NAME

vpcs − Virtual PC Simulator

SYNOPSIS

vpcs [*options*] [*scriptfile*]

DESCRIPTION

The VPCS provides a command line interface to nine simulated virtual PCs. You can ping/trace route from/to them, or ping/trace route other hosts/routers from the virtual PCs, making it an ideal study tool when you simulate Cisco or Juniper routers in a Dynamips or GNS3 environment.

Virtual PCs are able to generate and respond to ICMP (ping), TCP and UDP packets delivered to the application via a UDP pipe or Unix tap interface. If *scriptfile* is specified, then vpcs reads the file on start-up and executes the commands in the scriptfile. *scriptfile* must be in **vpcs script file format**.

vpcs listens for messages on nine consecutive UDP ports and sends messages on nine consecutive UDP ports. By the default, vpcs listens on UDP ports 20000 to 20008 and sends messages on UDP ports 30000 to 30008. Each UDP port pair (20000/30000, 20001/30001...20008/30008) represents a virtual PC. Virtual PCs are numbered 1 to 9.

OPTIONS

-h, **--help**
: Print the command line options and exit

-v
: Print the version information and exit

[**-r**] *scriptfile*
: If *scriptfile* is specified, then vpcs reads the file on start-up and executes the commands in the *scriptfile*. *scriptfile* must be in vpcs script file format. By default, if a file named **startup.vpc** exists in the directory where the vpcs program was started, it will be read and executed when vpcs starts. The **-r** option is optional if *scriptfile* is the last parameter.

-p *port*
: Run *vpcs* as a daemon process listening on TCP port specified by *port*. As a daemon process, vpcs does not present a command line interface to the user, but the command line interface can be accessed remotely using a TCP stream application such as *telnet* or netcat (*nc*). Once the daemon has been started, there is no internal mechanism for terminating the program, and the program must be terminated by sending a system signal 9, typically by using the command **kill -9 PID** (where PID is the process id of the vpcs instance)

-m *num*
: *vpcs* uses 9 consecutive MAC addresses for the 9 vpcs stating at 00:50:79:66:68:00 by default. The **-m** option adds *num* to the last byte of the base MAC address. Should any increment cause the last byte exceed 0xFF during this process, it will increment to 0x00.

-e
: On systems that support the **/dev/tapx** interface (Unix/Linux), run vpcs in tap mode rather than UDP mode. In tap mode, IP packets are sent and received via /dev/tapx interfaces rather than via UPD streams. Typically /dev/tapx interfaces are only available to the root user, meaning vpcs would also be required to be run by the root user (**sudo vpcs -e**) to use tap mode.

[**-u**]
: This option is the default and not necessary, but included to contrast with the **-e** option. By default, vpcs sends and receives IP packets to and from specified UDP ports. vpcs listens on UDP port 20000 and sends to port 127.0.0.1:30000 by default. The listening and sending ports can be manipulated using the **-s**, **-c** and **-t** options.

UDP Mode Options

-s *port*
: *port* specifies the base port number that vpcs uses to listen for messages. By default *vpcs* listens for messages on UDP ports *20000* to *20008*. By changing the base port that vpcs listens to using the **-s** option causes nine consecutive UDP ports to be used starting at the port specified by *port*.

-t *ip*
: **vpcs** streams packets to nine UDP ports commencing at *127.0.0.1:30000* by default. The **-t** option allows you to stream packets to a remote host as specified by IPv4 address *ip*. Typically the remote host will be running dynamips with a cloud connection configured to link to this host's IP address.

-c *port* **vpcs** streams packets to nine UDP ports commencing at *127.0.0.1:30000*. The **-c** option allows you to stream packets to a different set of nine ports commencing at the base port number specified by *port*.

