monRoutePing: Controlling an Interface Availability using Pings to IP Targets

Issue 1.3

Date 09 July 2014

1 Introduction

This document describes how to use the failover script monRoutePing to control an interface availability using pings to IP targets. This is typically used for an Ethernet or Bridged link but can be used for any interface type.

If failure is detected, the default route metric for this link is reduced to allow other configured default routes to take priority.

Note: in an Ethernet or Bridged environment this script is generally used if the requirement is to monitor an IP beyond the next hop. If monitoring the next hop modem to determine Ethernet connectivity is sufficient, the default route Query Next Hop for Numbered Route Availability feature should be used.

The script can be used in a scenario as below. The primary connection is out an Ethernet interface to a 3rd party gateway device (ADSL, Satellite, WiFi, etc.)

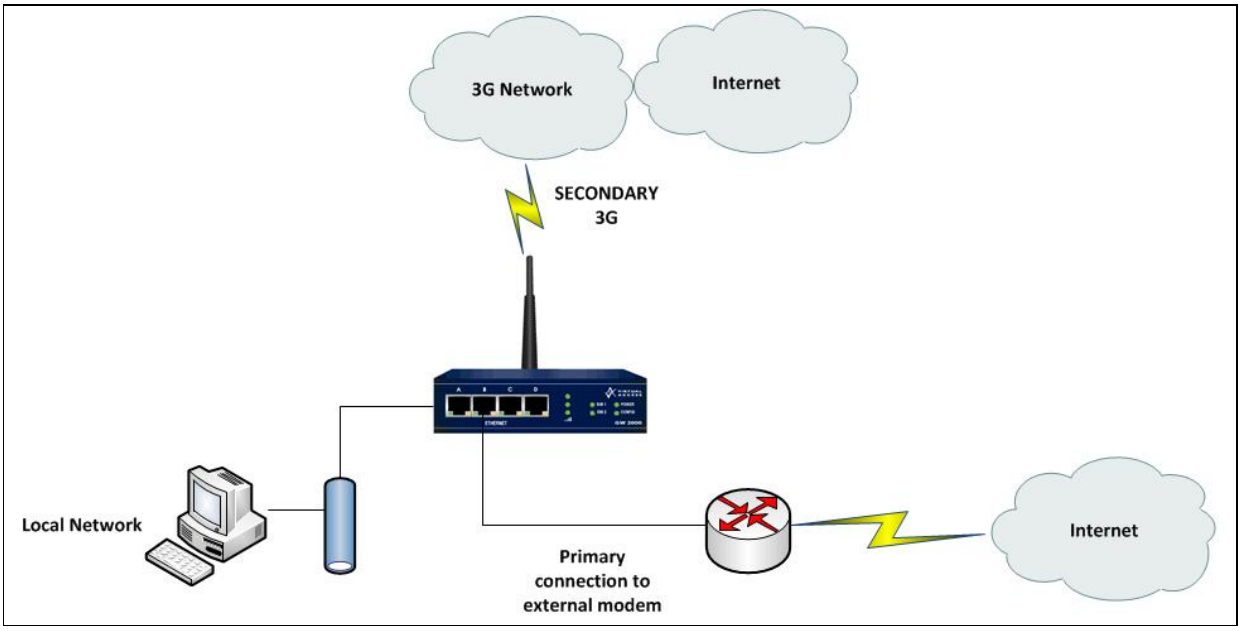


Figure 1: Network architecture

2 Configuring the monRoutePing script

2.1 Script overview

The script can only be run once. It is designed to be run on boot. On boot the script does the following:

- The script waits for a specified time to allow interfaces to connect.
- Sends pings to target destinations at configurable durations. If only one target destination is configured then only one ping is sent to this target. Pings are sourced from the eth-0 interface to allow for routing across a VPN tunnel if required and allow for a reply wait time to be specified.
- A ping from either target signifies that the router is operating as normal.
- When a number of configurable consecutive ping failures are detected:
  - the monitored interface default route metric is increased to 15 to allow other default routes to take priority. Note: If two ping targets are configured, pings must fail to both targets consecutively. An INFO event is generated for visibility of the routing change and also to allow other scripts to fire where required. A backup interface is optionally manually connected.
- The script optionally waits for a configurable period to avoid flapping on an unreliable primary connection.
- Pings continue to be sent out the Ethernet link to allow fall back. When a number of consecutive pings are successful:
  - The monitored interface metric is restored to required metric. An INFO event is generated for visibility of routing change and also to allow other scripts to fire where required. A backup interface is optionally manually disconnected.

