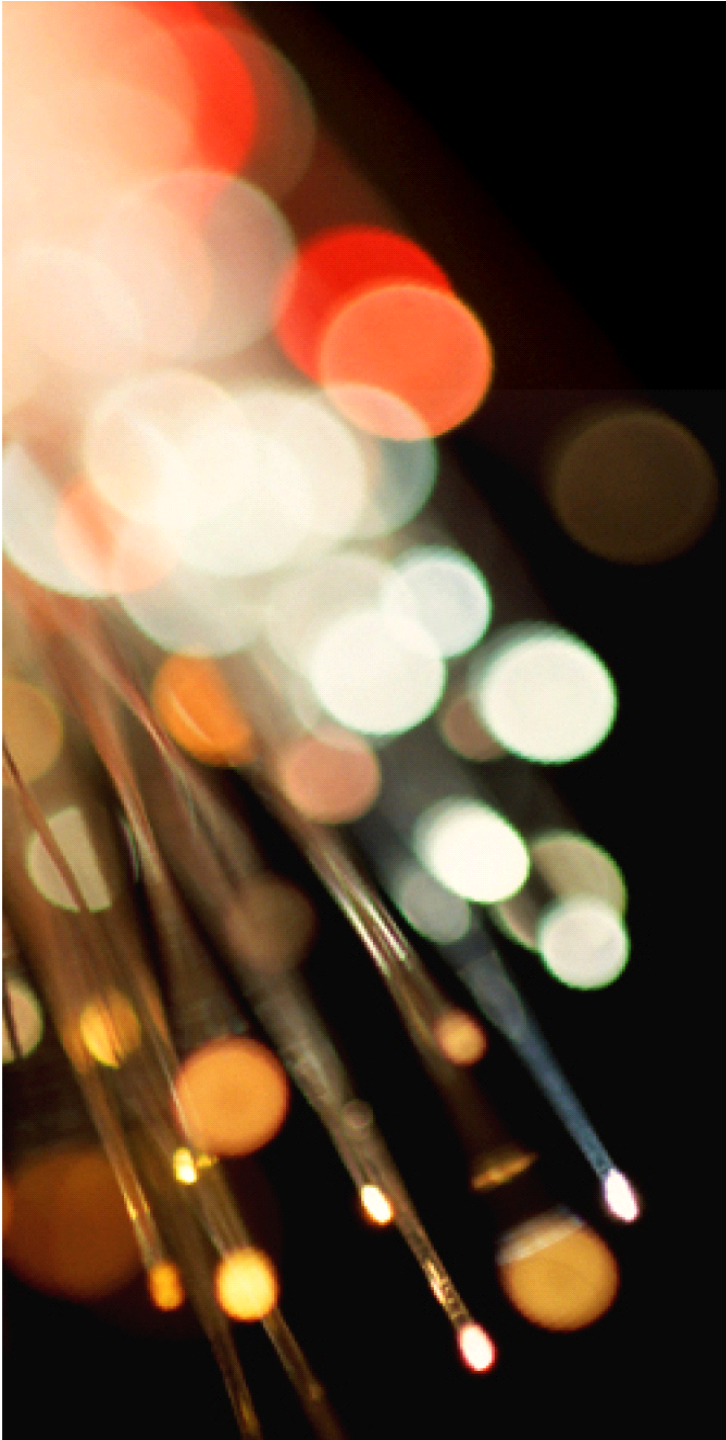
