

PingFailover Script Controlling an Interface Availability using Pings to Test Targets

Issue: 1.8

Date: 09 July 2014

1	Introduction	3
2	Configuring the pingFailover script.....	4
2.1	Script overview.....	4
2.2	Script requirements.....	5
2.3	Script parameters	5
2.4	Configuring the script	6
2.4.1	Pasting the script into the script editor	6
2.4.2	Scheduling the script to run on boot	7
3	Debugging commands	9
4	Script events	10
5	pingFailover script	11

1 Introduction

This document describes how to use the failover script pingFailover 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 for this link is disabled to allow other configured default routes to take priority. The script will optionally enable and disable a backup default route.

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 router 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 of an Ethernet interface to a third party gateway device such as ADSL, Satellite, WiFi and so on.

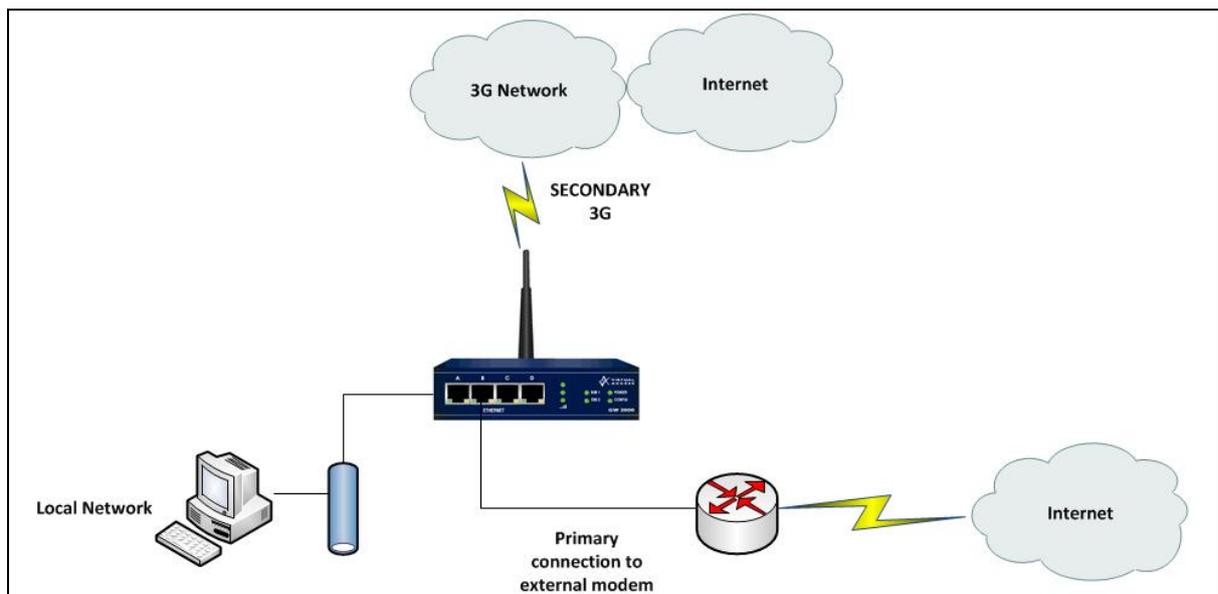


Figure 1: Example network architecture

2 Configuring the pingFailover 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:

- 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.
- Signifies using a ping from either target that router is operating as normal.
- When a number of configurable consecutive ping failures are detected:
 - the monitored interface default route is disabled to allow other default routes to take priority
 - a backup default route is optionally enabled
 - a backup interface is optionally manually connected
 - an INFO event is generated for visibility of the routing change and also to allow other scripts to fire where required.

Note: If two ping targets are configured pings must fail to both targets consecutively.

- The script optionally waits for a configurable period to avoid flapping on an unreliable primary connection.
- After the optional wait pings continue to be sent out the primary route to allow fall back. When a number of consecutive pings are successful:
 - the monitored interface default route is enabled
 - a backup default route is optionally disabled
 - a backup interface is optionally manually disconnected
 - an INFO event is generated for visibility of routing change and also to allow other scripts to fire where required.

