The terms TUN and TAP are common in computer networking terminology.

These are the virtual-network kernel devices.

TUN is the short of network TUNnel.

TAP is the short of network tap.

The TUN simulates a network layer device and it operates with layer 3 packets like IP packets.

TAP simulates a link layer device and it operates with layer 2 packets like Ethernet frames.

TUN is used with routing.

TAP is used for creating a network bridge.

---

在Ubuntu下，好像默认已经支持TUN / TAP了，最起码在Ubuntu 12.04 and 14.04上是这样的。

```
$ modprobe tun

$ sudo tunctl -u root -t tap0

$ ifconfig -a

eth0      Link encap:Ethernet  HWaddr 18:03:73:37:50:28

          inet addr:10.38.52.118  Bcast:10.38.53.255  Mask:255.255.254.0

          inet6 addr: fe80::1a03:73ff:fe37:5028/64 Scope:Link

          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1

          RX packets:146019 errors:0 dropped:0 overruns:0 frame:0

          TX packets:100320 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:125230837 (125.2 MB)  TX bytes:15958092 (15.9 MB)
          Interrupt:20 Memory:f7100000-f7120000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:5221 errors:0 dropped:0 overruns:0 frame:0
          TX packets:5221 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:508620 (508.6 KB)  TX bytes:508620 (508.6 KB)

tap0      Link encap:Ethernet  HWaddr aa:48:02:c7:a8:ae
          BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:500
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

$ ls -l /dev/net/tun
crw-rw-rwT 1 root root 10, 200  2月  5 08:52 /dev/net/tun
```

---

How to enable "eth0" in UNL ?

1. enbale tun / tap on host

2. uml_mconsole fedora config eth0=tuntap,,,192.168.0.254 (on host)

```
tap0      Link encap:Ethernet  HWaddr 96:b3:2f:7c:9c:fc

          inet addr:192.168.0.254  Bcast:192.168.0.255  Mask:255.255.255.255

          inet6 addr: fe80::94b3:2fff:fe7c:9cfc/64 Scope:Link

          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1

          RX packets:27 errors:0 dropped:0 overruns:0 frame:0

          TX packets:81 errors:0 dropped:0 overruns:0 carrier:0

          collisions:0 txqueuelen:500

          RX bytes:5396 (5.3 KB)  TX bytes:18529 (18.5 KB)
```

3. uml will show the following message when run the upper command on host

```
[root@localhost ~]# Choosing a random ethernet address for device eth0

Netdevice 0 (9e:0e:93:d7:a6:2a) :

TUN/TAP backend - IP = 192.168.0.254
```

4. # ifconfig -a (on UML)

```
eth0: flags=4098<BROADCAST,MULTICAST>  mtu 1500

        ether 9e:0e:93:d7:a6:2a  txqueuelen 1000  (Ethernet)

        RX packets 0  bytes 0 (0.0 B)

        RX errors 0  dropped 0  overruns 0  frame 0

        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
        device interrupt 5

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        loop  txqueuelen 0  (Local Loopback)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

5. enbale "eth0"for UML

```
# ifconfig eth0 192.168.0.253 up
* modprobe tun
* ifconfig tap0 192.168.0.254 netmask 255.255.255.255 up
* bash -c echo 1 > /proc/sys/net/ipv4/ip_forward
* route add -host 192.168.0.253 dev tap0
* bash -c echo 1 > /proc/sys/net/ipv4/conf/tap0/proxy_arp
# ifconfig -a
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.253  netmask 255.255.255.0  broadcast 192.168.0.255
        ether 9e:0e:93:d7:a6:2a  txqueuelen 1000  (Ethernet)
        RX packets 32  bytes 9678 (9.4 KiB)
        RX errors 0  dropped 13  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

        device interrupt 5


lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536

        inet 127.0.0.1  netmask 255.0.0.0

        loop  txqueuelen 0  (Local Loopback)

        RX packets 0  bytes 0 (0.0 B)

        RX errors 0  dropped 0  overruns 0  frame 0

        TX packets 0  bytes 0 (0.0 B)

        TX errors 0  dropped 0 overruns 0  carrier 0  collisions
```

---

Question:

当我在UML中设置"eth0"的ip address时

```
# ifconfig eth0 192.168.0.253 up
```

UML report the following error message

```
SIOCSIFFLAGS: Operation not permitted
```

Answer：

```
$ ls -l /dev/net/tun

crw-rw-rwT 1 root root 10, 200  2月  5 08:52 /dev/net/tun
```

在host上，/dev/net/tun device只允许root read/write,使用normal user运行，没有权限访问该

device。

临时解决方案：

```
$ sudo ./linux-4.3.4/linux ubda=Fedora21-AMD64-root_fs rw mem=512m umid=fedora
```

较理想的方案是使得/dev/net/tun device能被普通user access。