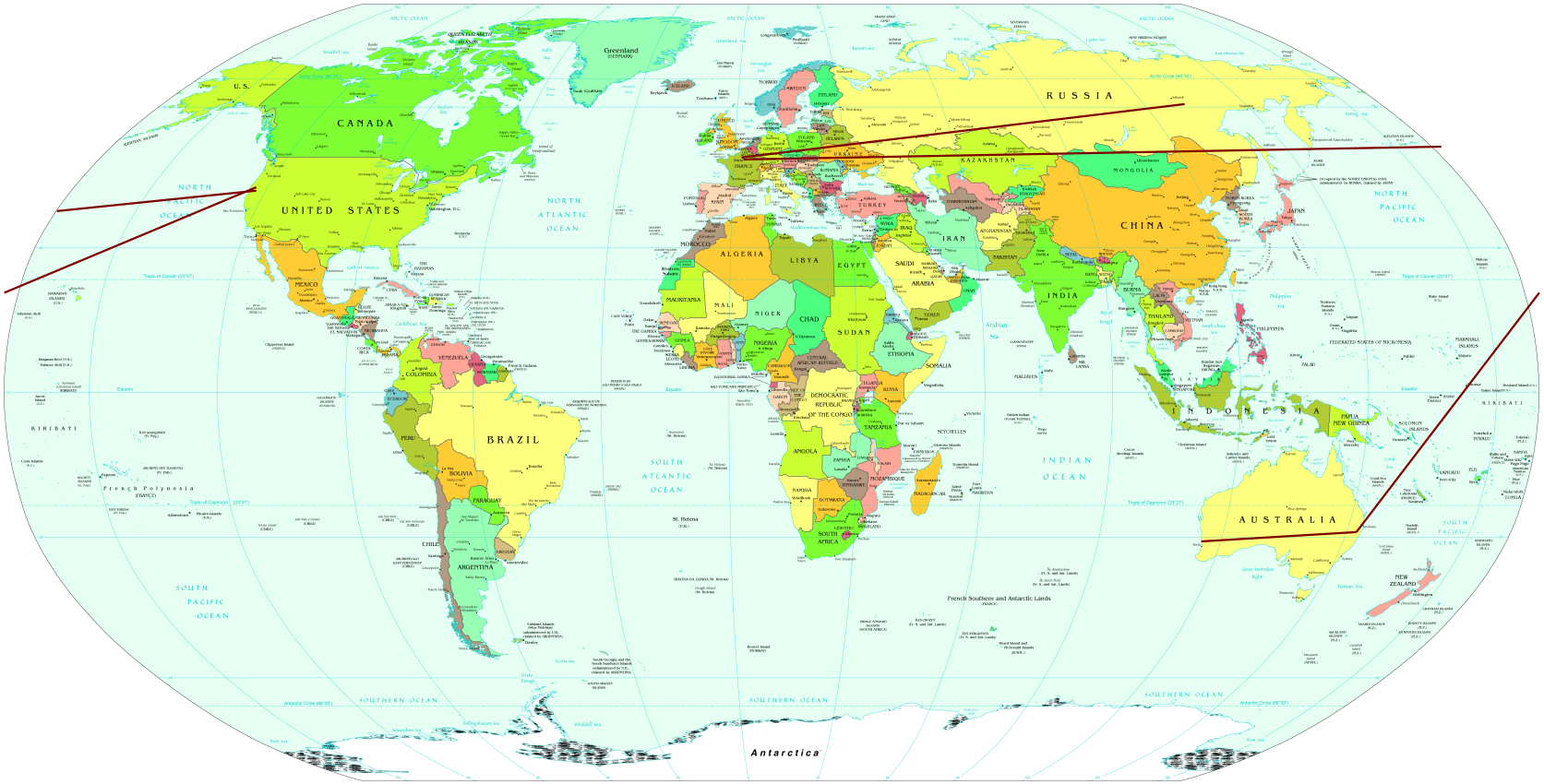


