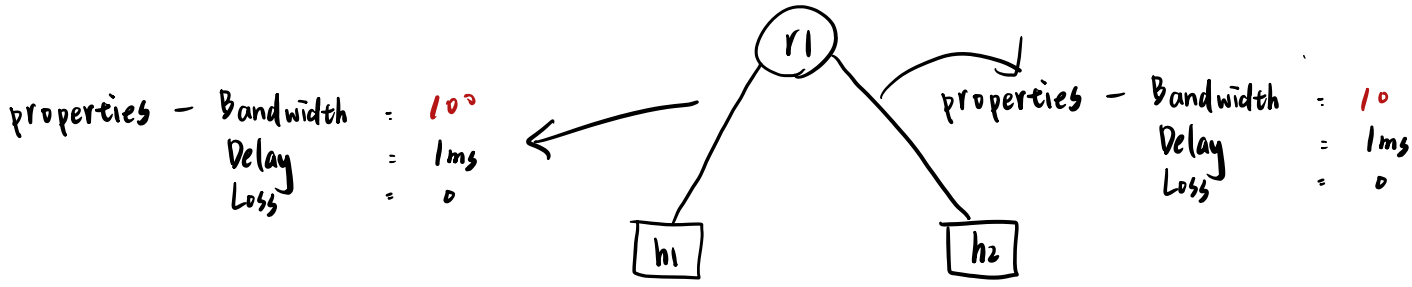
圖形化工具 = 快速建立網路的拓撲

miniedit

```
# cd /home/user/mininet/examples
# python miniedit.py
```

可以設定頻寬、遺失率、延遲



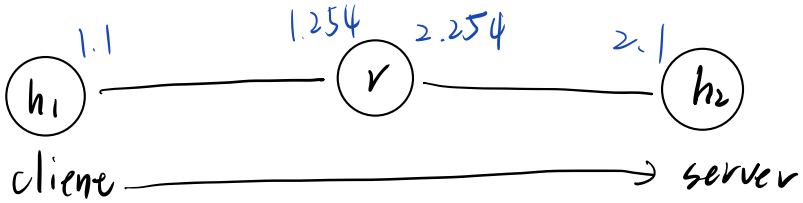
File - Export Level 2 Script - save

```
# gedit test1.py
```

軟件定義網路

```
h1r1 = {'bw':100,'delay':'1ms','loss':0}
```

頻寬100M，延遲1ms，遺失0




```
# python test.py
> xterm h1 h2
```

iperf (效能量測工具)

靜態

目的：量測效能比較，支援 tcp、udp 傳輸

```
# apt install iperf
```

tcp

```
h2                                  h1
# iperf -s -i 1                     # iperf -c 192.168.2.1 -t 10
```

server sec　client h2-ip　傳送時間

每秒吞吐量

UDP

h2
iperf -s -i 1 -u (udp)
port 5001

h2
iperf -s -i 1 -u -p 5555 (udp, port)

h1
iperf -c 192.168.2.1 -u -b 100M -t 100 (傳多快)
5001

h1
iperf -c 192.168.2.1 -u -b 200M -t 100 -p
5555

---

cp 3.py 3-1.py

gedit 3-1.py

```
hlr = {'bw':100,'delay':'1ms','loss':0}
net.addLink(h1, r, cls=TCLink, **hlr)
h2r = {'bw':100,'delay':'1ms','loss':0}
net.addLink(h2, r, cls=TCLink, **h2r)
```

python 3-1.py

> xterm h1 h2

h2
iperf -s -i 1

h1
iperf -c 192.168.2.1 -t 10

---

h2 | tee result -邊看 -邊存
iperf -s -i 1 > result 把結果存放在檔案裡

h1
iperf -c 192.168.2.1 -t 10

h2
cat result

留下時間、吞吐量 換掉sec 前10行 取代 取第4排、8排
cat result |grep "sec" |head -n 10 |tr "-" " " |awk '{print $4,$8}' > tcp_result

```
0       71
1       82
2       84
:
```

/mininet
cat tcp_result

```
0       71
1       82
2       84
:
```

```
# gnuplot
```
繪畫 tcp_result 用 點線表線 { lines / points

```
> plot "tcp_result" with linespoints
```

標示x軸為time

```
> set xlabel "time (sec)"
```

```
> replot 重畫
```

標示y軸為吞吐量

```
> set ylabel "throughput (Mbps)"
```

y軸最小為0，最大為100

```
> set yrange [0:100]
          min  Max
> set ytics 0,10,100
```

