

ドットによるゾーンプレフィクス処理とアスタリスクによるゾーンプレフィクス処理

目次

[概要](#)

[前提条件](#)

[要件](#)

[使用するコンポーネント](#)

[表記法](#)

[問題](#)

[解決策](#)

[ゲートキーパーのデフォルトでのゾーンプレフィクス処理の動作解説](#)

[ケーススタディ](#)

[設定と show コマンド](#)

[デバッグとその詳細](#)

[関連情報](#)

概要

このドキュメントでは、ネットワーク開発者がゾーンプレフィクスでワイルドカードとしてドットを使用する場合に起こる障害について解説します。また、この問題の一般的な解決法として、ワイルドカードにアスタリスク（「*」）を使用する方法を紹介します。最後に、この文書ではゾーン処理の論理を2種類のワイルドカード設定方式の違いに特定して明確にします。

前提条件

要件

このドキュメントの読者には、H.323 のフローと Cisco ゲートキーパーの概念、特にゾーン処理についての知識が必要です。Cisco ゲートキーパーとゾーン処理についての詳細は、『[Cisco IOS ゲートキーパーのコールルーティングについて](#)』および『[H.323 ゲートキーパーとプロキシの設定](#)』を参照してください。前者は、ゲートキーパーのゾーン処理について理解するのに役立ちます。

使用するコンポーネント

このドキュメントは、特定のソフトウェアやハードウェアのバージョンに限定されるものではありません。

表記法

ドキュメント表記の詳細は、『[シスコテクニカルティップスの表記法](#)』を参照してください。

問題

ドットとアスタリスクを使用する際に生じる混乱の原因は、プレフィクスを処理する際のゲートキーパーのデフォルトの動作にあります。この動作については、このドキュメントの「[ゲートキーパーのデフォルトでのゾーンプレフィクス処理の動作解説](#)」セクションで詳しく説明していますが、ダイヤルプランに重複する部分があり、設定でドットとアスタリスクの両方を使用している場合には、このためにあいまいな状況が生じることがあります。

次に、この問題の症状と特質を説明します。

- ローカルゲートキーパーはコールを複数のローカルゾーン、リモートゾーンにあるゲートキーパー、およびその両方にルーティングするように期待されています。
- ローカルゾーン内のコールは正常にルーティングされます。
- しかし、ゾーン間のコールは正常にルーティングされない場合があります。
- 正常にルートされないインターゾーンのコールは、特定の桁を持つ着番号に対するコールです。たとえば、10桁または9桁の番号へのコールは成功しますが、同じ数値で始まる3桁の番号へのコールは必ず失敗します。
- ゲートキーパーの設定では、ゾーンプレフィックスの中にドットワイルドカードを使用します。

解決策

ゾーンプレフィックスにワイルドカードの桁を指定する場合、可能であればドットの使用は避けてください。代わりに、限定度の低いアスタリスクのワイルドカードを使用します。また、次の規則に従うことでも、問題を回避できます。

