

## Measuring the performance of a Wired network

- note throughout the following tutorial the `courier` font indicates the terminal output
  - whereas the **`courier bold`** indicates keyboard input.
  - and *italic* indicates some thought is necessary in order to enter text

## Measuring the performance of a Wired network

- login to a GNU/Linux client
  - open up two terminal windows and arrange them side by side
- in the left hand window login to `j135-cluster.comp.glam.ac.uk` using `ssh`

## Measuring the performance of a Wired network

- do this by typing:

- ```
$ ssh j135-cluster.comp.glam.ac.uk
Password: yourpassword
```

## Measuring the performance of a Wired network

- from your browser download this: [performance monitoring tools](http://floppsie.comp.glam.ac.uk/download/notes/CompNets/performance.tar.gz) (`http://floppsie.comp.glam.ac.uk/download/notes/CompNets/performance.tar.gz`)
- and using the file manager move this downloaded archive into your home (or your top level) directory

## Measuring the performance of a Wired network

- unpack the archive by typing these commands into your right hand terminal window

```
$ tar xzf performance.tar.gz
$ cd performance
```

## Measuring the performance of a Wired network

- in the left hand window on j135-cluster.comp.glam.ac.uk type:

```
$ cd performance
```

- to run these performance tests you need to have changed to the performance directory

## Testing the performance of Wired Linux NFS

- in the client window (right hand window) type:

```
$ ./sendnfs
gcc -O2 -g -o bin/'arch'/send send.c
gcc -O2 -g -o bin/'arch'/rec rec.c
gcc -O2 -g -o bin/'arch'/sendnfs sendnfs.c
You should be on an NFS mounted directory - check mcgreg
Filesystem          1K-blocks      Used Available Use% M
etc etc etc
```

## Testing the performance of Wired Linux NFS

- the command `./sendnfs` writes many blocks across the network file system to the server and reports the speed
  - if the experiment runs too quickly or too slowly increase or decrease the value of the constant in the file: `sendnfs.c`
  - alter the line 5
  - `#define NBYTES (64*1024*10)`
  - to a number 10 times larger or smaller depending on the speed

## Testing the performance of Wired Linux NFS

- once you have made a change to the source code you need to type this command into the right hand window:

```
$ make clean
```

- when you get some interesting results you should capture the output of this test for your coursework

## Wired Linux TCP Performance

- in the left hand window on j135-cluster.comp.glam.ac.uk type:

```
$ ./tcprec
The receiving host is: j135-cluster.comp.glam.ac.uk, \
the port is 7000, the protocol is -tcp
```

- this confirms the host which will receive the tcp packets and it also tells you the tcp socket port number, in this case 7000

## Wired Linux TCP Performance

- now on the right hand window type:

```
$ ./tcp send j135-cluster.comp.glam.ac.uk 7000
```

- note the sender and receiver port number must match
  - in this case it was 7000 but yours might vary

## Wired Linux TCP Performance

- you should see output similar to this appear in your right hand window:

```
$ ./tcp send j135-cluster.comp.glam.ac.uk 7000
time to send    10 blocks of 67108864 bytes is 1.738 sec
time to send    20 blocks of 33554432 bytes is 1.736 sec
etc etc etc
```

- if you find the experiment taking too long then alter line 18 of the file send.c, which is:

```
#define NBYTES    65536*1024
```

- and divide this number by 8, or 16, (ensure that the divisor is a factor of 2)
- after modifying your test be sure to press <ctrl> C in both the right hand window and the left hand window
  - <ctrl> C terminates any test program
- also after any code modification type:

```
$ make clean
```

## Wired Linux TCP Performance

- in the right hand window
- when your test has finished be sure to press <ctrl> C in both the right hand window and the left hand window

## Wired Linux UDP Performance

- these performance results can be obtained by following the TCP performance instructions and altering every occurrence of *tcp* with *udp*
- the only significant difference is that UDP ignores errors and therefore the receiving side displays actual data received and the transmitter displays actual data sent