The script automatically configures:

- a numbered static route at index 11 to the primary ping target out the monitored interface.
- a numbered static route at index 12 to the second ping target out monitored interface, if second ping target is configured.
- an advanced filter at index 3 to pass test pings to ping targets out the monitored interface.
- an advanced filter at index 4 to block test pings to ping targets out any other interface.
- An IP table named [monRtPing] at ip table lines 97, 98 and 99 for ping targets.

2.2 Script requirements

- This script must not be run in conjunction with any other ping monitoring script.
- This script is embedded in firmware versions 9.08.99 and greater.

2.3 Script parameters

The script name is monRoutePing and it takes in eight required parameters and a further four optional parameters:

monRoutePing [test addr1] [test addr2] [primary interface] [default route index] [route metric] [ping threshold] [ping wait] [init wait] [force connect] [force connect interface] [ping reply wait]

These parameters are described in the example and table below.

monRoutePing 1.1.1.1, 2.2.2.2, eth-1, 1, 1, 6, 10, 60, 1, ppp-1, 2, 0

Parameter	Type	Description
1.1.1.1	Required	The first ping target.
2.2.2.2	Required	The second ping target. Set to 0 to signify no second ping target.
eth-1	Required	The interface to be monitored.
1	Required	The default route index for the interface default route to be monitored.
1	Required	The online metric for the interface default route to be monitored (used to ensure not incorrectly saved while in backup).
6	Required	The number of consecutive pings failures to cause recovery procedure.
10	Required	The wait between pings in seconds. Pings are sent to both ping targets at the same time.
60	Required	The wait on boot before sending the first ping.
1	Optional	Whether to manually connect and disconnect an interface when monitored route goes offline and online (0 - connect, 1 - no connect; default: 0).
ppp-1	Optional	The interface to manually connect and disconnect when monitored route goes offline and online. (default: ppp-1).
2	Optional	The time to wait for a ping reply in seconds (default: 2).
0	Optional	The minimum time to wait in secs after failover to backup before falling back to primary to avoid flapping on an unreliable primary connection (default: 0 [0 means no wait])

Table 1: monRoutePing parameter descriptions

2.4 Configuring the script

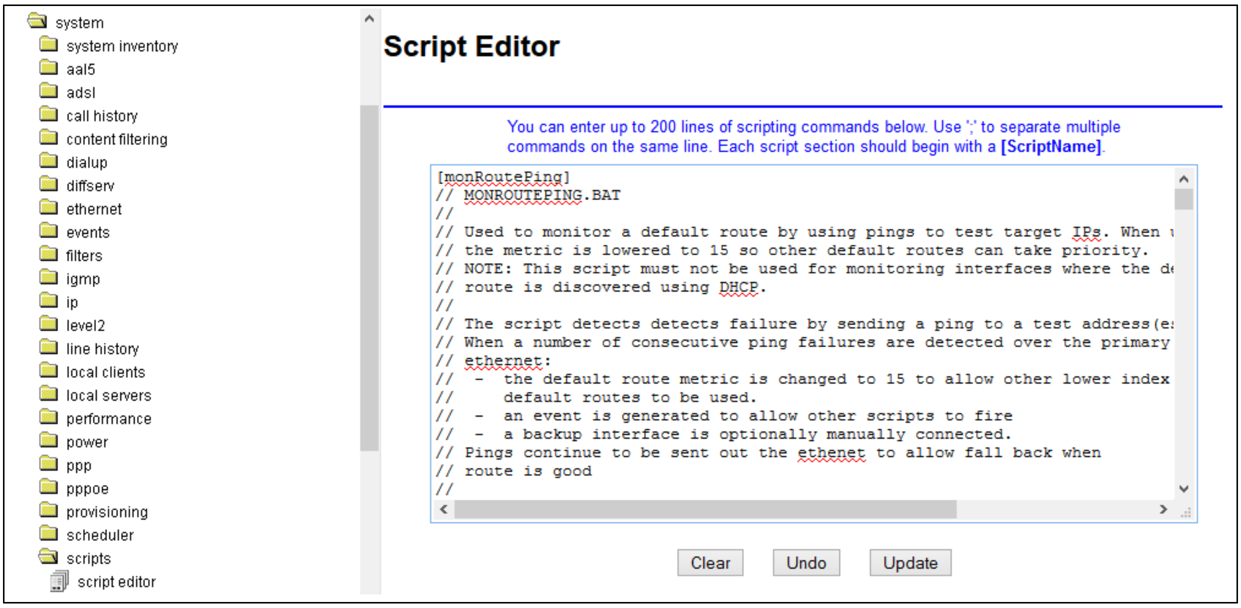
This script was introduced into firmware in versions 9.08.99. To use the script on older firmware versions first paste the script from Section5 'monRoutePing script' into the script editor and then use the scheduler to run the script at boot up.

