EEM Applets Used to Detect and Clear PfR Forwarding Loops

Document ID: 116206

Contributed by Fabrice Ducomble and Atri Basu, Cisco TAC Engineers.
Sep 25, 2013

Contents

Introduction

This document describes Embedded Event Manager (EEM) applets that are used in networks where Performance Routing (PfR) optimizes traffic through multiple Border Relays (BRs). Some forwarding loops are also observed. The applets are used in order to collect data when a loop is observed and mitigate the impact of a forwarding loop.

Prerequisites

Requirements

There are no specific requirements for this document.

Components Used

The information in this document is based on Cisco IOS® software that supports EEM Version 4.0.

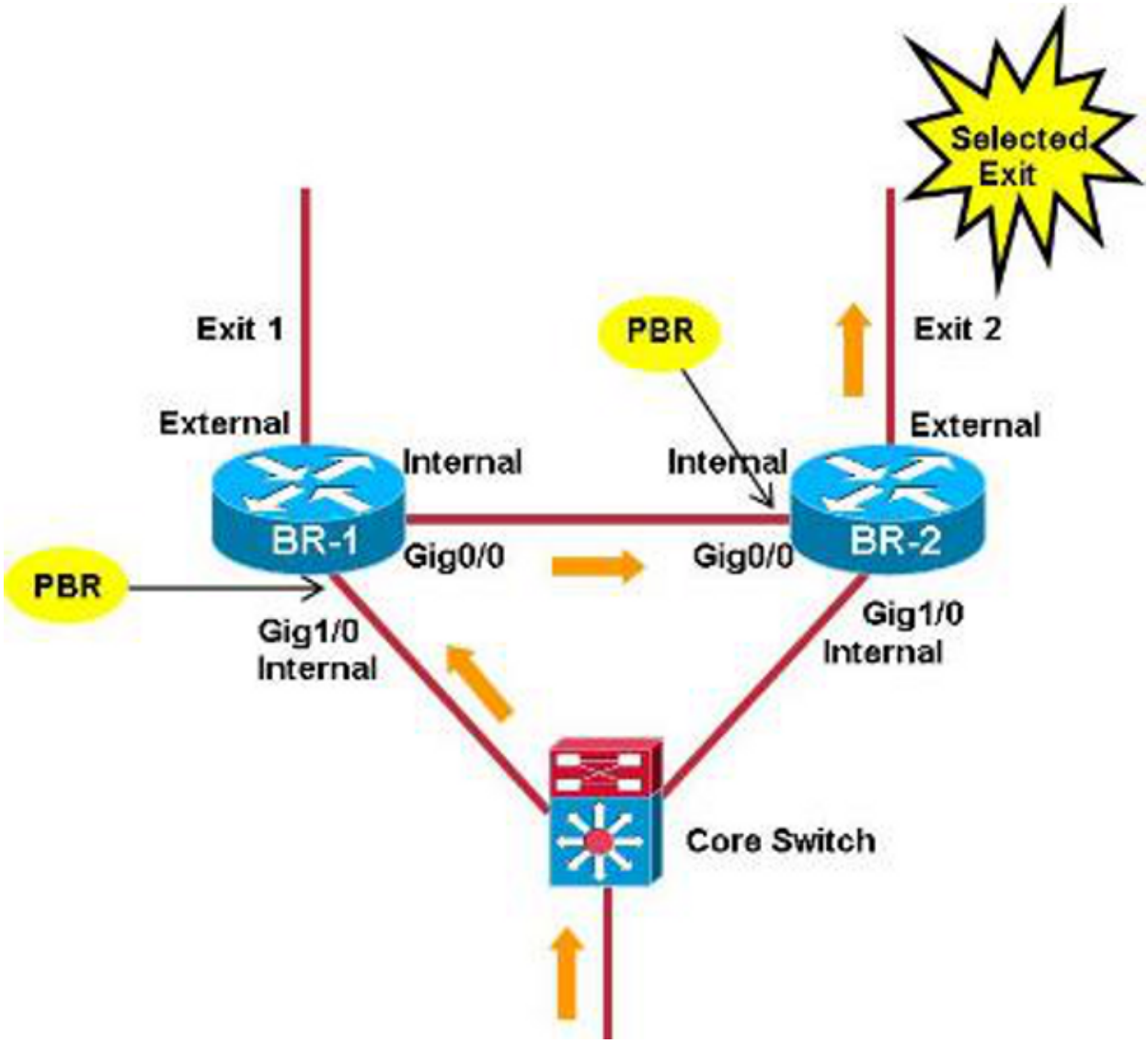
In order to check the EEM version supported by your Cisco IOS release, use this command:

```
Router#sh event manager version | i Embedded
Embedded Event Manager Version 4.00
Router#
```

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, make sure that you understand the potential impact of any command.

