```
Cisco Router Simulation Platform (version 0.2.16-amd64/Linux stable)
Copyright (c) 2005-2011 Christophe Fillot.
Build date: Jun  8 2017 18:15:28

Usage: ./dynamips [options] <ios_image>

Available options:
  -H [<ip_address>:]<tcp_port> : Run in hypervisor mode

  -P <platform>      : Platform to emulate (7200, 3600, 2691, 3725, 3745, 2600
or 1700) (default: 7200)

  -l <log_file>      : Set logging file (default is dynamips_log.txt)
  -j                 : Disable the JIT compiler, very slow
  --idle-pc <pc>     : Set the idle PC (default: disabled)
  --timer-itv <val>  : Timer IRQ interval check (default: 1000)

  -i <instance>      : Set instance ID
  -r <ram_size>      : Set the virtual RAM size (default: 256 Mb)
  -o <rom_size>      : Set the virtual ROM size (default: 4 Mb)
  -n <nvram_size>    : Set the NVRAM size (default: 128 Kb)
  -c <conf_reg>      : Set the configuration register (default: 0x2102)
  -m <mac_addr>      : Set the MAC address of the chassis
                       (default: automatically generated)
  -C, --startup-config <file> : Import IOS configuration file into NVRAM
  --private-config <file> : Import IOS configuration file into NVRAM
  -X                 : Do not use a file to simulate RAM (faster)
  -G <ghost_file>    : Use a ghost file to simulate RAM
  -g <ghost_file>    : Generate a ghost RAM file
  --sparse-mem       : Use sparse memory
  -R <rom_file>      : Load an alternate ROM (default: embedded)
  -k <clock_div>     : Set the clock divisor (default: 4)

  -T <port>          : Console is on TCP <port>
  -U <si_desc>       : Console in on serial interface <si_desc>
                       (default is on the terminal)

  -A <port>          : AUX is on TCP <port>
  -B <si_desc>       : AUX is on serial interface <si_desc>
                       (default is no AUX port)

  --disk0 <size>     : Set PCMCIA ATA disk0: size (default: 64 Mb)
  --disk1 <size>     : Set PCMCIA ATA disk1: size (default: 0 Mb)

  --noctrl           : Disable ctrl+] monitor console
  --notelnetmsg      : Disable message when using tcp console/aux
  --filepid filename : Store dynamips pid in a file

  -t <npe_type>      : Select NPE type (default: "npe-400")
  -M <midplane>      : Select Midplane (default: "vxr", "std" or "vxr") (DM
false option, M is for ATM bridge!!)
  -p <pa_desc>       : Define a Port Adapter
  -s <pa_nio>        : Bind a Network IO interface to a Port Adapter
  -I <serialno>      : Set Processor Board Serial Number

  -a <cfg_file>      : Virtual ATM switch configuration file
  -f <cfg_file>      : Virtual Frame-Relay switch configuration file
  -E <cfg_file>      : Virtual Ethernet switch configuration file
  -b <cfg_file>      : Virtual bridge configuration file
  -e                 : Show network device list of the host machine

<si_desc> format:
   "device{:baudrate{:databits{:parity{:stopbits{:hwflow}}}}}"
```

```
<pa_desc> format:
   "slot:sub_slot:pa_driver" (DM why subslot???)

<pa_nio> format:
   "slot:port:netio_type{:netio_parameters}" (DM not explained???)

Available C7200 NPE drivers:
  * npe-100
  * npe-150
  * npe-175
  * npe-200
  * npe-225
  * npe-300
  * npe-400
  * npe-g1 (NOT WORKING)
  * npe-g2 (NOT WORKING)

Available C7200 Port Adapter (PA) drivers:
  * NPE-G2
  * C7200-IO-FE
  * C7200-IO-2FE (NOT WORKING)
  * C7200-IO-GE-E (NOT WORKING)
  * PA-FE-TX
  * PA-2FE-TX (NOT WORKING)
  * PA-GE (NOT WORKING)
  * PA-4E
  * PA-8E
  * PA-4T+
  * PA-8T
  * PA-A1
  * PA-POS-OC3
  * PA-4B (NOT WORKING)
  * PA-MC-8TE1 (NOT WORKING)
  * C7200-JC-PA

Available NETIO types:
  * unix       : UNIX local sockets
  * vde        : Virtual Distributed Ethernet / UML switch
  * tap        : Linux/FreeBSD TAP device
  * udp        : UDP sockets
  * udp_auto   : Auto UDP sockets
  * tcp_cli    : TCP client
  * tcp_ser    : TCP server
  * linux_eth  : Linux Ethernet device
  * gen_eth    : Generic Ethernet device (PCAP)
  * fifo       : FIFO (intra-hypervisor)
  * null       : Null device
```

```
Help for Cisco router simulator (dynamips-0.2.8)
================================================
Authors of this document: Fabien Devaux, Christophe Fillot, MtvE

Emulated hardware
*****************

The emulator currently supports the following platforms:
  - Cisco 7200 (NPE-100 to NPE-400)
  - Cisco 3600 (3620, 3640 and 3660)
  - Cisco 2691
  - Cisco 3725
```

```
  - Cisco 3745
  - Cisco 2600 (2610 to 2650XM)
  - Cisco 1700 (1710 to 1760)

By default, a Cisco 7206VXR with NPE-200 (256 Mb of DRAM) is emulated.

To emulate another platform, use the "-P" command line option (for example,
"-P 3725" or "-P 3600").

For the 7200, you can change the NPE type with the "-t" option.
It is possible to select "npe-100", "npe-150", "npe-175", "npe-200",
"npe-225", "npe-300" and "npe-400". The "npe-g1" is not working.

For the 3600, a 3640 with 128 Mb is emulated by default. You can change this