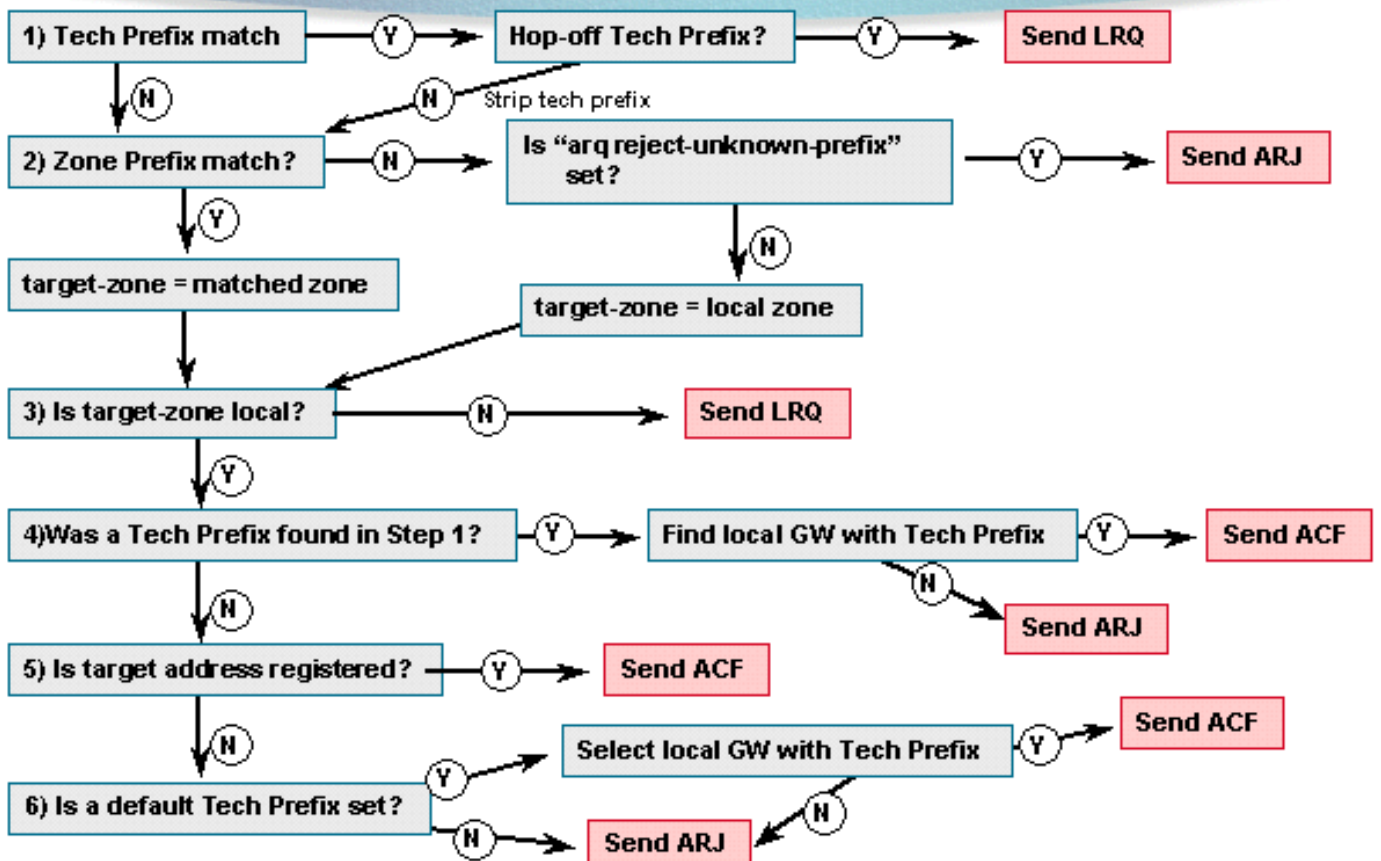
1. ダイヤルプランが一致する場合は、ドットのみ（またはアスタリスクのみ）を使用した設定が可能です。
2. ダイヤルプランに重複部分がある場合は、アスタリスクのみの設定が適しています。
3. ダイヤルプランに重複する部分があり、アスタリスクだけによる設定が無理な場合は、ゲートキーパーを設定する前に、ゲートキーパーがプレフィクスを推定する（ローカルエリアコードを推測して先頭に付け加える）際のデフォルト動作についてよく理解してください。

この3つ目の規則では、このドキュメントで説明するゲートキーパーの動作について詳細に理解することが求められています。

[ゲートキーパーのデフォルトでのゾーンプレフィクス処理の動作解説](#)

この例では、H.323 エンドポイントからの Admission Request (ARQ; 許可要求) の形式によるコール要求をゲートキーパーが処理する際の動作について説明しています。ステップ2と3がこのドキュメントが対象とする重要なポイントです。デバッグ例への参照の資料のこのフローチャート以降によって進むことができます: [失敗したコールを参照して、このフローチャートを進むことができます。](#)

GK Address Resolution on ARQ



ゾーンプレフィクス処理は宛先プレフィクス処理とは少し異なります。ゾーンプレフィクスを照合する際、Cisco ゲートキーパーは可能な限りエリアコードでゾーンを限定するための特別な試みを行います。着番号がローカルゾーンに該当した場合、ゲートキーパーでは着番号の先頭にそのローカルエリアコード（発番号のプレフィクス）を付け加える必要があるものと見なします。

たとえば、ARQ がゲートキーパーに送信されたと仮定します。発番号は「415xxxxxxx」（エリアコード 415）です。

ゲートキーパーはサポートプレフィクス "415....."（7つのドット）で設定される 415 ゾーンを備えています。このエントリによって、着番号が 5551212（7桁）であるとする、ゲートキーパーは発番号と同じプレフィクスをこの番号の先頭に付け加えます。その結果、ローカルゾーン内で処理される着番号は 4155551212 となります。

