作業：output.png 台灣地址

a.zip 解開pw密

編碼

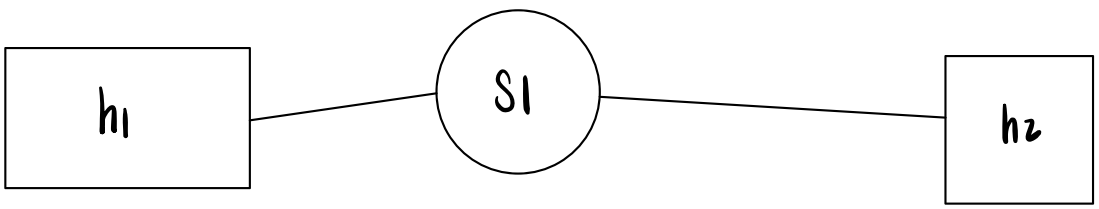
a.txt
b.txt
c.txt

difficult ↑ > 原本字串

⇒ easy

FFMPEG 實驗：影像傳輸、影像壓縮、串流、接收

安裝
apt install ffmpeg



encode

decode

stream 串流 ⇒ 影片 - 邊傳 - 邊播

3G ⇒ 多媒體，傳送速度：384 kbps，可以打視訊電話

格式

解析度

176 pixl

qcif

144

352

cif

288

抓 影檔

wget https ...

轉換格式　　格式

```
# ffmpeg -i foreman_cif.y4m foreman_cif.yuv
```

RGB三原色轉 亮度、色度 會比較好處理

壓縮

原始　指定格式

```
# ffmpeg -f rawvideo -s:v 352x288 -r 30 foreman_cif.yuv
  -c:v libx264 -qp 30 -g 12 -bf 2 -f mpeg foreman.mp4
```

壓縮格式　qp 1 20 10

壓縮比：數值越大，壓縮比越高，檔案越小，失真大
　　小　　低　　大　小

```
# ffmpeg -f rawvideo -s:v 352x288 -r 30 foreman_cif.yuv
  -c:v libx264 -qp 10 -g 12 -bf 2 -f mpeg foreman2.mp4
```

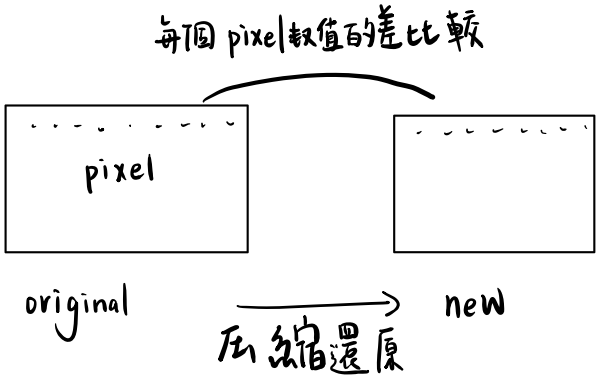
```
# ls -l foreman_cif.yuv -h
```
44M
```
# ls -l foreman.mp4 -h
```
qp 30

276k
```
# ls -l foreman2.mp4 -h
```
qp 10

8.8M

播放

```
# ffplay foreman.mp4
# ffplay foreman2.mp4
```



psnr值越大，失真越小
psnr值越小，失真越大

psnr值 30以上，不錯

```
# gcc -o psnr.c -lm
```

解壓縮

```
# ffmpeg -i foreman.mp4 1.yuv
# ffmpeg -i foreman2.mp4 2.yuv
# ls -l -h 1.yuv 2.yuv
```