The script optionally configures a static routes and filters for the test pings:

- a numbered static route at index 39 to the primary ping target out the monitored interface.
- a numbered static route at index 40 to the second ping target out monitored interface, if second ping target is configured.

- an advanced pass filter at index 37 to pass test pings to the first ping target out the monitored interface.
- an advanced pass filter at index 38 to pass test pings to the second ping target out the monitored interface, if second ping target is configured.
- an advanced block filter at index 39 to block test ping to the first ping target out any other WAN interface.
- an advanced block filter at index 40 to block test ping to the second ping target out any other WAN interface, if the second ping target is configured

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 9.08.85 and greater.
- Primary and backup default routes must be configured. If the script is not automatically enabling and disabling a backup default route then the backup default route must be enabled in the configuration; otherwise it can be configured and left disabled.
- If not allowing the script to automatically configure static routes and filters, your configuration must ensure pings are sent out the monitored interface when the monitored interface default route is disabled.

2.3 Script parameters

The script name is pingFailover and it takes in three required parameters and a further ten optional parameters:

```
pingFailover    [test-address1] [test-address2] [prim rt index] [sec rt index] [forceBackup]
                [forceBacupIf] [initwait] [ping failures] [ping wait] [createRouteandFilters]
                [primaryIf] [ping reply wait] [fallback guard time]
```

These parameters are described in the example and table below.

```
pingFailover 1.1.1.1, 2.2.2.2, 1, 0, 1, ppp-2, 60, 5, 2, 1, ppp-1, 2, 0
```

Parameter	Type	Description	Default
1.1.1.1	Required	The first ping target.	n/a
2.2.2.2	Required	The second ping target. Set to 0 to signify no second ping target.	n/a
1	Required	The default route index of the interface to be monitored.	n/a
0	Optional	The default route index of the backup interface to automatically enable or disable when the monitored interface is unavailable or available (0 for no automatic enable or disable)	0
1	Optional	Whether to force a manual connect or disconnect of the backup interface when the monitored interface is unavailable or available (0 for no manual connect)	1
Ppp-2	Optional	The backup interface to manually enable or disable when the monitored interface is unavailable or available	Ppp-2
60	Optional	The wait on boot before sending the first ping.	
5	Optional	The number of consecutive pings that signifies the monitored interface as available or unavailable.	5
2	Optional	The wait between pings in seconds. Pings are sent to both ping targets at the same time.	2
1	Optional	Whether to automatically configure static routes and filters for the pings. (0 to not automatically create)	1
ppp-1	Optional	The monitored interface the pings must go out.	Ppp-1
2	Optional	The time to wait for a ping reply in seconds (default: 2).	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 (def: 0 [0 means no wait])	0

Table 1: pingFailover parameter descriptions

2.4 Configuring the script

This script was introduced into firmware in versions 9.08.85. To use the script on older firmware versions first paste the script from Section 5: 'pingFailover 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, Expert View menu, **click system > management > scripts > script editor**. The Script Editor page appears.