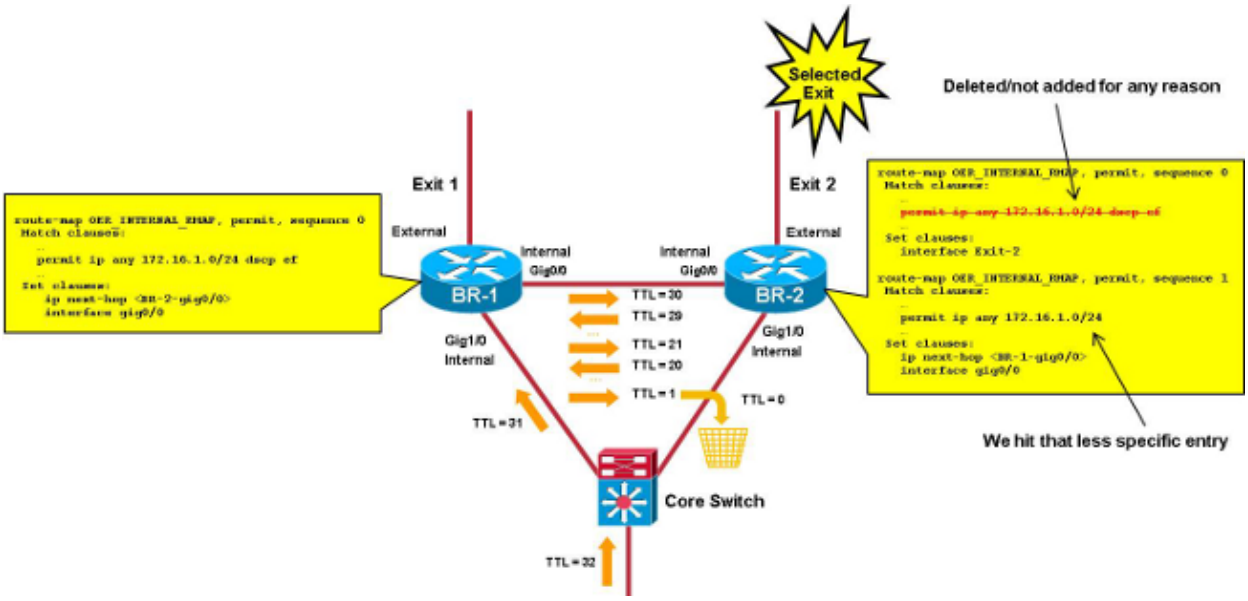
Background Information

When PfR controls a Traffic Class (TC), it creates a dynamic route map/Access Control List (ACL) on the BRs. The route map on a BR with a selected exit points to a selected exit, while a route map on other BRs points to an internal interface (next−hop = selected BR).



A problem occurs when the dynamic ACLs are not synched properly between the different BRs (due to bugs, for example).

In this picture, the focus is on TC matching any IP packets destined to 172.16.1.0/24 with DSCP EF. In this scenario, the related ACL entry is removed from the selected BR (BR−2), but not from BR−1. Packets of that TC hit on BR−2 with the prefix entry that matches all IP packets destined to 172.16.1.0/24. The selected exit for the prefix entry is ***Exit−1***, so the related route−map/ACL on BR−2 points to BR−1.



The packets of that TC now loop between the BRs untill the Time To Live (TTL) reaches 0.

This document provides the necessary EEM applets used in order to:

- Detect a forwarding loop between BRs
- Collect relevant information and clear the PfR

The applets used in the case of a Master Controller (MC)/BR combo are much easier (when MC runs on one of the BRs). The scenario with dedicated MCs is also covered.

EEM Applet Details

This section describes the Access−lists used for this process, as well as Applet Log files.