To open the Expert View menu, from the start page, click **Advanced**.

2.4.1 Pasting the script into the script editor

If you are using 9.09.xx firmware, in the Expert View menu, click **system > scripts->script editor**. The Script Editor page appears.

If you are using 10.00.xx firmware, in the Expert View menu, click **system > management > scripts > script editor.** The Script Editor page appears.



Script Editor

You can enter up to 200 lines of scripting commands below. Use ';' to separate multiple commands on the same line. Each script section should begin with a **[ScriptName]**.

```
[monRoutePing]
// MONROUTEPING.BAT
//
// Used to monitor a default route by using pings to test target IPs. When
// the metric is lowered to 15 so other default routes can take priority.
// NOTE: This script must not be used for monitoring interfaces where the de
// route is discovered using DHCP.
//
// The script detects detects failure by sending a ping to a test address(e
// When a number of consecutive ping failures are detected over the primary
// ethernet:
//  -  the default route metric is changed to 15 to allow other lower index
//     default routes to be used.
//  -  an event is generated to allow other scripts to fire
//  -  a backup interface is optionally manually connected.
// Pings continue to be sent out the ethenet to allow fall back when
// route is good
//
```

Figure 2: The script editor page in version 9.09.xx

Paste in the script from Section5 'monRoutePing script' from this document. The first line of the script should begin with the script name in square brackets, [monRoutePing]. This name will be used to call the script using the scheduler.

You can omit any line beginning with // (denotes a comment tag) if the number of script lines needs to be reduced. Also multiple script lines can be entered onto the same script editor line separated by ';' (semi colon). When you have completed the script, click **Update**.

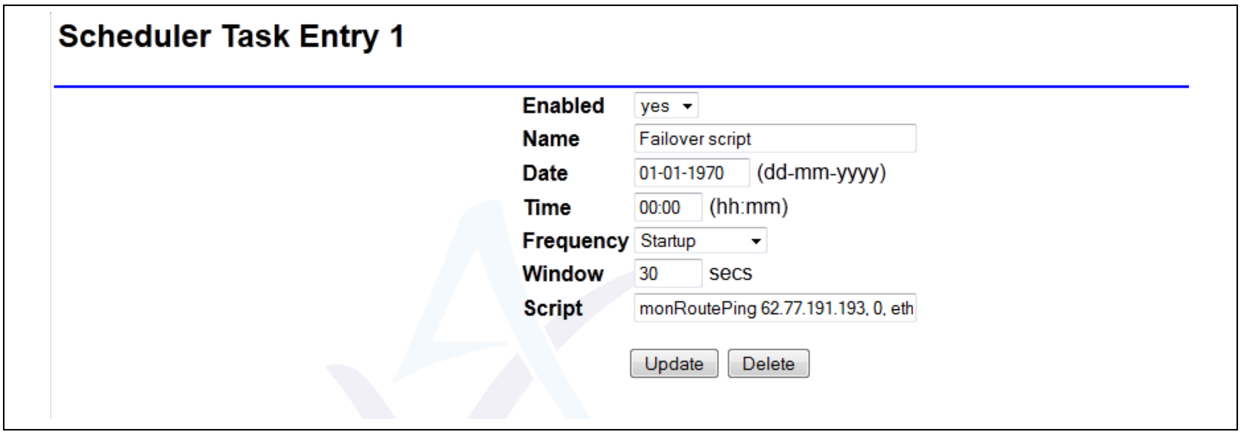
2.4.2 Scheduling the script to run on boot

If you are using 9.90.xx firmware, in the Expert View menu, click **system > scheduler > scheduler tasks**. The Scheduler Task Entry page appears.