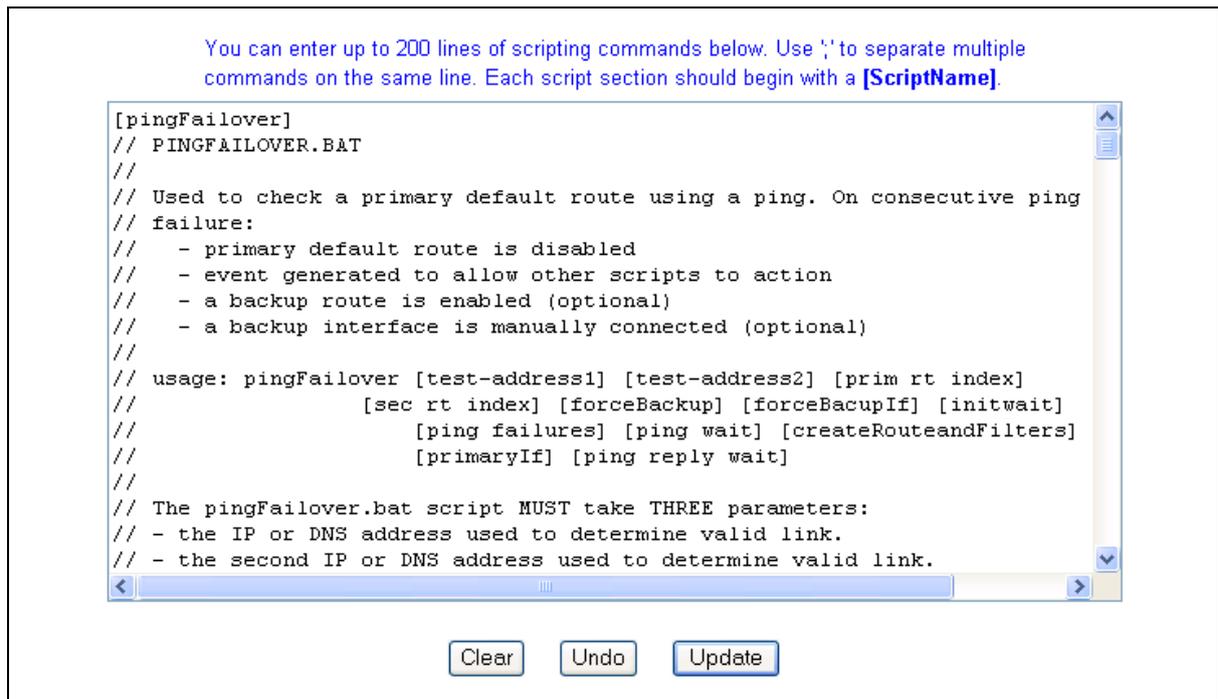


Figure 2: The script editor page

Paste in the script from Section 5 'pingFailover 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 9.09.xx firmware, in the Expert View menu, click **system > scheduler > scheduler tasks**. The Scheduler Task Entry page appears.

Click **add** in the Operation column of the list. The Scheduler Task form appears.

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

Click **add** in the Operation column of the list. The Scheduler Task form appears.

Scheduler Task Entry 1

Enabled

Name

Date (dd-mm-yyyy)

Time (hh:mm)

Frequency

Window secs

Script

Figure 3: The scheduler task entry page in version 9.09.xx

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: pingFailover 1.1.1.1, 2.2.2.2, 1

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 events	Shows event log
Dir scripts	Shows all scripts embedded in the firmware
Show pingFailover.bat	Displays pingFailover 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:icmp	Traces ICMP 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	pingFailover creating static route <route_index> next hop <nextHop>
INFO	49	40	pingFailover creating block filters <filter_index> for pings
INFO	49	40	pingFailover monitoring def rt <route_index> using <ping target IPs>
INFO	49	40	pingFailover primary default route down (index <route_index>)
INFO	49	40	pingFailover primary default route up (index <route_index>)
INFO	49	40	pingFailover no fallback for <fallback_guard_time>seconds

