

Zone Prefix Processing with Dots Versus Asterisks

Document ID: 23900

Introduction

Prerequisites

- Requirements

- Components Used

- Conventions

Problem

Solution

- Explanation of Gatekeeper Default Zone Prefix Processing Behavior

Case Study

- configuration and show Commands

- Debugs and Detailed Discussion

Related Information

Introduction

This document discusses a problem some network implementers have experienced with the use of dots as wildcards within zone prefixes. It then presents a general solution to this problem by proposing the use, where possible, of asterisk ("*") wildcards instead. Finally, this document clarifies zone processing logic with a specific reference to the difference between the two methods of configuring wildcards.

Prerequisites

Requirements

Readers of this document should be knowledgeable of H.323 flows and Cisco gatekeeper concepts, in particular zone processing. Refer to Understanding Cisco IOS Gatekeeper Call Routing and Configuring H.323 Gatekeepers and Proxies for more information on Cisco gatekeeper and zone processing. The first of these documents is useful for understanding gatekeeper zone processing.

Components Used

This document is not restricted to specific software and hardware versions.

Conventions

Refer to Cisco Technical Tips Conventions for more information on document conventions.

Problem

The root cause of confusion related to the use of dots and asterisks lies in the default behavior of the gatekeeper while processing prefixes. This behavior, described in detail in the Explanation of Gatekeeper Default Zone Prefix Processing Behavior section of this document, can create ambiguous situations if there is an overlap in the dial-plan and the configuration makes use of both dots and asterisks.

The symptoms and characteristics of the problem are:

- The local gatekeeper is expected to route calls to more than one local zone or is expected to route calls to gatekeepers in remote zones or both.
- Calls within a local zone can be routed successfully.
- Some, but not all, interzone calls can be routed successfully.
- The interzone calls that are not routed successfully are to called numbers with a specific number of digits. For example, calls to a 10–digit or nine–digit number may succeed, while a call to a three–digit number starting with the same digit reliably fails.
- The gatekeeper configuration makes use of dot wildcards within zone prefixes.

Solution

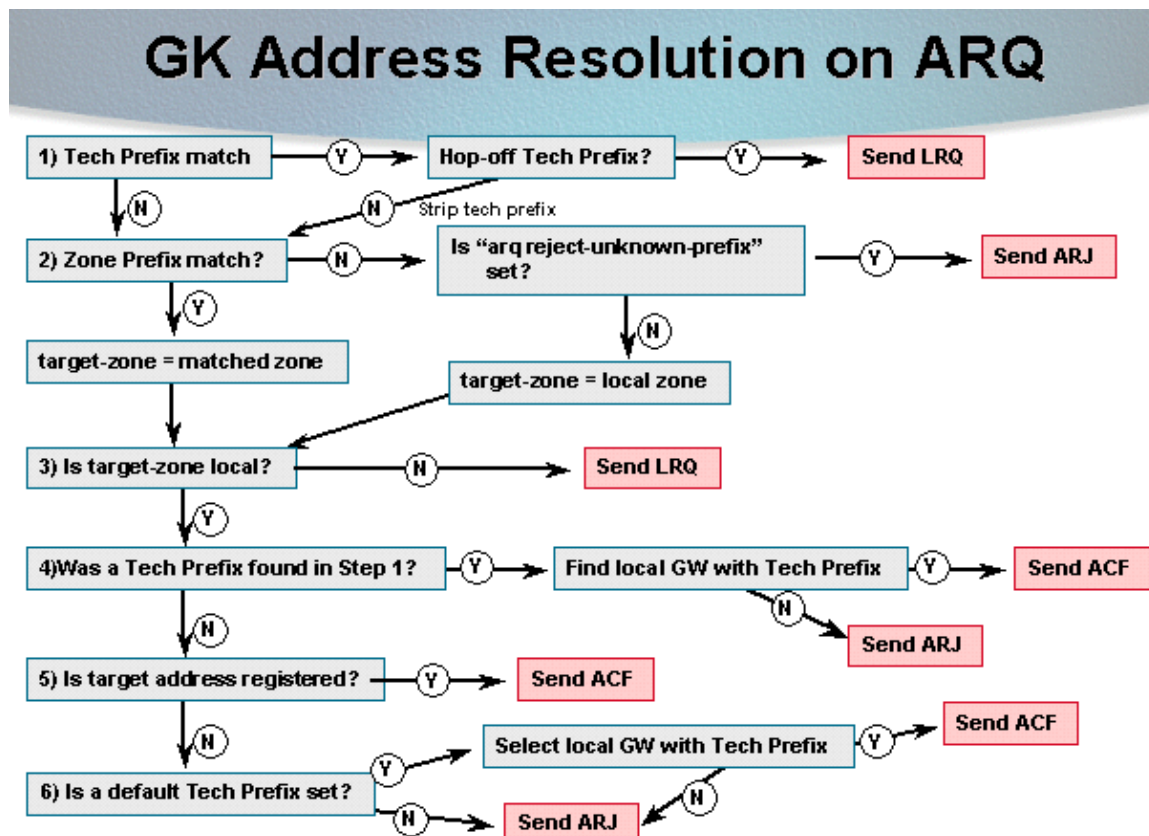
When you specify wildcard digits within a zone prefix, avoid using dots where possible. Instead, use the less–specific asterisk wildcard. You can also avoid the problem when you observe these rules:

1. If the dial–plan is consistent, you can use a configuration with only dots (or using only asterisks).
2. If there is an overlap in the dial–plan, it is best to stick to using configurations with asterisks.
3. If there is overlap in the dial–plan, and a configuration with only asterisks is not suitable, study the gatekeeper's default behavior of prefix guessing (deduce and prepend the local area code) before you configure the gatekeeper.

The third rule requires an understanding of the details of the gatekeeper's behavior as described in this document.

Explanation of Gatekeeper Default Zone Prefix Processing Behavior

This example illustrates the behavior of a gatekeeper when it processes a call request in the form of an Admission Request (ARQ) from an H.323 endpoint. Steps 2 and 3 are key for the scope of this document. You can step through this flow chart later in the document with a reference to the example debug: A failed call.



Zone prefix processing is slightly different than destination prefix processing. When you match zone prefixes, the Cisco gatekeeper makes a special attempt to qualify the zone by area code if possible. If a called number is matched in the local zone, the gatekeeper deduces that the local area code (the prefix of the calling number) should be prepended to the called number.

For example, an ARQ comes into a gatekeeper with calling number "415xxxxxxx" (with area code 415).

The gatekeeper has the 415 zone configured as supporting prefix "415....." (seven dots). Because of this entry, if the called number is 5551212 (specifically seven digits) then the gatekeeper prepends it with same prefix as the calling number. Therefore, the called number to be processed is 4155551212, in the local zone.

Note: The number of dots in a **zone prefix** command determines if a called number matches in the local zone. In the above, a six-digit number (for example: 555123) does not match with the configured zone prefix of "415....." (seven dots). Therefore, the called number is not deduced to be 415555123, but remains 555123 and matches the zone prefix of "555*".

However, if the local zone is configured as "415*", and the configuration also includes a default zone X which handles "*", then when asked to resolve the address 5551212, the gateway processes the ARQ as matching in zone X, if X is another zone on the local gatekeeper, or sends a Location Request (LRQ) to X if it is a remote zone.

This is an example that illustrates the concept with a reference to matching Cisco IOS® configuration snippets.

Zone Prefix Behavior with Dots Versus Asterisks Gatekeeper Configuration Snippets

```
!--- 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*

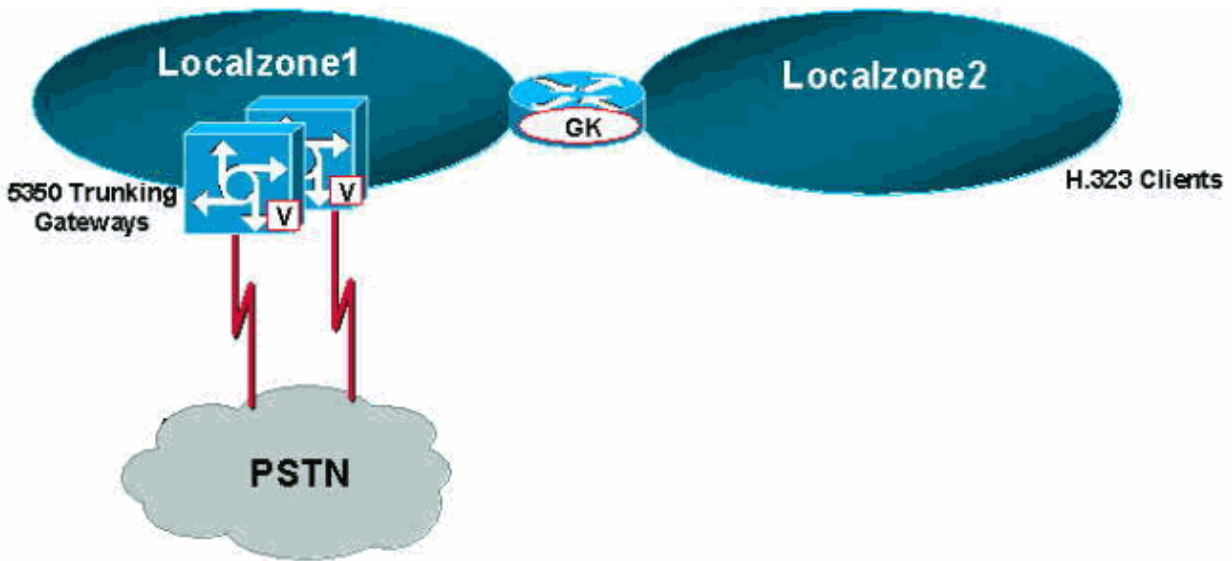
!--- In this case, the call is accepted or rejected based on registered
```

```
!--- resources in localzone1.
```

