NAME

VsysTun − Click element; Uses planetlab's /vsys/fd_tuntap.control to allocate a tun device. Uses the specified file to bring up the device on a planetlab node.

SYNOPSIS

VsysTun(ADDR/MASK, VSYS_FILE [, *keywords* HEADROOM, MTU, IGNORE_QUEUE_OVERFLOWS])

Ports: at most 1 input, 1-2 outputs
Drivers: userlevel
Package: local (core)

DESCRIPTION

Reads IP packets from and writes IP packets to a tun device reservered through vsys.

Opens a UNIX socket to /vsys/fd_tuntap.control and calls recvmsg. vsys will create a tun device and open it and return to click the network device name and a file descriptor. click will send the device name, the IP address and prefix length and "snat=1" to VSYS_FILE.in and read VSYS_FILE.out which is expected to return nothing. This will bring up the network device and set its IP and also enable source NAT.

To use it on planetlab, you need to have reserved your own private IP and vsys access to vif_up and fd_tuntap. After initialization, **VsysTun** behaves like *KernelTun*(n).

EXAMPLES

VsysTun(10.6.0.1/24, VSYS_FILE "/vsys/vif_up")

will set up a tun device with IP 10.6.0.1. An IP packet send to the device should have source IP 10.6.0.2.

Keyword arguments are

HEADROOM
: Integer. The number of bytes left empty before the packet data to leave room for additional encapsulation headers. Default is 28.

MTU
: Integer. The interface's MTU, not including any link headers. *KernelTun*(n) will refuse to send packets larger than the MTU. Default is 1500; not all operating systems allow MTU to be set.

IGNORE_QUEUE_OVERFLOWS
: Boolean. If true, don't print more than one error message when there are queue overflows error when sending/receiving packets to/from the tun device (e.g. there was an ENOBUFS error). Default is false.

NOTES

This is specfic for planetlab.

SEE ALSO

KernelTun(n)