Access−Lists Used

In order to detect forwarding loops, the applet relies on an ACL to match packets with low TTL.

Note: ACL matching on TTL is supported on Aggregation Service Routers (ASR) 1000 Series Version 3.7s (15.2(4)S) and later.

It is recommended to use ACE matching on 2x consecutive, relatively low, TTL values (20 and 21) in order to get one (and only one) hit for each packet that loops between BRs. The TTL value used should not be too low in order to avoid frequent hits from traceroute packets.

```
interface gig0/0 (internal interface)
 ip access−group LOOP in
!
ip access−list extended LOOP
 permit ip 10.116.48.0 0.0.31.255 any ttl range 20 21
 permit ip any any
```

The ACL should be placed on the internal interface reported in the ***show pfr master border topology*** command output.

The source IP range (here 10.116.48.0/20) should match the internal network(s) (prefixes reachable via

internal interfaces).

Note: If you can not summarize internal networks in one Access−list Entry (ACE), you can use several ACEs; however, the script needs to be slightly modified in order to check hit counts on several lines.

Note: The ***auto−tunne***l feature needs to be turned off (***no mode auto−tunnels*** in master PfR mode). If the BRs are not directly connected, manual Generic Routing Encapsulation (GRE) tunnel(s) must be created and the ACL placed on the tunnel interface.

In order to identify which remote site/TC is impacted by the loop, you can add a second ACL outbound on the interface, with more specific ACEs for each remote site/TC.

```
interface gig0/0 (internal interface)
 ip access−group LOOP−DETAIL out

!
ip access−list extended LOOP−DETAIL

permit ip 10.116.48.0 0.0.31.255 10.116.132.0 0.0.0.255 ttl range 20 21
permit ip 10.116.48.0 0.0.31.255 10.116.128.0 0.0.0.255 ttl range 20 21
  .... (add here one line per remote site)
permit ip any an
```

The destination IP matches the subnet in the different remote sites:

```
10.116.132.0/24 −> site−1
10.116.128.0/24 −> site−2
```

You can also add several lines per remote site if you need to identify the exact TC impacted by the loop.

Applet Duties

The applet checks the hitcounts of the ACE matching on the TTL in the ACL loop every thirty seconds. Based on the outcome of these checks, the applet might perform these tasks:

- If the hitcounts exceed a configured threshold (THRESHOLD_1), the applet clears the ACL count and rechecks the hitcounts in fifteen seconds.
- After the fifteen seconds, if the hitcounts are above a second threshold (`THRESHOLD_2`), there might be a loop. You must collect a set of outputs and clear the PfR in order to fix the loop problem.
- The second thresholds are defined as global variables, so they are tuned easily without an applet restart.
- The optimum value for these thresholds mainly depends on the average packet rate per TC.

Applet Log Files

The applet maintains a log file that keeps track of the number of hitcounts (when the count is greater than 0), and any encounters of temporary loops (when THRESHOLD_1 is exceeded but not THRESHOLD_2) or a real loop (when both THRESHOLD_1 and THRESHOLD_2 are exceeded).

Applets for MC/BR Combo and Other BR Scenarios

These are the simplest scenarios described in this document. Loop detection and PfR clearing are done on the same device, so there is no need to enter device EEM applet communication. A separate applet runs on a MC/BR combo and other BRs.

Applet on MC/BR Combo

This output displays important information for the applet used on MC/BR combo. Here are some important notes for this specific output:

- The value shown for ***THRESHOLD_1*** is 1000, and the values shown for ***THRESHOLD_2*** is 500. This implies that the applet launches if the rate of the TC impacted by the loop is higher than 1000/30 (33 pps).
- The ***DISK*** variable identifies where the log and output files are pushed (shown here on bootflash).
- The timestamp of entries in the log file derive from the ***show clock*** command output. The characters in the middle (shown here as "est") depend on the timezone and must be adjusted (see ***action 240***).
- The outputs that must be collected in case of loop are pushed in the ***script−output−xxxxxxx*** file in bootflash, where "xxxxxx" is the number of seconds since 1970 (used to make unique file names for each loop occurrence).
- The commands collected are listed in ***actions 330, 340, 350, and 360***. Some further/different commands can be added.

```
event manager environment THRESHOLD_1 1000
event manager environment THRESHOLD_2 500
event manager environment DISK bootflash
!
event manager applet LOOP−MON authorization bypass
 event timer watchdog name LOOP time 30
 action 100 cli command "enable"
 action 110 cli command "show ip access−list LOOP"
 action 120 set regexp_substr 0
 action 130 regexp "range 20 21 \(([0−9]+) matches\)"
 $_cli_result _regexp_result regexp_substr
 action 140 cli command "clear ip access−list counters LOOP"
 action 150 if $regexp_substr gt 0
 action 200 set MATCHES $regexp_substr
 action 210 file open LOGS $DISK:script−logs.txt a
 action 220 cli command "enable"
 action 230 cli command "show clock"
 action 240 regexp "[0−9]+:[0−9]+:[0−9]+.[0−9]+ est [A−Za−z]+
  [A−Za−z]+ [0−9]+ 201[0−9]" $_cli_result _regexp_result
 action 250 set TIME $_regexp_result
 action 260 if $MATCHES gt $THRESHOLD_1
 action 270 wait 15
 action 280 cli command "show ip access−list LOOP"
 action 290 set regexp_substr 0
 action 300 regexp "range 20 21 \(([0−9]+) matches\)"
  $_cli_result _regexp_result regexp_substr
 action 310 if $regexp_substr gt $THRESHOLD_2
 action 320 cli command "enable"
 action 330 cli command "show ip access−list LOOP−DETAIL
  | tee /append $DISK:script−output−$_event_pub_sec.txt"
 action 340 cli command "show pfr master traffic−class perf det
  | tee /append $DISK:script−output−$_event_pub_sec.txt"
 action 350 cli command "show route−map dynamic detail
  | tee /append $DISK:script−output−$_event_pub_sec.txt"
 action 360 cli command "show ip route
  | tee /append $DISK:script−output−$_event_pub_sec.txt"
 action 370 cli command "clear pfr master *"
 action 380 cli command "clear ip access−list counters LOOP−DETAIL"
 action 390 file puts LOGS "$TIME − LOOP DETECTED − PfR CLEARED −
  matches $MATCHES > $THRESHOLD_1 and $regexp_substr
 > $THRESHOLD_2 − see $DISK:script−output−$_event_pub_sec.txt"
 action 400 syslog priority emergencies msg "LOOP DETECTED −
  PfR CLEARED − see $DISK:script−output−$_event_pub_sec.txt !"
 action 410 else
 action 420 file puts LOGS "$TIME − TEMPORARY LOOP : matches
```

```
 $MATCHES > $THRESHOLD_1 and $regexp_substr < or = $THRESHOLD_2"
action 430 cli command "clear ip access-list counters LOOP-DETAIL"
action 440 end
action 450 else
action 460 cli command "en"
action 470 cli command "clear ip access-list counters LOOP-DETAIL"
action 480 file puts LOGS "$TIME - number of matches =
 $MATCHES < $THRESHOLD_1"
action 490 end
action 500 else
action 510 cli command "clear ip access-list counters LOOP-DETAIL"
action 520 end
```

Applet for Other BRs

This section describes the applet used for other BRs. Here are some important notes for this specific output:

- The applet runs every twenty seconds while the script on an MC/BR combo runs every thirty seconds. This ensures that the applet on the BR launches before the PfR is cleared via the applet that runs on the MC/BR.
- A unique threshold is used, so there is no need to avoid fault positive.
- The value shown for the ***THRESHOLD*** is 700, and should be set according to the ***THRESHOLD_1*** value in the MC/BR applet.
- The applet log file is pushed in the ***script-logs.txt*** file in ***flash0***. This can be changed in ***action 170*** and the ***DISK*** variable.
- The timestamp of entries in the log file derive from the ***show clock*** command output. The characters in the middle (shown here as "est") depends on the timezone and must be adjusted (see ***action 190***).
- The outputs that must be collected in case of loop are pushed in the ***script-output-xxxxxxx*** file, where "xxxxxx" is the number of seconds since 1970 (used to make unique file names for each loop occurrence).
- The commands collected are listed in ***action 230*** and ***action 240***. Some further/different commands can be added.

```
event manager environment THRESHOLD 700
event manager environment DISK flash 0
!
event manager applet LOOP-BR authorization bypass
  event timer watchdog name LOOP time 20
 action 100 cli command "enable"
 action 110 cli command "show ip access-list LOOP"
 action 120 set regexp_substr 0
 action 130 regexp "range 20 21 \(([0-9]+) matches\)"
  $_cli_result _regexp_result regexp_substr
 action 140 cli command "clear ip access-list counters LOOP"
 action 150 if $regexp_substr gt 0
 action 160  set MATCHES $regexp_substr
 action 170  file open LOGS $DISK:script-logs.txt a
 action 180  cli command "show clock"
 action 190  regexp "[0-9]+:[0-9]+:[0-9]+.[0-9]+
  est [A-Za-z]+ [A-Za-z]+ [0-9]+ 201[0-9]" $_cli_result _regexp_result
 action 200  set TIME $_regexp_result
 action 210  if $MATCHES gt $THRESHOLD
 action 220   cli command "enable"
 action 230   cli command "show route-map dynamic detail | tee /append
  $DISK:script-output-$_event_pub_sec.txt"
 action 240   cli command "show ip route | tee /append
  $DISK:script-output-$_event_pub_sec.txt"
 action 250   file puts LOGS "$TIME : matches = $MATCHES >
  $THRESHOLD - see $DISK:script-output-$_event_pub_sec.txt"
 action 260   syslog priority emergencies msg "LOOP DETECTED -
  Outputs captured - see $DISK:script-output-$_event_pub_sec.txt !"
```

```
action 270  else
action 280   file puts LOGS "$TIME : matches = $MATCHES < or = $THRESHOLD"
action 290  end
action 300 end
```

Applets for Dedicated MC Scenario

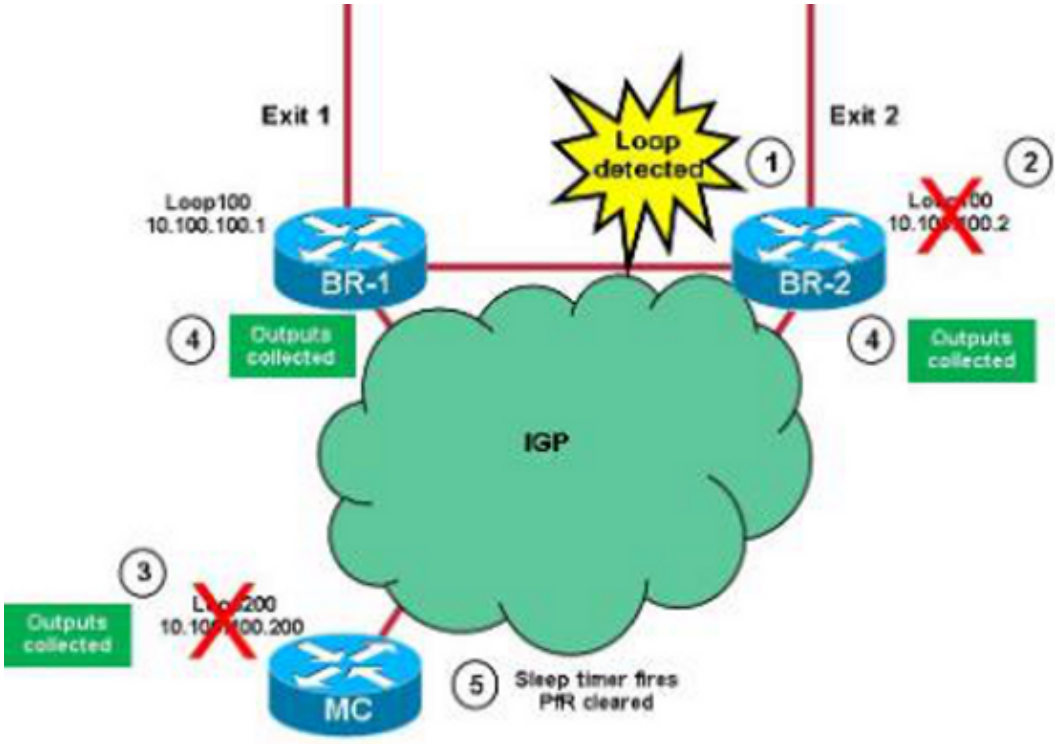
The loop detection and PfR clearing/stats collection is completed on different devices that must have inter−device EEM applet communication. The communication between the devices occurs in different ways. This document describes device communication via objects tracked in order to check reachability of dedicated loopbacks advertised in IGP. When an event is detected, the loopback is shut down, which allows applets on remote devices to launch when the object that is tracked goes offline. You can use different loopbacks if different information needs to be exchanged.