注: zone prefix コマンド内のドットの数により、着番号がローカルゾーンに該当するかどうかが決まります。上記について 6 桁数（たとえば: 555123）"415....."（7つのドット）の構成済みのゾーンプレフィクスと一致する。従って、呼出し番号は 415555123 であるために推論されないし、555123 に残り、"555*" のゾーンプレフィックスを一致します。

しかし、ローカルゾーンが「415*」に設定されていて、さらに設定に「*」対応するデフォルトゾーン X が含まれている場合、アドレス 5551212 の解決が依頼されると、ゲートウェイでは、X がローカルゲートキーパー上の別のゾーンである場合には ARQ がゾーン X に該当するとして処理されます。またリモートゾーンである場合は、Location Request (LRQ; ロケーション要求) が X に送信されます。

これは一致する Cisco IOS[®] コンフィギュレーションの断片への参照の概念を説明する例です。

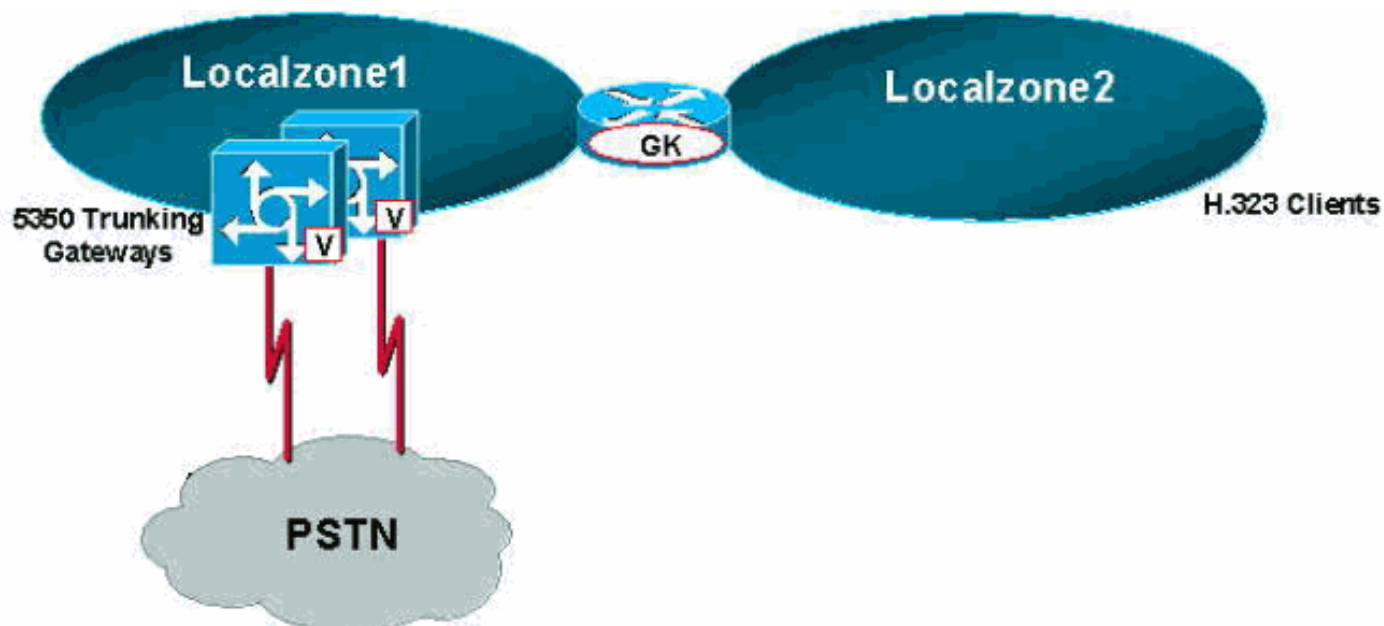
ドットとアスタリスクによるゾーンプレフィクス動作 - ゲートキーパの設定の断片

!--- 5551212 is the called number !--- and the request comes into zone localzone2. !--- It is important to know that the calling number has prefix 415. zone prefix localzone2 415..... zone prefix localzone1 555 !--- In this case, this line is what the match is with. Zone prefix localzone2 415..... !--- The match is due to these reasons: !--- 1. The calling number begins with 415. !--- 2. There is a local wildcard entry for 415 with seven dots. This entry !--- causes the gatekeeper to assume that the the seven-digit called !--- number is local and therefore expands 5551212 to 4155551212 by !--- prepending the area code of the calling number. This expanded !--- number matches, and the call will be accepted or rejected based on !--- the registered resources, in localzone2. !--- If the configuration is changed, as shown here, then there is no !--- expansion of the number (because there is no seven-dot entry). zone prefix localzone2 415* zone prefix localzone1 555* !--- This line is what the match is now with. Zone prefix localzone1 555* !--- In this case, the call is accepted or rejected based on registered !--- resources in localzone1.*