with the "-t" option and by specifying "3620" or "3660".
Don't forget to set the chassis type depending on your IOS image,
a c3660 image will not run on c3640 hardware and vice-versa.

Remark: PCMCIA card emulation is not supported yet with Cisco 3600.


Command Line Options overview
*****************************

  -l <log_file>      : Set logging file (default is dynamips_log.txt)
  -j                 : Disable the JIT compiler, very slow
  --exec-area <size> : Set the exec area size (default: 64 Mb)
  --idle-pc <pc>     : Set the idle PC (default: disabled)
  --timer-itv <val>  : Timer IRQ interval check (default: 1000)

  -i <instance>      : Set instance ID
  -r <ram_size>      : Set the virtual RAM size
  -o <rom_size>      : Set the virtual ROM size
  -n <nvram_size>    : Set the NVRAM size
  -c <conf_reg>      : Set the configuration register
  -m <mac_addr>      : Set the MAC address of the chassis
                       (default: automatically generated)
  -C <cfg_file>      : Import an IOS configuration file into NVRAM
  -X                 : Do not use a file to simulate RAM (faster)
  -R <rom_file>      : Load an alternate ROM (default: embedded)
  -k <clock_div>     : Set the clock divisor (default: 4)

  -T <port>          : Console is on TCP <port>
  -U <si_desc>       : Console in on serial interface <si_desc>
                       (default is on the terminal)

  -A <port>          : AUX is on TCP <port>
  -B <si_desc>       : AUX is on serial interface <si_desc>
                       (default is no AUX port)

  --disk0 <size>     : Set PCMCIA ATA disk0: size
  --disk1 <size>     : Set PCMCIA ATA disk1: size

  -a <cfg_file>      : Virtual ATM switch configuration file
  -f <cfg_file>      : Virtual Frame-Relay switch configuration file
  -E <cfg_file>      : Virtual Ethernet switch configuration file
  -b <cfg_file>      : Virtual bridge configuration file
  -e                 : Show network device list of the host machine

Options specific to the Cisco 7200 series:

  -t <npe_type>      : Select NPE type (default: "npe-200")
  -M <midplane>      : Select Midplane ("std" or "vxr")
  -p <pa_desc>       : Define a Port Adapter
```

```
  -s <pa_nio>        : Bind a Network IO interface to a Port Adapter

Options specific to the Cisco 3600 series ("dynamips -P 3600 --help"):

  -t <chassis_type>  : Select Chassis type (default: "3640")
  --iomem-size <val> : IO memory (in percents, default: 5)
  -p <nm_desc>       : Define a Network Module
  -s <nm_nio>        : Bind a Network IO interface to a Network Module

Options specific to the Cisco 2691 series ("dynamips -P 2691 --help"):

  --iomem-size <val> : IO memory (in percents, default: 5)
  -p <nm_desc>       : Define a Network Module
  -s <nm_nio>        : Bind a Network IO interface to a Network Module

Options specific to the Cisco 3725 series ("dynamips -P 3725 --help"):

  --iomem-size <val> : IO memory (in percents, default: 5)
  -p <nm_desc>       : Define a Network Module
  -s <nm_nio>        : Bind a Network IO interface to a Network Module

Options specific to the Cisco 3745 series ("dynamips -P 3745 --help"):

  --iomem-size <val> : IO memory (in percents, default: 5)
  -p <nm_desc>       : Define a Network Module
  -s <nm_nio>        : Bind a Network IO interface to a Network Module



Command Line Options details
****************************

-k <clock_div> :

   Specify the clock divider (integer) based on the host clock.
   Alter the value to match the CISCO clock with the real time.
   The command "show clock" at the IOS' CLI will help you set this value.

--idle-pc <pc> :

   The "idle PC" feature allows you to run a router instance without having
   a 100% CPU load. This implies that you can run a larger number of instances
   per real machine.

   To determine the "idle PC", start normally the emulator with your Cisco IOS
   image, and a totally IOS empty configuration (although not mandatory, this
   will give better results). When the image is fully booted, wait for the
   "Press RETURN to get started!" message prompt, but do not press Enter key.
   Wait about 5 seconds, then press "Ctrl-] + i". Some statistics will be
   gathered during 10 seconds. At the end, the emulator will display a list of
   possible values to pass to the "--idle-pc" option. You may have to try some
   values before finding the good one. To check if the idle PC value is good,
   just boot the Cisco IOS image, and check your CPU load when the console
   prompt is available. If it is low, you have found a good value, keep it
   preciously.

   Important remarks:
   ==================

      * An "idle PC" value is *specific* to a Cisco IOS image. You cannot
        boot a different IOS image without proceeding as described above.

      * Do not run the process while having the "autoconfiguration" prompt.