Click **Add** or **modify/delete** to access the scheduler Task Entry Page.

If you are using 10.00.xx firmware, in the Expert View menu, click s**ystem > management > scheduler > scheduler tasks**. The Scheduler Task Entry page appears.

Click **Add** or **modify/delete** to access the scheduler Task Entry Page.



Scheduler Task Entry 1

Enabled	yes
Name	Failover script
Date	01-01-1970 (dd-mm-yyyy)
Time	00:00 (hh:mm)
Frequency	Startup
Window	30 secs
Script	monRoutePing 62.77.191.193, 0, eth

Figure 3: The scheduler task entry page

Field	Description
Enabled	Enables or disables a particular schedule. Set to **Yes**.
Name	The name associated with the schedule. Enter a descriptive name
Date	The date the script initiates. This field is ignored when frequency is set to start up. Leave at **default**.
Time	The time the script initiates. This field is ignored when frequency is set to start up. Leave at **default**.
Frequency	Sets the frequency the script executes. Set to **startup**.
Window	This parameter sets how long the system will wait if it is busy before executing the script. For example if the script is set to execute at 10:00 and the window is set to 30 seconds, the system will try executing the script within this window only. Set to **30**.
Script	The name of the script to be executed. Enter the script name, followed by the relevant parameters as shown in the above image. Separate the parameters by commas. Example: **monRoutePing 1.1.1.1, 2.2.2.2, eth-1, 1, 1, 6, 10, 60, 1, ppp-1, 2**

Table 2: The scheduler task fields and their descriptions

3 Debugging commands

Useful debug commands via command line are described in the table below.

Diagnostic Command	Description
Show tasks	Shows all running tasks
Show task <tasknum>	Shows running task. Also indicates position task is currently at.
Show task vars <tasknum>	Shows variables and variable values associated with task.
Show ip route	Shows routing table
Show ip addresses	Shows all interface IP address
Show active virtual route hits	Shows active virtual routes and hits count
Show iptable	Shows configured IP table names
Show iptable <iptable_name>	Shows IP addresses/subnets for <iptable_name> IP table
Show events	Shows event log
Dir scripts	Shows all scripts embedded in the firmware
Show monRoutePing.bat	Displays monRoutePing script
Show config script ALL	Displays all scripts in the script editor.
Show config script <scriptname>	Displays the <scriptname> script as configured in script editor. Includes line numbers.

Table 3: Debug command lines and their descriptions

Useful trace commands via the command line are described in the table below.

Trace command	Description
++All 6	Traces all INFO events
++ip	Traces IP traffic
++ip:icmp	Traces ICMP IP traffic
++script	Traces script events
--script	Stops script event tracing
--	Stops all event tracing
Trace on <script_name>	Traces each line in a script as it executes
Trace off <script_name>	Turns off tracing for script

Table 4: Trace command lines and their descriptions

4 Script events

Severity	Class	Subclass	Text
INFO	49	40	monRoutePing script error invalid ping test addresses
INFO	49	40	monRoutePing monitoring <interface> with <ping target IPs> (route currently online)
INFO	49	40	monRoutePing monitoring <interface> with <ping target IPs> (route currently offline)
INFO	49	40	monRoutePing incorrect default route index <route_index> (route disabled)
INFO	49	40	monRoutePing added adv filters 3/4 for <ping target IPs>out <interface>
INFO	49	40	routeAdd script error invalid index <route_index>
INFO	49	9999	routeAdd index <route_index> <target>/<mask>/<metric> via <nextHop>
INFO	49	40	monRoutePing added routes <route_index> for <ping target IPs> out <interface>
INFO	49	40	monRoutePing route down <interface> default route <route_index> metric now <offline metric>
INFO	49	40	monRoutePing route up <interface> default route <route_index> metric now <online metric>
INFO	49	40	monRoutePing no fallback for <fallback_guard_time>seconds