ケーススタディ

注: この事例では、2つのローカルゾーンが設定されたゲートキーパーを1つ使用します。ローカルゲートキーパーがLQRをリモートゾーンのゲートキーパーに転送するのと同じ方式を、複数のゲートキーパーの設計に適用します。

次の図は、サービスプロバイダーネットワーク「new world」の簡素化したH.323ゾーンの様子を表しています。このネットワークはVoice over IP (VoIP; インターネットプロトコル上の音声) コールをlocalzone2というゾーンにあるH.323クライアント間で提供し、同一クライアントからPublic Switched Telephone Network(PSTN; 公衆交換電話網)にアクセスします。Trunking Gateways (TGW; トランキングゲートウェイ)はlocalzone1と呼ばれる別のゾーンに常駐するPSTNへのアクセスを提供します。



注: H.323 クライアントは、ネイティブの H.323 IP テレフォニー ユーザ (Cisco ATA や同様のサードパーティ製品などのシンプルなアナログ ツー H.323 アダプタ デバイス) か、大規模なゲートウェイのいずれかです。大規模なゲートウェイ設計、特にリモートテレフォニー ユーザを持つような設計をサポートするには、この事例で説明するものよりも遙かに複雑なゾーン構造が必要になります。さらに、5350 の TGWs は Primary Rate ISDN またはチャネル連携信号 (CAS) のようなデジタル E1/T1 接続を通して PSTN アクセスを提供できます。また、Cisco SC2000 や PGW2200 などの適切な SS7 コール エージェントを使用した、直接での SS7 相互接続にも対応しています。

設定と show コマンド

ゲートキーパーに関連するコマンドで、ゲートキーパーにインストールされているものを次に示します。設定の行のうち、強調表示されているものは、後ほど 3 桁の電話番号による問題の説明の際に重要になります。この説明では、localzone2 から localzone1 に対してコールを発信します。

```

ゲートキーパーの設定 ( ゲートキーパーのコマンドのみ )
gatekeeper
  zone local localzone1 dns.au 10.1.1.228
  zone local localzone2 dns.au
  no zone subnet localzone1 default enable
  zone subnet localzone1 10.1.1.240/28 enable
  no zone subnet localzone2 default enable
  zone subnet localzone2 10.99.0.0/16 enable
  zone prefix localzone1 0*
zone prefix localzone1 1* zone prefix localzone1 6* zone
prefix localzone1 8* zone prefix localzone2 9999931..
Zone prefix localzone2 9999932.. Zone prefix localzone2
9999933.. Zone prefix localzone2 9999934.. Zone prefix
localzone2 9999935.. Zone prefix localzone2 9999936..
Zone prefix localzone2 9999937.. Zone prefix localzone2
9999938.. Zone prefix localzone2 9999939.. Zone prefix
localzone2 999994... zone prefix localzone2 999995...
zone prefix localzone1 9* accounting vsa gw-type-prefix
1#* default-technology arq reject-unknown-prefix lrq
reject-unknown-prefix no use-proxy localzone2 default
inbound-to terminal no use-proxy localzone2 default
outbound-from terminal no shutdown endpoint ttl 60

```

この `show gatekeeper endpoints` コマンドの出力では、ゲートキーパーに登録されている H.323 エンドポイントと、これらが登録されているゾーンが表示されています。

注: TGW は localzone1 でゲートキーパーに正しく登録されています。また、H.323 ターミナルは localzone2 で登録されています。

```

show gatekeeper endpoints
GK#show gatekeeper endpoints GATEKEEPER ENDPOINT
REGISTRATION =====
CallSignalAddr Port RASSignalAddr Port Zone Name Type
Flags -----
--- ---- - 10.99.0.10 1720 10.99.0.10 45690
localzone2 TERM E164-ID: 999995988 10.99.0.11 1720
10.99.0.11 29249 localzone2 TERM E164-ID: 999995981
10.99.0.12 1720 10.99.0.12 19227 localzone2 TERM E164-
ID: 999995985 10.99.0.15 1720 10.99.0.15 36889
localzone2 TERM E164-ID: 999995989 10.99.0.16 1720