```

```
--exec_area <size> :

   The exec area is a pool of host memory used to store pages translated by
   the JIT (they contain the native code corresponding to MIPS code pages).



Cisco 7200 Port Adapter Description "<pa_desc>":
------------------------------------------------
   Format: slot:pa_driver

   slot: the number of the physical slot (starts from 0)

   pa_driver: the name of a Port Adapter driver in:
        - C7200-IO-FE  (FastEthernet, slot 0 only)
        - PA-FE-TX     (FastEthernet, slots 1 to 6)
        - PA-4E        (Ethernet, 4 ports)
        - PA-8E        (Ethernet, 8 ports)
        - PA-4T+       (Serial, 4 ports)
        - PA-8T        (Serial, 8 ports)
        - PA-A1        (ATM)


Cisco 3600 Network Module Description "<nm_desc>":
--------------------------------------------------
   Format: slot:nm_driver

   slot: the number of the physical slot (starts from 0)

   nm_driver: the name of a Network Module driver in:
        - NM-1E        (Ethernet, 1 port)
        - NM-4E        (Ethernet, 4 ports)
        - NM-1FE-TX    (FastEthernet, 1 port)
        - NM-4T        (Serial, 4 ports)
        - NM-16ESW     (Ethernet switch module, 16 ports)
        - Leopard-2FE  (Cisco 3660 FastEthernet in slot 0, automatically used)


Cisco 2691/3725/3745 Network Module Description "<nm_desc>":
------------------------------------------------------------
   Format: slot:nm_driver

   slot: the number of the physical slot (starts from 0)

   nm_driver: the name of a Network Module driver in:
        - NM-1FE-TX    (FastEthernet, 1 port)
        - NM-4T        (Serial, 4 ports)
        - NM-16ESW     (Ethernet switch module, 16 ports)
        - GT96100-FE   (2 integrated ports, automatically used)


NIO binding to Port Adapter "<pa_nio>" and Network Modules "<nm_nio>":
----------------------------------------------------------------------

   Format: slot:port:netio_type[:netio_parameters]

   slot       : the number of the physical slot (starts from 0)
   port       : the port in the specified slot (starts from 0)

   netio_type : host interface for communication

        unix:<local_sock>:<remote_sock>
           Use unix sockets for local communication.
```

```
            <local_sock> is created and represents the local NIC.
            <remote_sock> is the file used by the other interface.
            (ex. "/tmp/local:/tmp/remote")

         vde:<control_sock>:<local_sock>
            For use with UML (User-Mode-Linux) or VDE switches.
            VDE stands for "Virtual Distributed Ethernet".
            Please refer to : http://sourceforge.net/projects/vde/

         tap:<tap_name>
            Use a virtual ethernet device for communication.
            <tap_name> is the name of the tap device (ex. "tap0")

         gen_eth:<dev_name>
            Use a real ethernet device for communication, using libpcap 0.9
            or WinPcap. Works on Windows and Unix systems.

            <dev_name> is the name of the Ethernet device (ex. "eth0")

            The device list can be found using the "-e" option.

         linux_eth:<dev_name>
            Use a real ethernet device for communication (Linux specific).
            <dev_name> is the name of the Ethernet device (ex. "eth0")

         udp:<local_port>:<remote_host>:<remote_port>
            Use an UDP socket for connection between remote instances.
            <local_port> is the port we listen to.
            <remote_host> is the host listening the port you want to connect to.
            <remote_port> is the port you want to connect to.
          (ex. "1000:somehost:2000" and "2000:otherhost:1000" on the other
            side)

         tcp_cli:<host>:<port>
            Client side of a tcp connection.
            <host> is the ip address of the server.
            <port> is the port to connect to.

         tcp_ser:<port>
            Server side of a tcp connection.
            <port> is the port to listen to.

         null
            Dummy netio (used for testing/debugging), no parameters needed.


   VTTY binding to real serial port device "<si_desc>":
   ----------------------------------------------------

      Format: <device>{:baudrate{:databits{:parity{:stopbits{:hwflow}}}}}

      device: character device name, e.g. /dev/ttyS0
      baudrate: baudrate
      databits: number of databits.
      parity: data parity: N=none, O=odd, E=even,
      stopbits: number of stop bits
      hwflow: hardware flow control (0=disable, 1=enable)

      Note that the device field is mandatory, however other fields are optional.
      (dynamips will default to 9600, 8, N, 1, no hardware flow control)

      Note that access to the escape commands (described below) through a serial
      port are deliberately prevented, as the escape commands interfere with
```