EXAMPLES

No command line options

If you start the *vpcs* with no arguments, *vpcs* will start and look for the script **startup.vpc** in the current directory. If it exists, it will run the script. This is the normal way of running the *vpcs*. It is simply envoked from the command line like this:

vpcs

Starting vpcs with an alternative startup file

To start *vpcs* with a startup script file called say **alternate.vpc**, use the file name as an argument to the **vpcs** command:

vpcs alternate.vpc

Running more than nine Virtual PCs

Suppose you needed more than nine Virtual PCs, so you want to run a second instance of vpcs on your local host. You would have to consider:

1. The VPCs MAC addresses for the second instance would need to be different,

2. The "local" or listening UDP port numbers for the second instance would have to differ from the first instance.

3. The remote UDP port numbers for the second instance would have to differ from the first instance.

Since the default local listening port is 20000, and the default remote port is 30000, you would want to start *vpcs* with a local listening port of 20009 (or greater) and remote port of 30009 (or greater) . You would also want the base MAC address to be offset by at least nine to avoid any clashes. In this case you would use the command:

vpcs -s 20009 -c 30009 -m 9

Running two instances of vpcs that can communicate with each other on the one host

Suppose you wanted to run a second instance of vpcs on your local host that can communicate with the instance already running with a default configuration. You would have to consider:

1. The VPCs MAC addresses for the second instance would need to be different,

2. The "local" or listening UDP port numbers for the second instance would have to match the "remote" port numbers of the first instance

3. The remote UDP port numbers would have to match the "local" or listening UDP port numbers of the first instance

Since the default local listening port is 20000, and the default remote port is 30000, you would want to start *vpcs* with a local listening port of 30000 and remote port of 20000. You would also want the base MAC address to be offset by at least nine to avoid any clashes. In this case you would use the command:

vpcs -s 30000 -c 20000 -m 9

INTERFACE

vpcs presents the user with a command line interface (unless daemon mode has been invoked by the **-p** option). The interface prompt indicates which of the 9 virtual PCs currently has focus by indicating the VPC number in brackets. Eg.:
VPCS[1]
Here the digit 1 inside the brackets indcates that VPC 1 has focus, and any traffic generated will be sent from VPC 1, and basic **show** commands will relate to VPC 1.

Basic commands supported are:

?	Print help
<digit>	Switch to the VPC<digit>. <digit> range 1 to 9
arp	Shortcut for: **show arp**. Show arp table
clear [arguments]	Clear IPv4/IPv6, arp/neighbor cache, command history
dhcp [-options]	Shortcut for: **ip dhcp**. Get IPv4 address via DHCP
echo <text>	Display <text> in output
help	Print help
history	Shortcut for: **show history**. List the command history
ip [arguments]	Configure VPC's IP settings
load <filename>	Load the configuration/script from the file <filename>
ping <host> [-options]	Ping the network <host> with ICMP (default) or TCP/UDP
quit	Quit program
rlogin [<ip>] <port>	Telnet to host relative to HOST PC
save <filename>	Save the configuration to the file <filename>
set [arguments]	Set VPC name, peer ports, dump options, echo on or off
show [arguments]	Print the information of VPCs (default). Try **show ?**
sleep <seconds> [text]	Print <text> and pause the running script for <seconds>
trace <host> [-options]	Print the path packets take to network <host>
version	Shortcut for: **show version**

vpcs script file format

Any text file consisting of valid vpcs commands can be used as a vpcs script file. Lines in the file beginning with the **#** character will be treated as comments and ignored. Command files can make use of the **echo** and **sleep** commands to create some form of interactive script.
Script file exececution can be aborted at any time by pressing Ctrl+c. This means that the **ping <host> -t** command (which must be terimated by Ctrl+c) is not useful in vpcs script files.

BUGS

IPv6 implementation is a basic implementation that is not fully implemented.

The **ping <host> -t** command (which must be terimated by Ctrl+c) can not be used in vpcs script files because when Ctrl+c is pressed to stop the ping, it also aborts the script file execution.

Please send problems, bugs, questions, desirable enhancements, patches etc to the author.

AUTHOR

Paul Meng <mirnshi[AT]gmail.com>
Documentation by Chris Welhs <rednectar.chris[AT]gmail.com>

COPYRIGHT