Applet Communication

These applets and communication methods are used:

Applet Name	*Where ?*	*What ?*	*Trigger ?*	*Communication ?*
LOOP−BR	BRs	Check ACL hitcounts in order to detect loops	Periodic	shut Loop100
LOOP−MC	MC	− Collect PfR infos − Clears PfR	Track reachability Loop100	shut Loop200
COLLECT−BR	BRs	Collect Information	Track reachability Loop200	none

Here is an image that illustrates this:



This is the process used by the applets:

1. A loop is detected by the ***LOOP−BR*** applet on the BRs. It is assumed that the loop is detected on BR−2 first.
2. The applet shuts ***Loop100*** on BR−2, and the information is advertised on the Interior Gateway Protocol (IGP).
3. The tracked object for ***Loop100*** of BR−2 goes offline on the MC, and the ***LOOP−MC*** applet starts. PfR master outputs are collected, and ***Loopback 200*** on the MC is shut down. The information is advertised on IGP. A ten−second sleep timer begins.
4. The tracked object for ***Loop200*** on the MC goes offline on both BRs. This triggers the ***COLLECT−BR*** applet that collects BR−specific information.
5. The sleep timer (Step 3) begins, and the MC clears the PfR.

Note: If BR−1 detects the loop before the PfR is cleared, the tracked object that goes offline is ignored on MC (the ***LOOP−MC*** applet runs once a minute).

Create Track Objects and Loopbacks

This section describes how to create loopbacks (ensure the IPs are advertised on the IGP) and track objects.

Track Objects

Here are some important points to keep in mind when you create track objects:

- A single track object is needed on BRs, which is used in order to track ***loopback200*** on MC (this triggers data collection).
- Several track objects are needed on MC:
    - Tracks 1 and 2 are used in order to track ***loopback100*** on BR−1 and BR−2, respectively.
    - Tracks 11 and 12 are used in order to track connectivity between BR−1 and BR−2, respectively (avoids PfR clearing when there are connectivity issues between BRs).
    - Track 20 tracks the logical AND between tracks 11 and 12. This is used in order to verify that MC gets reachability to all BRs.
- The ***track timer ip route*** value is set to one second in order to speed up reachability problem detection (the default value is 15 seconds).

BR−1

```
interface Loopback100
  ip address 10.100.100.1 255.255.255.255
!
track timer ip route 1
track 1 ip route 10.100.100.200 255.255.255.255 reachability
```

BR−2

```
interface Loopback100
 ip address 10.100.100.2 255.255.255.255
!
track timer ip route 1
track 1 ip route 10.100.100.200 255.255.255.255 reachability
```

MC

```
interface Loopback200
 ip address 10.100.100.200 255.255.255.255
!
```

```
track timer ip route 1

track 1 ip route 10.100.100.1 255.255.255.255 reachability
track 2 ip route 10.100.100.2 255.255.255.255 reachability
track 11 ip route 10.116.100.1 255.255.255.255 reachability
track 12 ip route 10.116.100.2 255.255.255.255 reachability
track 20 list boolean and
 object 11
 object 12
```

BR and MC Loopbacks

LOOP−BR

This section describes how to create loopbacks on the BRs. Here are some important notes to keep in mind:

- The ***THRESHOLD_1*** value is 1000 and the ***THRESHOLD_2*** value is 500. This implies that the applet launches if the rate of the TCs impacted by the loop is higher than 1000/30 (33 p/s).
- The applet log file is pushed in the ***script−detect−logs.txt*** file in bootflash. This is changed in ***action 210*** and with the ***DISK*** variable.
- The timestamp of entries in the log file derives from the ***sh clock*** output. The characters in the middle (shown as 'est') depend on the timezone and require adjustment (***action 240***).
- After you close the ***Loopback100*** in order to notifiy MC, wait five seconds (in order to ensure IGP has time to propagate the information) and reopen it (***action 370***).

```
event manager environment THRESHOLD_1 100event manager environment
  THRESHOLD_2 500event manager environment DISK bootflash
!event manager applet LOOP−BR authorization bypass

 event timer watchdog name LOOP time 30 maxrun 27
 action 100  cli command "enable"
 action 110  cli command "show ip access−list LOOP"
 action 120  set regexp_substr 0
 action 130  regexp "range 20 21 \(([0−9]+) matches\)"
  $_cli_result _regexp_result regexp_substr
 action 140  cli command "clear ip access−list counters LOOP"
 action 150  if $regexp_substr gt 0
 action 200  set MATCHES $regexp_substr
 action 210  file open LOGS $DISK:script−detect−logs.txt a
 action 220  cli command "enable"
 action 230  cli command "show clock"
 action 240  regexp "[0−9]+:[0−9]+:[0−9]+.[0−9]+
  est [A−Za−z]+ [A−Za−z]+ [0−9]+ 201[0−9]"
  $_cli_result _regexp_result
 action 250  set TIME $_regexp_result
 action 260  if $MATCHES gt $THRESHOLD_1
 action 270  wait 15
 action 280  cli command "show ip access−list LOOP"
 action 290  set regexp_substr 0
 action 300  regexp "range 20 21 \(([0−9]+) matches\)"
  $_cli_result _regexp_result regexp_substr
 action 310  if $regexp_substr gt $THRESHOLD_2
 action 320  cli command "enable"
 action 330  cli command "conf t"
 action 340  cli command "interface loop100"
 action 350  cli command "shut"
 action 360  file puts LOGS "$TIME − LOOP DETECTED − Message sent to MC −
  matches $MATCHES > $THRESHOLD_1 and $regexp_substr > $THRESHOLD_2"
 action 370  wait 5
 action 375  cli command "enable"
 action 380  cli command "conf t"
 action 390  cli command "interface loop100"
```

```
 action 400  cli command "no shut"
 action 410  else
 action 420  file puts LOGS "$TIME − TEMPORARY LOOP : matches $MATCHES >
  $THRESHOLD_1 and $regexp_substr < or = $THRESHOLD_2"
 action 430  cli command "clear ip access−list counters LOOP−DETAIL"
 action 440  end
 action 450  else
 action 460  cli command "en"
 action 470  cli command "clear ip access−list counters LOOP−DETAIL"
 action 480  file puts LOGS "$TIME − number of matches =
  $MATCHES < $THRESHOLD_1"
 action 490  end
 action 500  else
 action 510  cli command "clear ip access−list counters LOOP−DETAIL"
 action 520  end
```

LOOP−MC

This section describes how to create loopbacks on the MC. Here are some important notes to keep in mind:

- The ratelimit value depends on how often the applet runs with a ratelimit value of 60 (script runs once per minute max). This is used in order to avoid the PfR clearing twice when the same loop is detected by both BRs.
- In ***action 130***, wait two seconds before you check the reachability to all BRs. This is in order to avoid a false positive caused by connectivity issues between the MC and BRs.
- In ***action 240***, wait ten seconds after you shut down ***Loopback200***, before you clear the PfR. This is in order to make sure that the BRs have time to collect the data.

```
event manage environment DISK bootflash
event manager applet LOOP−MC authorization bypass

event syslog pattern "10.100.100.[0−9]/32 reachability Up−>Dow" ratelimit 60
 action 100 file open LOGS $DISK:script−logs.txt a
 action 110 regexp "10.100.100.[0−9]" "$_syslog_msg" _regexp_result
 action 120 set BR $_regexp_result
 action 130 wait 2
 action 140 track read 20
 action 150 if $_track_state eq "up"
 action 160  cli command "enable"
 action 170  cli command "show clock"
 action 180  regexp "[0−9]+:[0−9]+:[0−9]+.[0−9]+
  est [A−Za−z]+ [A−Za−z]+ [0−9]+ 201[0−9]"
  "$_cli_result" _regexp_result
 action 190  set TIME "$_regexp_result"
 action 200  cli command "show pfr master traffic−class perf det
  | tee /append $DISK:script−output−$_event_pub_sec.txt"
 action 210  cli command "conf t"
 action 220  cli command "interface loop200"
 action 230  cli command "shut"
 action 240  wait 10
 action 250  cli command "conf t"
 action 260  cli command "interface loop200"
 action 270  cli command "no shut"
 action 280  cli command "end"
 action 290  cli command "clear pfr master *"
 action 300  file puts LOGS "$TIME − LOOP DETECTED by $BR −
  PfR CLEARED − see $DISK:script−output−$_event_pub_sec.txt"
 action 310  syslog priority emergencies msg "LOOP DETECTED by $BR −
  PfR CLEARED − see $DISK:script−output−$_event_pub_sec.txt !"
 action 320 else
 action 330  file puts LOGS "$TIME − REACHABILITY LOST with
  $BR − REACHABILITY TO ALL BRs NOT OK − NO ACTION"
 action 340 end
```