44M
44M

壓縮率

```
# ./psnr 352 288 420 foreman_cif.yuv 1.yuv > psnr1
```

長 寬 原本的影像 qp 30 壓縮還原

```
# ./psnr 352 288 420 foreman_cif.yuv 2.yuv > psnr2
```

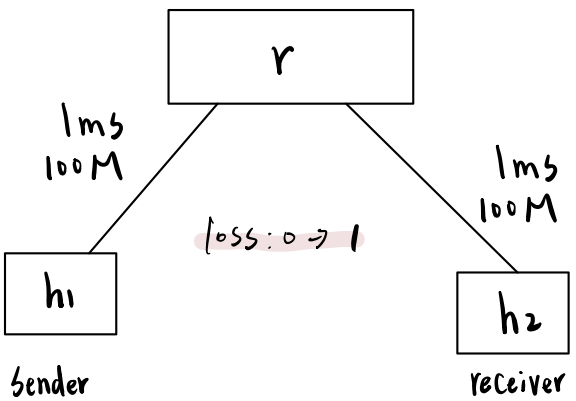
qp 10 壓縮還原

```
# gunplot
> plot "psnr1" with linespoints, 'psnr2" with linespoints
```

psnr1 ⇒ 值大約在 40-15，精度大

psnr2 ⇒ 值大約在 50-55，精度小

視訊串流




```
# gedit 3-1.py
```

h2r . . . . 'loss'=0
h2r . . . . 'loss'=1

```
# python 3-1.py
> xterm h1 h2
```

h2
ifconfig
192.168.2.1

ffplay -f udp://192.168.2.1:1234
壓縮完用 udp

h1
ffmpeg -re -i foreman.mp4 -c copy -f mpegts
udp://192.168.2.1:1234
送到↗
轉換成 mpegts，比較適合傳輸

loss rate 大，可以從影像上看出比較模糊
loss rate不同，对影像傳輸品質有影响

繪製成圖表

gedit 3-1.py
or h2r ..... 'loss'=0
or h2r ..... 'loss'=1
h2r ..... 'loss'=3

python 3-1.py
> xterm h1 h2

h2
ifconfig
192.168.2.1

ffmpeg -f udp://192.168.2.1:1234 -e copy 1-0.ts
1-1.ts
1-3.ts

h1
ffmpeg -re -i foreman.mp4 -c copy -f mpegts
udp://192.168.2.1:1234

h2
ffmpeg -i 1-0.ts 1-0.yuv
1-1.ts 1-1.yuv
1-3.ts 1-3.yuv

```
# ./psnr 352 288 420 foreman_cif.yuv 1-0.yuv> psnr1-0
                                    1-1.yuv> psnr1-1
                                    1-3.yuv> psnr1-3
```

```
#gunplot
  > plot "psnr1-0" with linespoints,"psnr1-1" with linespoints,
    ,"psnr1-3" with linespoints,

  > set xlabel "frames"
  > set ylabel "PSNR"
  > set title "foreman"
  > set key buttom
  > replot
```

---

rtp 即時傳輸協定

loss : 1

h2

h1 udp

作業：4/11 分別 qp=10，qp=20

Mininet 使用空間隔離，檔案沒有辦法分割

Mininet 擴充 ⇒ Containernet

到 Containernet 的環境

```
# cd /home/user/containernet
```

```
# python ./setup.py install
```

測試

```
# cd example
# python3 dockerhost.py
```

```
# cd /home/user/mininet-wifi
# util/install.sh -n
```

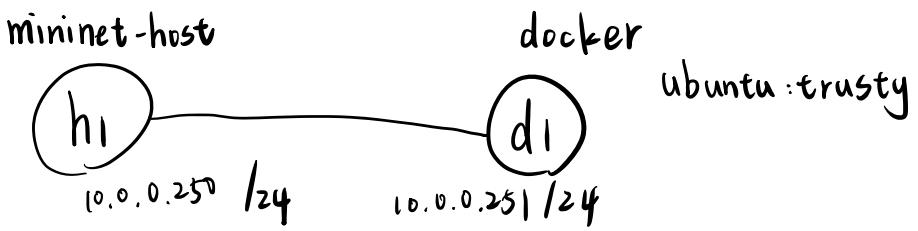
```
> net
> h1 ping d1 -c 3
```

---

docker 複習

```
# docker images
```
查看docker image

```
# docker pull httpd

# netstat tunple | grep 80

# docker run -itd -p 80:80 httpd:latest

# docker ps

# curl 127.0.0.1

# docker rm -f [ID]
```




```
#gedit test-docker.py
# python3 test-docker.py
```

```
> xterm h1
```

h1

```
# ifconfig
```

另外 d1

```
# docker ps
# docker exec -it mn.d1 bash
/# ifconfig
```