

**NAME**

**nos-tun** - implement “nos” or “ka9q” style IP over IP tunnel

**SYNOPSIS**

**nos-tun -t** *tunnel* **-s** *source* **-d** *destination* **-p** *protocol\_number* [*source*] *target*

**DESCRIPTION**

The **nos-tun** utility is used to establish an *nos* style tunnel, (also known as *ka9q* or *IP-IP* tunnel) using a tun(4) kernel interface.

*Tunnel* is the name of the tunnel device */dev/tun0* for example.

*Source* and *destination* are the addresses used on the tunnel device. If you configure the tunnel against a cisco router, use a netmask of "255.255.255.252" on the cisco. This is because the tunnel is a point-to-point interface in the FreeBSD end, a concept cisco does not really implement.

*Protocol number* sets tunnel mode. Original KA9Q NOS uses 94 but many people use 4 on the worldwide backbone of ampr.org.

*Target* is the address of the remote tunnel device, this must match the source address set on the remote end.

**EXAMPLES**

This end, a FreeBSD box on address 192.168.59.34:

```
nos-tun -t /dev/tun0 -s 192.168.61.1 -d 192.168.61.2 192.168.56.45
```

Remote cisco on address 192.168.56.45:

```
interface tunnel 0
ip address 192.168.61.2 255.255.255.252
tunnel mode nos
tunnel destination 192.168.59.34
tunnel source 192.168.56.45
```

**HISTORY**

The **nos-tun** utility appeared in FreeBSD 3.0.

**AUTHORS**

Nickolay N. Dudorov <nnd@itfs.nsk.su> wrote the program, Poul-Henning Kamp

<*phk@FreeBSD.org*> wrote the man-page. Isao SEKI <*iseki@gongon.com*> added a new flag, IP protocol number.

**BUGS**

We do not allow for setting our source address for multihomed machines.