COLLECT−BR

This section describes how to collect the BR. The applet launches in when a BR looses reachability to ***Loopback200*** (10.100.100.200) on MC. The commands used in order to collect are listed in ***actions 120***, ***130***, and ***140***.

```
event manager environment DISK bootflash
event manager applet COLLECT−BR authorization bypass

event syslog pattern "10.100.100.200/32 reachability Up−>Dow" ratelimit 45
 action 100 file open LOGS $DISK:script−collect−logs.txt a
 action 110 cli command "enable"
 action 120 cli command "sh ip access−list LOOP−DETAIL |
  tee /append $DISK:script−output−$_event_pub_sec.txt"
 action 130 cli command "show route−map dynamic detail
  | tee /append $DISK:script−output−$_event_pub_sec.txt"
 action 140 cli command "show ip route | tee /append
  $DISK:script−output−$_event_pub_sec.txt"
 action 150 cli command "show clock"
 action 160 regexp "[0−9]+:[0−9]+:[0−9]+.[0−9]+ CET [A−Za−z]+ [A−Za−z]+
  [0−9]+ 201[0−9]" "$_cli_result" _regexp_result
 action 170 set TIME "$_regexp_result"
 action 180 file puts LOGS "$TIME − OUTPUTs COLLECTED −
  see $DISK:script−output−$_event_pub_sec.txt"
```

SYSLOG−MC

Here is the syslog on MC when a loop is detected:

```
MC#
*Mar  8 08:52:12.529: %TRACKING−5−STATE: 1 ip route 10.100.100.1/32
 reachability Up−>Down
MC#
*Mar  8 08:52:16.683: %LINEPROTO−5−UPDOWN:
 Line protocol on Interface Loopback200, changed state to down
*Mar  8 08:52:16.683: %LINK−5−CHANGED: Interface Loopback200,
 changed state to administratively down
MC#
*Mar  8 08:52:19.531: %TRACKING−5−STATE: 1
 ip route 10.100.100.1/32 reachability Down−>Up
MC#
*Mar  8 08:52:24.727: %SYS−5−CONFIG_I: Configured from console by
  on vty0 (EEM:LOOP−MC)
*Mar  8 08:52:24.744: %PFR_MC−1−ALERT: MC is inactive due to PfR
 minimum requirements not met;
 Less than two external interfaces are operational
MC#
*Mar  8 08:52:24.757: %HA_EM−0−LOG: LOOP−MC:
 LOOP DETECTED by 10.100.100.1 − PfR CLEARED
 − see unix:script−output−1362732732.txt !
MC#
*Mar  8 08:52:26.723: %LINEPROTO−5−UPDOWN:
 Line protocol on Interface Loopback200, changed state to up
MC#
*Mar  8 08:52:26.723: %LINK−3−UPDOWN: Interface Loopback200,
 changed state to up
MC#
*Mar  8 08:52:29.840: %PFR_MC−5−MC_STATUS_CHANGE: MC is UP
*Mar  8 08:52:30.549: %TRACKING−5−STATE: 2
 ip route 10.100.100.2/32 reachability Up−>Down
MC#
*Mar  8 08:52:37.549: %TRACKING−5−STATE: 2
 ip route 10.100.100.2/32 reachability Down−>Up
MC#
```

Note: These applets can be used with three or more BRs with some tuning.

Updated: Sep 25, 2013 Document ID: 116206