Table 5: Script events showing severity, class, subclass and text

5 pingFailover script

```
[pingFailover]
// PINGFAILOVER.BAT
//
// Used to check a primary default route using a ping. On consecutive ping
// failure:
// - primary default route is disabled
// - event generated to allow other scripts to action
// - a backup route is enabled (optional)
// - a backup interface is manually connected (optional)
//
// usage: pingFailover [test-address1] [test-address2] [prim rt index]
//           [sec rt index] [forceBackup] [forceBacupIf] [initwait]
//           [ping failures] [ping wait] [createRouteandFilters]
//           [primaryIf] [ping reply wait] [fallback guard time]
//
// The pingFailover.bat script MUST take THREE parameters:
// - the IP or DNS address used to determine valid link.
// - the second IP or DNS address used to determine valid link.
//   (zero or empty parameter for no second ping address)
// - the index of the primary route
//
// The script can then optionally take TEN more parameters
// - the secondary route to enable on ping failure (def: 0 or "" to do
nothing)
// - whether to manually connect the backup interface (def: 0 [means no])
// - the backup interface to connect (def: ppp-2)
// - the duration (secs) to wait until sending pings to test addresses
//   (to allow time for interface become active) [def: 60 secs]
// - the number of consecutive pings that signifies primary link down/up.
//   (def: 5)
// - the wait after pinging each destination before pinging again (def:: 2)
// - whether to create the static route and filters (def: 1) (0 to not
create)
// - the primary interface that the pings must go out pings (def: 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 REQUIREMENTS:
// - scheduler to invoke the script at startup
// - required default routes configured. The backup optionally disabled
(See above).
// - static route(s) to route test pings out primary link
// - block filter(s) to stop test pings going out backup link
// (Note: the script can configure the static routes and filters (see
above)
//
// EXAMPLES
// -----
// pingFailover 1.1.1.1, 2.2.2.2, 1, 2, 1, ppp-2, 5, 6, 10, 1, ppp-1
// (Delay 5 seconds after initial bootup before starting test ping. The
script will
// create static routes and filters for the script. Pings sent every 10
secs.
// On 6 consecutive ping failures the default route index 1 is disabled and
// index 2 is enabled and backup interface ppp-2 is manually connected
// After 6 consecutive ping successes default route index 2 is disabled and
// index 1 is enabled. Backup interface ppp-2 is disconnected.
//
!unique
!speed 50
!arg Paddress, Saddress, rtIndex
$brtIndex = $4
$forceBackup = $5
$forceBackupIf = $6
$initWait = $7
```

```
$Pcount = $8
$Pwait = $9
$createRouteandFilters = $10
$primaryIf = $11
$ping_reply_wait = $12
$fallback_wait= $13
$enable_backup = 1
$pingSaddress = 1

//checking optional params
!if Saddress = ""
    $pingSaddress = 0
!endif
!if Saddress = 0
    $pingSaddress = 0
!endif
!if brtIndex = ""
    $enable_backup = 0
!endif
!if brtIndex < 1
    $enable_backup = 0
!endif
!if forceBackup = ""
    $forceBackup = 0
!endif
!if forceBackupIf = ""
    $forceBackupIf = ppp-2
!endif
!if brtIndex > 4
    $enable_backup = 0
!endif
!if initWait = ""
    $initWait = 60
!endif
!if Pcount = ""
    $Pcount = 5
```

```
!endif
!if Pwait = ""
    $Pwait = 2
!endif
!if createRouteandFilters = ""
    $createRouteandFilters = 1
!endif
!if primaryIf = ""
    $primaryIf = 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

//create filters and static routes
!if createRouteandFilters <> 0

    //find default route info
    !if '`sh ip route default indexed numbered $rtIndex`' <> 'yes'
        $nextHop = `sh ip route default indexed next hop interface $rtindex`
        $z = `set ip route Static Numbered 39 no`
        $z = `set ip route Static Next Hop Interface 39 $nextHop`
        !if $pingSaddress <> 0
            $z = `set ip route Static Numbered 40 no`
```

```
$z = `set ip route Static Next Hop Interface 40 $nexthop`