serial encapsulation protocols.

Escape commands

You can press ^] (Ctrl +]) at any time, followed by one of these characters:

```
o : Show the VM object list
d : Show the device list
r : Dump MIPS CPU registers
t : Dump MIPS TLB entries
m : Dump the latest memory accesses
s : Suspend CPU emulation
u : Resume CPU emulation
q : Quit the emulator
b : Dump the instruction block tree
h : JIT hash table statistics
l : MTS64 cache statistics
c : Write IOS configuration to disk (ios_cfg.txt)
j : Non-JIT mode statistics
i : Determine an idling pointer counter
x : Experimentations (can crash the box!)
^]: Send ^]
```

If you press an unrecognized key, help will be shown.

Note: on Windows, it may be the "Ctrl + $" sequence.

Virtual Bridge

The virtual bridge is used to emulate a shared network between emulator instances.

Any emulator instance can act as a virtual bridge.

The configuration file (specified by the "-b" option) contains a list of NetIO descriptors, with the following syntax:

```
interface_name:netio_type[:netio_parameters]
```

Example:

```
# Connection to instance "I0"
I0:udp:10000:127.0.0.1:10001

# Connection to instance "I1"
I1:udp:10002:127.0.0.1:10003

# Connection to instance "I2"
I2:udp:10004:127.0.0.1:10005
```

The "I0" instance would be launched with the following parameters:

```
dynamips ios.bin -p 1:PA-FE-TX -s 1:0:udp:10001:127.0.0.1:10000
```

Virtual Ethernet switch

The virtual ethernet switch is used to emulate an Ethernet network between emulator instances. This switch supports access and trunk ports (802.1Q).

ISL will be available in a future release.

Any emulator instance can act as a virtual ethernet switch.

The configuration file (specified by the "-E" option) contains a list of NetIO descriptors (representing interfaces) and a list of interface properties (access/trunk port, VLAN info...)

The interface definition is similar to Port Adapters:

```
   IF:interface_name:netio_type[:netio_parameters]
```

1) Configuring an Access Port

syntax: ACCESS:interface_name:vlan_id

2) Configuration a 802.1Q Trunk Port

syntax: DOT1Q:interface_name:native_vlan

The native VLAN is not tagged. On Cisco devices, by default the native VLAN is VLAN 1.