Table 5: Script events

5 monRoutePing script

```
[monRoutePing]
// MONROUTEPING.BAT
//
// Used to monitor a default route by using pings to test target IPs. When unavailable
// the metric is lowered to 15 so other default routes can take priority.
// NOTE: This script must not be used for monitoring interfaces where the default
// route is discovered using DHCP.
//
// The script detects detects failure by sending a ping to a test address(es).
// When a number of consecutive ping failures are detected over the primary
// ethernet:
//  -  the default route metric is changed to 15 to allow other lower index
//     default routes to be used.
//  -  an event is generated to allow other scripts to fire
//  -  a backup interface is optionally manually connected.
// Pings continue to be sent out the ethenet to allow fall back when
// route is good
//
// usage: monRoutePing   [test addr1] [test addr2] [primary interface]
//                       [default route index] [route metric] [ping threshold]
//                       [ping wait] [init wait] [force connect]
//                       [force connect interface] [ping reply wait]
//                       [fallback guard time]
//
// The monRoutePing.bat script MUST take eight parameters:
// - the IP or DNS address used to determine valid ethernet route.
// - a second IP or DNS address used to determine valid ethernet route
//   (set as 0 if do not want a second test address)
// - the primary interface NB: must be eth-x/ppp-x NOT ex/px
// - the default route index that is to be monitored
// - the metric for the monitored default route when online
//   (used to ensure not incorrectly saved while in backup)
// - the number of consecutive pings that signifies primary link down.
//   (this value will also be used to determine link up again).
// - the wait after pinging all test addresses before pinging again (0 for no wait)
// - the initial wait before starting the test pings
//
// The monRoutePing.bat script can OPTIONALLY take a futher four parameters:
// - whether to force a connect on an interface when the monitored route is offline
//   (0 - connect, 1 - no connect) Default: 0
//   Note: also disconnects monitored route is online again
// - the interface to force a connect. Default: ppp-1
// - the time to wait for a ping reply (def: 2 secs)
// - the minimum time to wait in secs after failover to backup before falling
//  back to primary to avoid flapping on an unreliable primary connection
//  (def: 0 -[ 0 means no wait])
//
// Configuration
// -------------
// This script will automatically configure:
// - static route(s) at index 11(/12) to test addr1/addr2 out primary interface
// - an advanced filter at index 3 to pass test pings to test addr 1 and 2
//   out the primary interface
// - an advanced filter at index 4 to block test pings to test addr 1 and 2
//   out any other interface
// - IP table lines 94, 95 and 96 for test pings
//
// EXAMPLES
// --------
// monRoutePing 1.1.1.1, 2.2.2.2, eth-1, 1, 1, 6, 10, 60
// On 6 consecutive ping failures to 1.1.1.1 AND 2.2.2.2 eth-1 route index 1
// is changed to 15. Ping wait time is 2 secs.
// On 6 consecutive ping sucesses to either 1.1.1.1 OR 2.2.2.2 eth-1 route
// index 1 metric is restored.
// Script waits 60 secs before sending the first ping and waits 10 secs
// after pinging both destinations.
//


!unique
!arg testAddr1, testAddr2, primaryIf, rtIndex, rtMetric, pingCount,  pingWait, initWait
```

```
$force_connect = $9
$force_connect_itf = $10
$ping_reply_wait = $11
$fallback_wait= $12

!if force_connect = ''
  $force_connect = 0
!endif
!if force_connect_itf = ''
  $force_connect_itf = ppp-1
!endif
!if ping_reply_wait = ''
  $ping_reply_wait = 2
!endif
!if ping_reply_wait = 0
  $ping_reply_wait = 2
!endif
!if fallback_wait = ''