endif
!else
$nexthop = `sh ip route default indexed next hop ip $rtindex`
$z = `set ip route Static Numbered 39 yes`
$z = `set ip route Static Next Hop IP 39 $nexthop`
!if $pingSaddress <> 0
$z = `set ip route Static Numbered 40 yes`
$z = `set ip route Static Next Hop IP 40 $nexthop`
endif
endif

$z = `set ip route static configured 39 yes`
$z = `set ip route static mask 39 255.255.255.255`
$z = `set ip route Static Metric 39, 1`
$z = `set ip route Static Target 39 $Paddress`

$z = `set filter ip 37 pass out on $primaryIf proto icmp from any to
$Paddress`

$z = `set filter ip 39 block out on wan proto icmp from any to $Paddress`

!if $pingSaddress <> 0
$z = `set ip route static configured 40 yes`
$z = `set ip route static mask 40 255.255.255.255`
$z = `set ip route Static Metric 40, 1`
$z = `set ip route Static Target 40 $Saddress`
$z = `set filter ip 38 pass out on $primaryIf proto icmp from any to
$Saddress`

$z = `set filter ip 40 block out on wan proto icmp from any to
$Saddress`
endif

//event static route and filter details
$filternum = 37 and 39
$staticroutenum = 39
!if $pingSaddress <> 0
```

```
$filternum = 37 to 40
$staticroutenum = 39 and 40
!endif

!log pingFailover creating static route $staticroutenum next hop $nexthop
!log pingFailover creating block filters $filternum for pings

$z = `commit`
$z = `set ip rt reconfigure`

!endif

//event ping details
$eventtxt = $pAddress
!if $pingSaddress <> 0
    $eventtxt = $Paddress and $Saddress
!endif

!log pingFailover monitoring def rt $rtindex using $eventtxt

//Checking default routes were misconfigured during a SAVE
!if ``sh IP Route Default Indexed Configured $rtIndex`` <> 'yes'
    $z = `set IP Route Default Indexed Configured $rtIndex yes`
    $z = `commit`
    $z = `set ip route reconfigure`
!endif

!if $enable_backup <> 0
    !if ``sh IP Route Default Indexed Configured $brtIndex`` <> 'no'
        $z = `set IP Route Default Indexed Configured $brtIndex no`
        $z = `commit`
        $z = `set ip route reconfigure`
    !endif
!endif

!pause $initWait
```

```
!label Primary_up
// Routing change so force management update of primary interface IP.
$z = `dl $$ip none`
$Pfail = 0
!while $Pfail < $Pcount
    $z = `st ping results reset`
    $z = `quiet ping $Paddress -w $ping_reply_wait_msec`
    !if $pingSaddress <> 0
        $z = `quiet ping $Saddress -w $ping_reply_wait_msec`
    !endif
    !pause $ping_reply_wait
    $result = `sh ping replies`
    !if $result > 0
        $Pfail = 0
    !else
        !inc Pfail
    !endif
    !pause $PWait
!endwhile

!label Primary_down
$Ppass = 0
$z = `set IP Route Default Indexed Configured $rtIndex no`
!if $enable_backup <> 0
    $z = `set IP Route Default Indexed Configured $brtIndex yes`
!endif
$z = `commit`
$z = `set ip route reconfigure`
!if $forceBackup > 0
    $z = `connect $forceBackupIf`
!endif
// Routing change so force management update of primary interface IP.
$z = `dl $$ip none`
!log pingFailover primary default route down (index $rtIndex)
!if $fallback_wait <> 0
    !log pingFailover no fallback for $fallback_wait seconds
```

```
    !pause $fallback_wait
!endif

!while $Ppass < $Pcount
    $z = `st ping results reset`
    $z = `quiet ping $Paddress -w $ping_reply_wait_msec`
    !if $pingSaddress <> 0
        $z = `quiet ping $Saddress -w $ping_reply_wait_msec`
    !endif
    !pause $ping_reply_wait
    $result = `sh ping replies`
    !if $result > 0
        !inc Ppass
    !else
        $Ppass = 0
    !endif
    !pause $PWait
!endwhile

//set primary route active
$z = `set IP Route Default Indexed Configured $rtIndex yes`
!if $enable_backup <> 0
    $z = `set IP Route Default Indexed Configured $brtIndex no`
!endif
$z = `commit`
$z = `set ip route reconfigure`
!if $forceBackup > 0
    $z = `disconnect $forceBackupIf`
!endif
!log pingFailover primary default route up (index $rtIndex)

!goto Primary_up
```

6 Script history

Version	Changes	Firmware version changes introduced
1.8	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