Case Study

Note: This case study makes use of a single gatekeeper with two local zones. The same principles apply to multiple gatekeeper designs where the local gatekeeper is configured to forward LRQs to remote zone gatekeepers.

This diagram shows a simplified H.323 zone view of a "new world" service provider network. This network provides Voice over IP (VoIP) calls between H.323 clients in the zone called localzone2, and also access to the Public Switched Telephone Network (PSTN) from those same clients. The Trunking Gateways (TGWs) that provide access to the PSTN reside in a separate zone called localzone1.



Note: The H.323 clients can either be native H.323 IP Telephony users, simple analogue-to-H.323 adapter devices, such as the Cisco ATA or other similar third party products, or larger scale gateways. Support for larger scale gateway designs, particularly those with remote Telephony users, likely would entail a more complex zone structure than what is discussed in this case study. In addition, the 5350 TGWs can provide PSTN access through digital E1/T1 connections such as primary rate ISDN or Channel Associated Signaling (CAS). They also can provide direct SS7 interconnection with the use of a suitable SS7 call agent, such as the Cisco SC2000 or PGW2200.

configuration and show Commands

The gatekeeper-related commands installed on the gatekeeper are shown here. The lines in the configuration that are highlighted are significant in later demonstrating the problem with, in this case, three-digit telephone numbers where a call is attempted from localzone2 to localzone1.

Gatekeeper Configuration (gatekeeper commands only)

```
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

```

This **show gatekeeper endpoints** command output shows the H.323 endpoints registered with the gatekeeper along with the zones in which they are registered.

Note: The TGWs have registered correctly to the gatekeeper in localzone1 while the H.323 terminals are registered in localzone2.

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						

This **show gatekeeper zone prefix** command output correctly indicates the zone to which respective E.164 prefixes are to be routed.

show gatekeeper zone prefix	
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*

```

This **show gatekeeper gw-type-prefix** command output shows the tech prefixes configured for this gatekeeper.

Notice that only the default tech-prefix (1#) is configured on the gatekeeper. Additionally, only the 5350 TGWs (tgw1 and tgw2) in zone localzone1 are configured to register with this default technology prefix.

```

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)

```

Debugs and Detailed Discussion

This is debug output from the gatekeeper that shows Registration, Admission, and Status Protocol (RAS) flows and zone prefix processing for:

- A failed call
- A successful call

It includes a detailed commentary that explains the behavior of the gatekeeper when processing dot wildcards in zone prefixes as compared to asterisk wildcards.

```

debug h225 asn1 and debug gatekeeper main 10 failed call
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"}
  willSupplyUUIEs 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
    }

```

This debug is an extract from the output of the **debug gatekeeper main 10** command and shows a successful call.

debug gatekeeper main 10 successful call
<pre> 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 </pre>

Related Information

- **Configuring H.323 Gatekeepers and Proxies**
 - **Understanding H.323 Gatekeepers**
 - **VoIP with Gatekeeper**
 - **Understanding Cisco IOS Gatekeeper Call Routing**
 - **Voice Technology Support**
 - **Voice and Unified Communications Product Support**
 - **Recommended Reading: Troubleshooting Cisco IP Telephony**
 - **Technical Support & Documentation – Cisco Systems**
-

[Contacts & Feedback](#) | [Help](#) | [Site Map](#)

© 2009 – 2010 Cisco Systems, Inc. All rights reserved. [Terms & Conditions](#) | [Privacy Statement](#) | [Cookie Policy](#) | [Trademarks of Cisco Systems, Inc.](#)

Updated: May 22, 2006

Document ID: 23900
