

Устранение неполадок и общие сведения о TTL сервера-привратника и процессе старения

Содержание

[Введение](#)

[Предварительные условия](#)

[Требования](#)

[Используемые компоненты](#)

[Условные обозначения](#)

[Примечания по сроку жизни](#)

[Отладка привратника Cisco](#)

[Дополнительные сведения](#)

[Введение](#)

Этот документ описывает, как возрасты Сторожевого устройства Cisco оконечные точки с помощью Времени жизни (TTL) оценивают в других случаях. Отладки и команды показа используются, чтобы показать, как TTL работает в различных способах.

[Предварительные условия](#)

[Требования](#)

Читатели этого документа должны знать следующее:

- Реализация Cisco N.323, включая Сторожевое устройство Cisco. Для основного понимания Сторожевых устройств N.323 обратитесь к [Пониманию Сторожевых устройств N.323](#).

[Используемые компоненты](#)

Сведения, содержащиеся в данном документе, касаются следующих версий программного обеспечения и оборудования.

- В целях этого документа релиз 12.3 программного обеспечения Cisco IOS (9) используется для сбора информации. Гарантируйте использование Cisco IOS с функцией функциональности Сторожевого устройства N.323. Это обозначено с х в Имени изображения Cisco IOS. Например, допустимая Cisco IOS для Cisco 3640 для действия как сторожевое устройство является **c3640-ix-mz.123-9.bin**.

- Сторожевое устройство Cisco (все платформы). **Примечание:** NetMeeting использовался в качестве оконечной точки H.323 в примере в этом документе, потому что это не предоставляет значение TTL. Для получения информации о том, как настроить NetMeeting со шлюзами Cisco IOS, обратитесь к [Инструкции, Настраивают Microsoft NetMeeting со шлюзами Cisco IOS](#).

Сведения, содержащиеся в данном документе, были получены с устройств в специальной лабораторной среде. Все устройства, описанные в данном документе, были запущены с конфигурацией по умолчанию. При работе с реальной сетью необходимо полностью осознавать возможные результаты использования всех команд.

[Условные обозначения](#)

[Дополнительные сведения об условных обозначениях см. в документе Технические рекомендации Cisco. Условные обозначения.](#)

[Примечания по сроку жизни](#)

Это некоторые советы относительно того, как TTL работает на Сторожевое устройство Cisco и как процесс старения работает для оконечных точек в других случаях.

- Сторожевое устройство Cisco ожидает получать известие от оконечной точки периодически с помощью сообщений Запроса регистрации (RRQ) (или легкий вес или полный).
- Для таймаутов оконечной точки Сторожевое устройство Cisco обращает внимание на значение TTL, предоставленное оконечной точкой в сообщении RRQ. Существует твердый закодированный по умолчанию 1800 секунд (тридцать минут) для таймаута оконечной точки. Это значение может быть изменено с [TTL-схемой оконечной точки команды в командной строке](#) Сторожевого устройства Cisco `<time_value>`. Это изменяет поведение для всех оконечных точек v1 H.323, или H.323 v2 и более поздние оконечные точки, которые не включают значение TTL в сообщении RRQ.
- Сторожевое устройство Cisco выполняет "процесс старения оконечной точки" периодически. Этот процесс варьируется на основе текущей Загрузки ЦПУ с каждой минуты до каждых пяти минут. Для каждых двадцати процентов загрузки ЦПУ стареющий интервал увеличивается на одну минуту максимум до пяти минут. Для не перегрузки ЦП, когда существует много оконечных точек, процесс старения только пробегает пятьдесят оконечных точек на проход. Если существует больше, то те отсрочены до следующего населения таймера Это может быть с одной до пяти минут.
- Если поле `timetolive` включено в Регистрацию RRQ, Разрешение и Статус (RAS) сообщение, сторожевое устройство использует значение того поля для переопределения системного параметра по умолчанию, или значение настроило использование [endpoint ttl](#) `<time_value>` команда CLI сторожевого устройства. Как только тот период времени истекает, не получая известие от оконечной точки, следующая популярность таймера проходит процесс очистки для той оконечной точки. Наихудший случай является TTL, передаваемым оконечной точкой плюс пять минут (если Сторожевое устройство Cisco последовательно находится под тяжелой Загрузкой ЦПУ). Более вероятный наихудший случай является таймаутом TTL плюс одна минута.
- Когда оконечная точка не включает поле `timetolive` в сообщении RRQ, Сторожевое устройство Cisco рассматривает оконечную точку, как будто это не поддерживает TTL. В

