processus de nettoyage de ce point de terminaison. Le pire est le TTL envoyé par le point de terminaison plus cinq minutes (si le contrôleur d'accès Cisco est constamment soumis à une charge CPU importante). Le scénario le plus défavorable est le délai d'attente de durée de vie plus une minute.
- Lorsque le point de terminaison n'inclut pas le champ timeToLive dans le message RRQ, le contrôleur d'accès Cisco traite le point de terminaison comme s'il ne prenait pas en charge TTL. Dans ce cas, une fois que le contrôleur d'accès n'a plus reçu de RRQ de ce point de terminaison, il exécute le délai de durée de vie (soit la valeur par défaut de 1 800 secondes, soit la valeur spécifiée dans la commande `terminal ttl`). Il envoie ensuite trois demandes d'informations (IRQ) avec des intervalles allant de une à cinq minutes chacune (en fonction de

la charge CPU du contrôleur d'accès). Après avoir envoyé trois IRQ et ne recevoir aucune réponse, le contrôleur d'accès Cisco supprime finalement le point d'extrémité.

- Si le point de terminaison est en cours d'appel, il n'est pas dépassé tant que l'appel n'est pas terminé.
- Le contrôleur d'accès Cisco s'attend à recevoir les informations des terminaux impliqués dans un appel avec le message de réponse aux informations (IRR). Si le contrôleur d'accès Cisco ne reçoit pas de messages IRR périodiques qui contiennent des références au GUID d'un appel, le contrôleur d'accès attend pendant quatre minutes, puis envoie une IRQ au(x) point(s) d'extrémité pour lequel l'appel est effectué. Si, après huit minutes supplémentaires, le contrôleur d'accès Cisco n'a toujours rien entendu à propos de cet appel, l'appel est nettoyé et le contrôleur d'accès envoie des demandes de désengagement (DRQ) aux points d'extrémité. Un total d'environ douze minutes s'écoule avant le nettoyage d'un appel « suspendu » (et la bande passante libérée). Ce compteur d'appels n'est pas configurable.
- Les terminaux qui appartiennent à un autre contrôleur d'accès Cisco ne sont pas mis à expiration directement (car ce contrôleur d'accès ne possède pas le point d'extrémité).
- Les terminaux statiques (créés avec la commande de configuration [alias static xxxxx](#)) ne sont pas obsolètes.

## Débogues du contrôleur d'accès Cisco

Voici quelques-unes des commandes **show** et **debug** que vous pouvez utiliser pour vérifier le fonctionnement de TTL de différentes manières :

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

Ces deux sections traitent de deux cas où Cisco Gatekeeper vieillit les terminaux en utilisant des valeurs de durée de vie différentes.

### Cas 1

Cette sortie provient des commandes [debug ras](#) et [debug h225 asn1](#) et provient d'un contrôleur d'accès Cisco. Dans le débogage, la valeur TTL de la passerelle est de 60 secondes. Le contrôleur d'accès Cisco le confirme et l'accepte dans son message de confirmation d'enregistrement (RCF), quelle que soit la valeur de durée de vie par défaut ou configurée du point d'extrémité. En effet, le point de terminaison inclut une valeur 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
```

## Cas 2

Ceci est un autre exemple où un point de terminaison qui n'a pas envoyé de valeur TTL dans son message RRQ a été averti d'envoyer un RRQ léger avant 120 secondes, soit la valeur configurée sur le contrôleur d'accès. Vous pouvez voir dans ce résultat comment le contrôleur d'accès Cisco ne supprime le point de terminaison que trois messages IRQ sans réponse, même si un message de demande de désinscription (URQ) a été reçu. Les durées entre l'IRQ sont comprises 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	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  RASSignalAddr  Port  Zone Name          Type  Flags
-----
Total number of active registrations = 0
```

## [Informations connexes](#)

- [Contrôleur d'accès hautes performances 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](#)