```

```
10.99.0.16 42366 localzone2 TERM E164-ID: 999995982
10.99.0.18 1720 10.99.0.18 18300 localzone2 TERM E164-
ID: 999995986 10.99.0.19 1720 10.99.0.19 32345
localzone2 TERM E164-ID: 999995980 10.99.0.20 1720
10.99.0.20 23155 localzone2 TERM E164-ID: 999995984
10.1.1.240 1720 10.1.1.240 50737 localzone1 VOIP-GW
H323-ID: tgw1@dns.au 10.1.1.241 1720 10.1.1.241 50737
localzone1 VOIP-GW H323-ID: tgw2@dna.au Total number of
active registrations = 10
```

この `show gatekeeper zone prefix` コマンドの出力には、それぞれの E.164 プレフィクスがルーティングされるゾーンが正しく示されています。

ゲートキーパのゾーンプレフィクスの表示

```
ZRZ-GK1#show gatekeeper zone prefix ZONE PREFIX TABLE
===== GK-NAME E164-PREFIX -----
- localzone1 0* localzone1 1* localzone1 6* localzone1
8* localzone2 9999931.. localzone2 9999932.. localzone2
9999933.. localzone2 9999934.. localzone2 9999935..
localzone2 9999936.. localzone2 9999937.. localzone2
9999938.. localzone2 9999939.. localzone2 999994...
localzone2 999995... localzone1 9*
```

この `show gatekeeper gw-type-prefix` コマンドの出力には、このゲートキーパーに設定されているテクノロジープレフィクスが示されています。

ゲートキーパーには、デフォルトのテクノロジープレフィクス (1#) だけが設定されていることに注意してください。また、localzone1 ゾーンにある 5350 TGW (tg1 および tgw2) だけが、このデフォルトのテクノロジープレフィクスで登録されるように設定されています。

show gatekeeper gw-type-prefix

```
GK#show gatekeeper gw-type-prefix GATEWAY TYPE PREFIX
TABLE ===== Prefix: 1#* (Default
gateway-technology) Zone localzone1 master gateway list:
10.1.1.240:1720 tgw1 10.1.1.241:1720 tgw2 (out-of-
resources)
```

デバッグとその詳細

このデバッグ出力はゲートキーパーから得られたもので、次のコールについての Registration, Admission, and Status Protocol (RAS) のフローおよびゾーンプレフィクス処理を示しています

。

- [失敗したコール](#)
- [成功したコール](#)

このデバッグの出力には、ゾーンプレフィクスでドットワイルドカードをアスタリスクのワイルドカードと比較して処理した際の、ゲートキーパーの動作についての詳細な解説が含まれます

。

debug h225 asn1 と debug gatekeeper main 10 - 失敗したコール

```
GK#show debug gk main debug level = 10 H.225: H.225 ASN1
Messages debugging is on !--- This output is from the
debug h225 asn1 command issued on the gatekeeper. It
shows !--- an incoming RAS ARQ for called number 112. It
```