Example of configuration file:

```
IF:E0:udp:10000:127.0.0.1:10001
IF:E1:udp:10002:127.0.0.1:10003
IF:E2:gen_eth:eth0

DOT1Q:E0:1
ACCESS:E1:4
DOT1Q:E2:1
```

Virtual ATM switch

The virtual ATM switch fabric is used to emulate an ATM backbone between emulator instances. The use of this virtual switch is not mandatory, you can directly connect emulator instances for point-to-point ATM connections. Please note that only basic VP/VC switching is supported, there is no support for ILMI/QSAAL/... or other specific ATM protocols.

Any emulator instance can act as a virtual ATM switch.

Example of configuration file (specified by the "-a" option):

```
# Virtual Interface List
IF:A0:udp:10001:127.0.0.1:10000
IF:A1:udp:10002:127.0.0.1:10003
IF:A2:udp:10004:127.0.0.1:10005

# VP connection between I0 and I1
VP:A0:10:A1:20
VP:A1:20:A0:10

# VP connection between I0 and I2
VP:A0:11:A2:30
VP:A2:30:A0:11

# VC connection between I1 and I2
VC:A1:5:2:A2:7:3
VC:A2:7:3:A1:5:2
```

In this example, we have 3 virtual interfaces, A0, A1 and A2. The syntax
for interface definition is similar to Port Adapters:

```
   IF:interface_name:netio_type[:netio_parameters]
```

You can do VP switching or VC switching:

1) VP switching

syntax: VP:input_if:input_vpi:output_if:output_vpi

2) VC switching

syntax: VC:input_if:input_vpi:input_vci:output_if:output_vpi:output_vci

Testing the Virtual ATM switch with one dynamips instance

(Contribution of Mtv Europe)

Virtual ATM switch configuration file ("atm.cfg"):

```
IF:A0:udp:10003:127.0.0.1:10001
IF:A1:udp:10004:127.0.0.1:10002
# a0/vpi=1/vci=100 connects to a1/vpi=2/vci=200
VC:A0:1:100:A1:2:200
VC:A1:2:200:A0:1:100
```

Invoking dynamips:

```
./dynamips -p 1:PA-A1 -s 1:0:udp:10001:127.0.0.1:10003 \
           -p 2:PA-A1 -s 2:0:udp:10002:127.0.0.1:10004 \
           -a atm.cfg IOS.BIN
```

(note input ports of IOS interfaces are output ports of ATM switch
interfaces, and vice versa).

IOS configuration:

```
ip cef
ip vrf test
 rd 1:1
 route-target both 1:1
int a1/0
 no shut
int a1/0.2 p
 ip addr 1.1.1.1 255.255.255.0
 pvc 1/100
interface a2/0
 no shut
interface a2/0.2 p
 ip vrf forwarding test
 ip addr 1.1.1.2 255.255.255.0
 pvc 2/200
!

# ping 1.1.1.2
!!!!!
```

Virtual Frame-Relay switch

The virtual Frame-Relay switch fabric is used to emulate a Frame-Relay

backbone between emulator instances. The use of this virtual switch is not mandatory, you can directly connect emulator instances with appropriate IOS configuration.

Any emulator instance can act as a virtual Frame-Relay switch.

There is only a basic implementation of the LMI protocol (ANSI Annex D), which is probably not conforming but works with Cisco IOS. Fortunately, Cisco IOS is able to detect automatically the LMI protocol.

Example of configuration file (specified by the "-f" option):

```
# Virtual Interface List
IF:S0:udp:10001:127.0.0.1:10000
IF:S1:udp:10002:127.0.0.1:10003

# DLCI switching between S0 and S1
VC:S0:200:S1:100
VC:S1:100:S0:200
```

In this example, we have 2 virtual interfaces, S0 and S1. The syntax for interface definition is similar to Port Adapters:

```
   IF:interface_name:netio_type[:netio_parameters]
```

DLCI switching syntax:

```
   VC:input_if:input_dlci:output_if:output_dlci
```

In the example above, the switch is configured to switch packets received on interface S0 with DLCI 200 to interface S1 with DLCI 100, and vice-versa.

== EOF ==