этом случае, как только сторожевое устройство больше не получает RRQ от той конечной точки, оно делает таймаут TTL (или по умолчанию 1800 секунд или значение, заданное в [команде endpoint ttl](#)). Затем это отправляет три Информационных запроса (IRQ) с интервалами где угодно от одной до пяти минут каждый (на основе Загрузки ЦПУ сторожевого устройства). После того, как это передаст три IRQ и не получит ответа, Сторожевое устройство Cisco наконец удаляет конечную точку.

- Если конечная точка находится в активном вызове, это не в возрасте, пока не завершён вызов.
- Сторожевое устройство Cisco ожидает получить известие от конечной точки (точек), вовлеченной в вызов с Information Response (IRR) сообщение. Если Сторожевое устройство Cisco не получает периодические сообщения IRR, которые содержат ссылки на "гид" для вызова, сторожевое устройство ждет в течение четырех минут, и затем отправляет IRQ в конечную точку (точки), для которой вызов. Если после еще восьми минут Сторожевое устройство Cisco все еще ничего не услышало о том вызове, вызов очищен, и сторожевое устройство передает Запросы на освобождение канала (DRQ) к конечным точкам. В общей сложности приблизительно двенадцать минут истекают, прежде чем "повисший" вызов очищен (и освобожденная пропускная способность). Этот таймер вызова не конфигурируем.
- Конечные точки, которые принадлежат альтернативному Сторожевому устройству Cisco, не в возрасте непосредственно (так как это сторожевое устройство фактически "не владеет" конечной точкой).
- Статические конечные точки (созданный с [alias static](#) команды настройки `xxxxx`) не в возрасте.

[Отладка привратника Cisco](#)

Это часть **показа** и **команд отладки**, которые можно использовать, чтобы проверить, как TTL работает в различных способах:

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

Эти два раздела обсуждают два случая, где возрасты Сторожевого устройства Cisco конечные точки с помощью другого TTL оценивают.

[Пример 1](#)

Эти выходные данные от [команд debug ras и debug h225 asn1](#) и взяты от Сторожевого устройства Cisco. В отладке шлюз имеет значение TTL 60 секунд. Сторожевое устройство Cisco подтверждает и принимает это в его сообщении Подтверждения регистрации (RCF), независимо от того каково его значение TTL или настраиваемой конечной точки по умолчанию. Это вызвано тем, что конечная точка включает значение 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
```

Случай 2

Это - другой пример, где оконечная точка, которая не передала значение TTL в его сообщении RRQ, была уведомлена для передачи легковесного RRQ перед 120 секундами, который является значением, настроенным на сторожевом устройстве. Вы видите в этих выходных данных, как Сторожевое устройство Cisco не удаляет оконечную точку до трех оставшихся без ответа сообщений IRQ, даже при том, что был получен Запрос на отмену регистрации (URQ) сообщение. Времена между IRQ между одной и пятью минутами.

```
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 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

```

Дополнительные сведения

- [Высокопроизводительный привратник Cisco](#)
- [Поддержка H.323 версии 2](#)
- [Устранение ошибок при регистрации привратника](#)
- [Поддержка голосовых технологий](#)
- [Поддержка продуктов Голосовой и Унифицированной связи](#)
- [Устранение неполадок в системах IP-телефонии Cisco](#)
- [Техническая поддержка - Cisco Systems](#)