  $fallback_wait = 0
!endif
$i = 1
$ping_reply_wait_msec = 1000
!while $i < $ping_reply_wait
  !add ping_reply_wait_msec, 1000
  !inc i
!endwhile

//checking
!if testAddr1 = ''
  $testAddr1 = 0
!endif
!if testAddr2 = ''
  $testAddr2 = 0
!endif
```

```
!if $testAddr1 = 0
  !if $testAddr2 = 0
    !log monRoutePing script error invalid ping test addresses
    !exit
  !else
   $testAddr1 = $testAddr2
   $testAddr2 = 0
  !endif
!endif

$offlineMetric = 15
$logpingstr = $testAddr1
!if $testAddr2 <> 0
  $logpingstr = $logpingstr/$testAddr2
!endif

//checking
!if "`sh ip route default indexed configured $rtIndex`" = "yes"
   //numbered route?
   !if "`sh ip route default indexed numbered $rtIndex`" = "yes"
     $rtNumbered = 1
     $nextHop = `sh ip route default indexed next hop ip $rtindex`
   !else
     $rtNumbered = 0
     $nextHop = $primaryIf
   !endif
   //route currently online or offline?
   !if `sh ip route default indexed metric $rtIndex` <> $offlineMetric
     $rtActive = 1
     !log monRoutePing monitoring $primaryIf with $logpingstr (route currently online)
   !else
     $rtActive = 0
     !log monRoutePing monitoring $primaryIf with $logpingstr (route currently offline)
   !endif
!else
  !log monRoutePing incorrect default route index $rtIndex (route disabled)
  !exit
!endif

//create IP tables (94, 95, 96)
$z = `set iptable line 94 [monRtPing]
$z = `Set Iptable Line 95,$testAddr1
!if $testAddr2 <> 0
  $z = `Set Iptable Line 96,$testAddr2
!else
  $z = `set Iptable Line 96,`
!endif
//add adv filters (3,4)
$z = `Set filter IP 3 name monRtPingpass pass out on $primaryIf proto icmp from any to (monRtPing)`
$z = `Set filter IP 4 name monRtPingBlock block out on any proto icmp from any to (monRtPing)`
!log monRoutePing added adv filters 3/4 for $logpingstr out $primaryIf

//add static routes
$staticroutenum = 11
!if $testAddr2 <> 0
  $staticroutenum = $staticroutenum/12
  !call routeAdd.bat 12, $rtNumbered, $testAddr2, 255.255.255.255, 1, $nextHop, 0
!else
  $z = `set ip route static configured 12 no`
!endif
!call routeAdd.bat 11, $rtNumbered, $testAddr1, 255.255.255.255, 1, $nextHop, 1
!log monRoutePing added routes $staticroutenum for $logpingstr out $primaryIf

!pause $initWait

$z = `dl $$.ip none`
$pingFail = 0
$pingPass = 0
!while 1

  //route check
  $z = `st ping results reset`
```

```
$z = `quiet ping $testAddr1 -w $ping_reply_wait_msec`
!if $testAddr2 <> 0
  $z = `quiet ping $testAddr2 -w $ping_reply_wait_msec`
!endif
!pause $ping_reply_wait
$result = `sh ping replies`
!if $result > 0
  $pingFail = 0
  !inc pingPass
!else
  $pingPass = 0
  !inc pingFail
!endif

!if $rtActive <> 1
  !if $pingPass >= $pingCount
    $z = `set ip route default indexed metric $rtIndex $rtMetric`
    $z = `commit`
    $z = `set ip rt reconfigure`
    !log monRoutePing route up $primaryIf default route $rtIndex metric now $rtMetric
    //reset fail count
    $pingFail = 0
    $rtActive = 1
    !if $force_connect <> 0
      $z = `disconnect $force_connect_itf`
    !endif
  !endif
!else
  !if $pingFail >= $pingCount
    $z = `set ip route default indexed metric $rtIndex $offlineMetric
    $z = `commit`
    $z = `set ip rt reconfigure`
    !log monRoutePing route down $primaryIf default route metric now $offlineMetric
    //reset pass count
    $pingPass = 0
    $rtActive = 0
```

```
   !if $force_connect <> 0
    $z = `connect $force_connect_itf`
   !endif
   !if $fallback_wait <> 0
    !log monRoutePing no fallback for $fallback_wait seconds
    !pause $fallback_wait
   !endif
  !endif
 !endif

 !pause $pingWait
!endwhile
```

6 Script history

Version	Changes	Firmware version changes introduced
1.3	Addition of guard time to stop fall back to primary, before a predefined period has elapsed, to stop flapping on an unreliable primary link	9.08.29 10.00.23