間距為10

```
> set xrange [0:15]
> set xtics 0,1,15
```

間距為1

```
> set title "TCP Flow Trough put"
> replot
```

變成圖檔

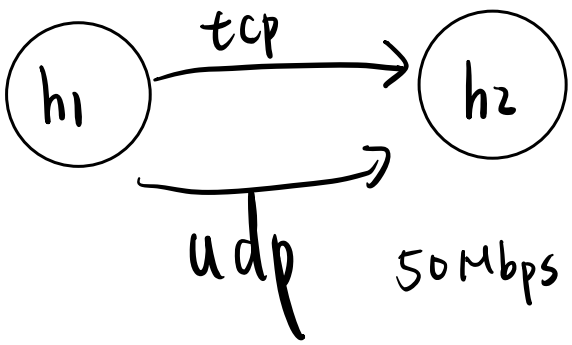
```
> set terminal gif
> set output "a.gif"
> replot
```

腳本

```
# gedit plot.plt
```

```
plot "tcp_result" with linespoints
replot
```

```
open tcp.gif
```



h2
iperf -s -i 1 -p 5555 |tee tcp

h1
iperf -c 192.168.2.1 -t 50 -p 5555

h2
iperf -s -i 1 -u -p 6666 |tee udp

h1
iperf -c 192.168.2.1 -u -b 50M -t 30 -p 6666

h2
cat tcp |grep "sec" | head -n 50 | tr "-" " " | awk '{print $4, $8}' > tcp_result

h2
cat udp |grep "sec" | grep -v out-of-order | head -n 30 | tr "-" " " |
awk '{print $4, $8}' > udp_result

排除

/mininet
gedit plot.plt

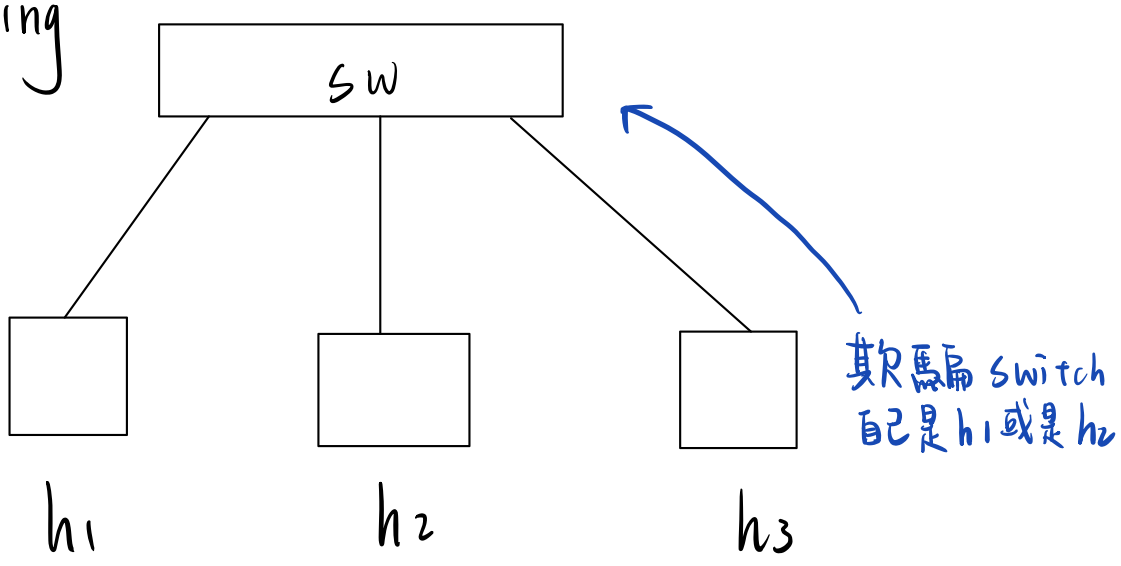
```
plot "tcp_result" with linespoints, "udp_result" with linespoints
replot
```

gunplot plot.plt
open tcp_udp.gif

---

port stealing

欺騙 sw



./4.py

> xterm h1 h2 h3 h3

h3
wireshark

h1
ping h2

h3 - wireshark 聽不到

ettercap -G

Ettercap

Hosts - Scan for hosts

Hosts - Hosts list - 192.168.10.1 ⇒ Add to Target 1 - ARP poisoning
192.168.10.2 ⇒ Add to Target 2

- ok

h3 - wireshark 聽到了

h1
arp -s 192.168.10.2 00:00:00:00:00:02

arp -n

h2
arp -s 192.168.10.1 00:00:00:00:00:01

arp -n

ping 192.168.10.1

Ettercap

port stealing - sniff remote connections

防範

SW可以綁定卡號 防止不明卡號

下載 和 ettercap

```
# wget  https
解 壓
# tar  vfxs  ettercap-0.8.3_tar.gz

# mkdir  build
# cd  build                    # cp  /usr/local/bin/ettercap     /usr/local/bi
#  cmake ../
#  make install
#  ettercap -G
```