is important to !--- note that the calling number (source endpoint) comes from the zone localzone2 and, !--- assuming three-digit numbers, its prefix (source endpoint prefix) is 999995. Mar 11 21:48:15: **RAS INCOMING PDU** ::= value RasMessage ::= **admissionRequest** : { requestSeqNum 36784 callType pointToPoint : NULL callModel gatekeeperRouted : NULL endpointIdentifier {"618FED9800000008"} destinationInfo { e164 : "**112**", e164 : "**112**" } srcInfo { h323-ID : {"999995985"}, e164 : "**999995985**" } srcCallSignalAddress ipAddress : { ip '0A14000C'H port 11309 } bandWidth 1280 callReferenceValue 31633 conferenceID '5634343434EF21002B211E5226E91D26'H activeMC FALSE answerCall FALSE canMapAlias FALSE callIdentifier { guid '5634343434EF20002B211E5226E91D26'H } gatekeeperIdentifier {"**localzone2**"} willSupplyUIEs FALSE } !--- This output is from the **debug gatekeeper main 10** command !--- issued on the gatekeeper. It !--- shows the gatekeeper zone prefix processing logic (rassrv_get_addrinfo). !--- Comments are inserted throughout. Mar 11 21:48:15: gk_rassrv_arq: arqp=0x61A09EE4, crv=0x7B91, answerCall=0 Mar 11 21:48:15: ARQ Didn't use GK_AAA_PROC !--- Tech-prefix matching occurs first. In this case study, no !--- tech-prefixes are configured so no match is found. Mar 11 21:48:15: rassrv_get_addrinfo(**112**): Tech-prefix match failed. !--- The next line in the trace is the key to what, in this case study, is unexpected !--- behavior. The expected behavior is for 112 to match with the wildcard "1*" entry !--- in localzone1. !--- The local (source) zone of the calling number is localzone2. !--- It has been configured as !--- supporting the prefix "999995..." with three wildcard digits. !--- (Note the configuration line !--- "zone prefix localzone2 999995...".) !--- The gatekeeper, when asked to resolve a three-digit number 112, !--- deduces this to mean "999995-112" in the local zone because !--- "112" matches with the specific-length three-dot !--- wildcard configuration for the local zone. !--- This behavior is exactly the same as a local area code being assumed when a local !--- call is made. !--- If the configuration line "zone prefix localzone2 999995..." was removed from the !--- configuration, or if the line "zone prefix localzone2 999995*" was inserted instead, !--- then the three-digit number "112" would not match in the local !--- zone but would rather match localzone1 through the !--- configuration line "zone prefix localzone1 1*". Mar 11 21:48:15: rassrv_get_addrinfo(**112**): **Defaulting to source endpoint's zone prefix 999995** Mar 11 21:48:15: No tech-prefix Mar 11 21:48:15: Alias not found !--- The gatekeeper attempts to find a default technology prefix, But although "#1" is !--- configured, the H.323 endpoints in localzone2 correctly do not register with that. The !--- conclusion drawn is that there is an "unknown address and no default !--- technology defined": Mar 11 21:48:15: rassrv_get_addrinfo(112): **default-tech gateway selection failed**, status = 0x805 Mar 11 21:48:15: rassrv_get_addrinfo(112): **unknown address and no default technology defined**. Mar 11 21:48:15: rassrv_get_addrinfo(112): Tech-prefix match failed. Mar 11 21:48:15: rassrv_get_addrinfo(112): Defaulting to source endpoint's zone prefix 999995 Mar 11 21:48:15: No tech prefix Mar 11 21:48:15: Alias not found !--- The gatekeeper indicates that it has failed

```
to find a registered match for the !--- called number in
localzone2: Mar 11 21:48:15: rassrv_get_addrinfo(112):
default-tech gateway selection failed, status = 0x805
Mar 11 21:48:15: rassrv_get_addrinfo(112): unknown
address and no default technology defined. Mar 11
21:48:15: gk_rassrv_sep_arq(): rassrv_get_addrinfo()
failed (return code = 0x103) !--- The gatekeeper sends
the Admission Reject (ARJ) because the called party is
not !--- registered: Mar 11 21:48:15: RAS OUTGOING PDU
::= value RasMessage ::= admissionReject : {
requestSeqNum 36784 rejectReason
calledPartyNotRegistered : NULL }
```

このデバッグは `debug gatekeeper main 10` コマンドの出力の一部であり、成功したコールを示しています。

debug gatekeeper main 10 –成功コール

```
GK#show debug gk main debug level = 10 H.225: H.225 ASN1
Messages debugging is on !--- The four-digit called
number 1003 does not match with the three-dot wildcard
!--- for localzone2 noted earlier. Instead, it matches
with the less-specific !--- asterisk wildcard for
localzone1. Feb 19 16:52:19: rassrv_get_addrinfo(1003):
Tech-prefix match failed. Feb 19 16:52:19:
rassrv_get_addrinfo(1003): Matched zone prefix 1 and
remainder 003 Feb 19 16:52:19: No tech prefix Feb 19
16:52:19: Alias not found !--- The gatekeeper finds a
default technology prefix (of #1) since the 5350 !---
TGWs register with this prefix as per the show
gatekeeper gw-type-prefix command. Feb 19 16:52:19:
Technology GW selected
```

関連情報

- [H.323 ゲートキーパーとプロキシの設定](#)
- [H.323 ゲートキーパーについて](#)
- [ゲートキーパーを使用した VoIP](#)
- [Cisco IOS ゲートキーパーのコール ルーティングについて](#)
- [音声に関する技術サポート](#)
- [音声とユニファイド コミュニケーションに関する製品サポート](#)
- [Cisco IP Telephony のトラブルシューティング](#)
- [テクニカルサポートとドキュメント - Cisco Systems](#)