Applications and network performance

glen.turner@aarnet.edu.au

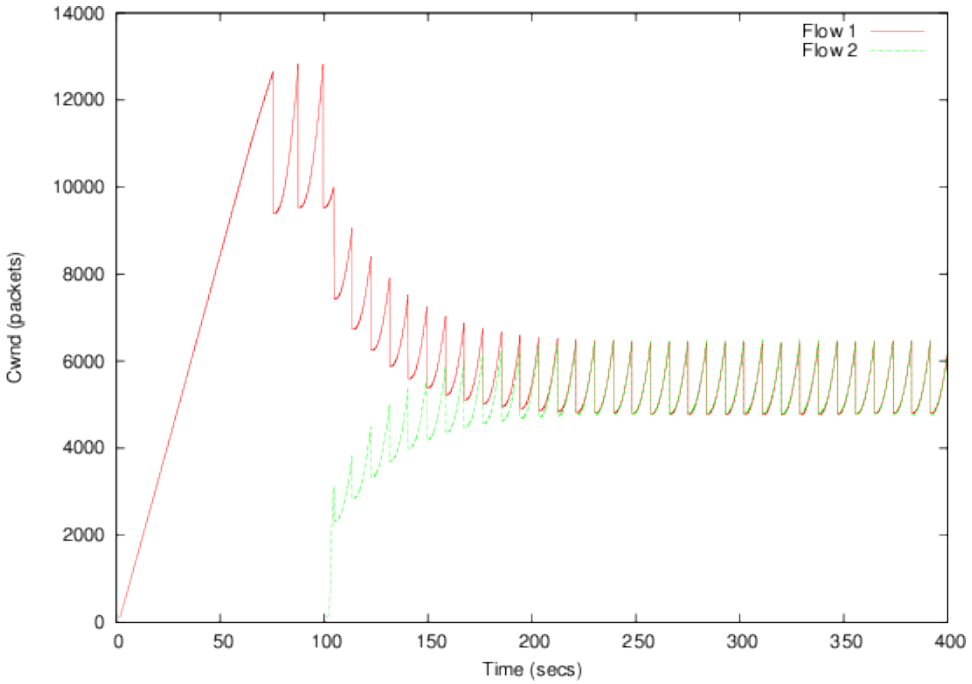
aarnet
Australia's Academic and Research Network

How long? 600ms



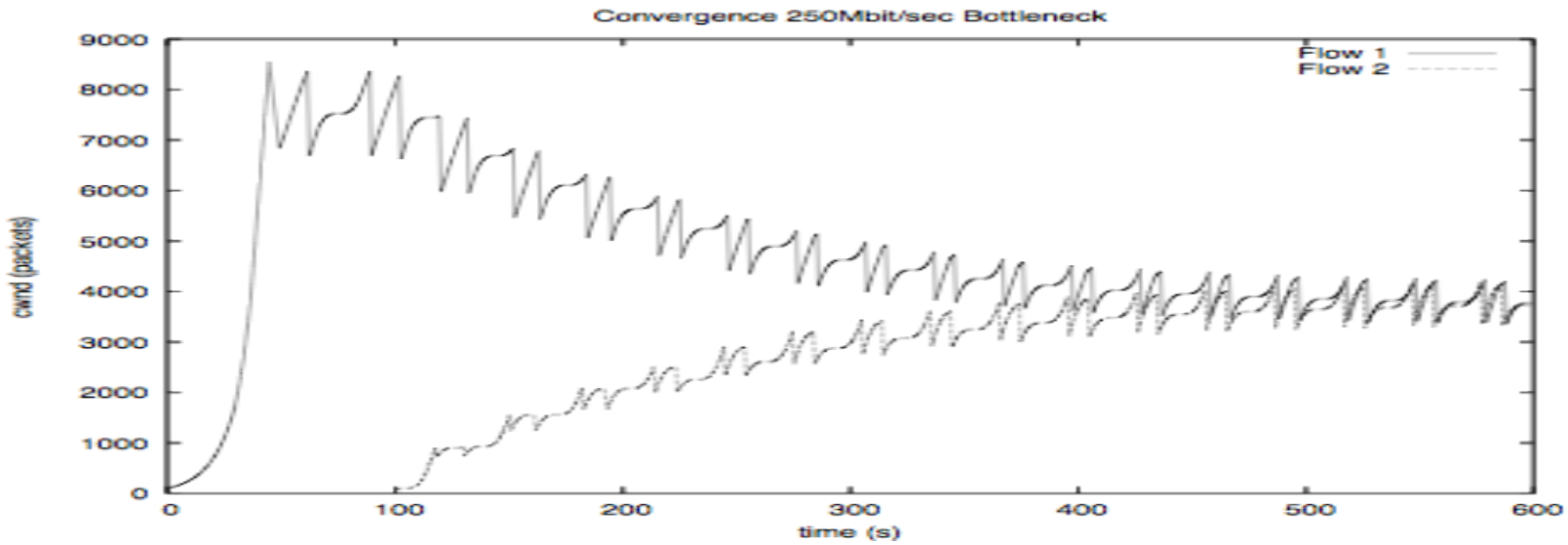
TCP basics

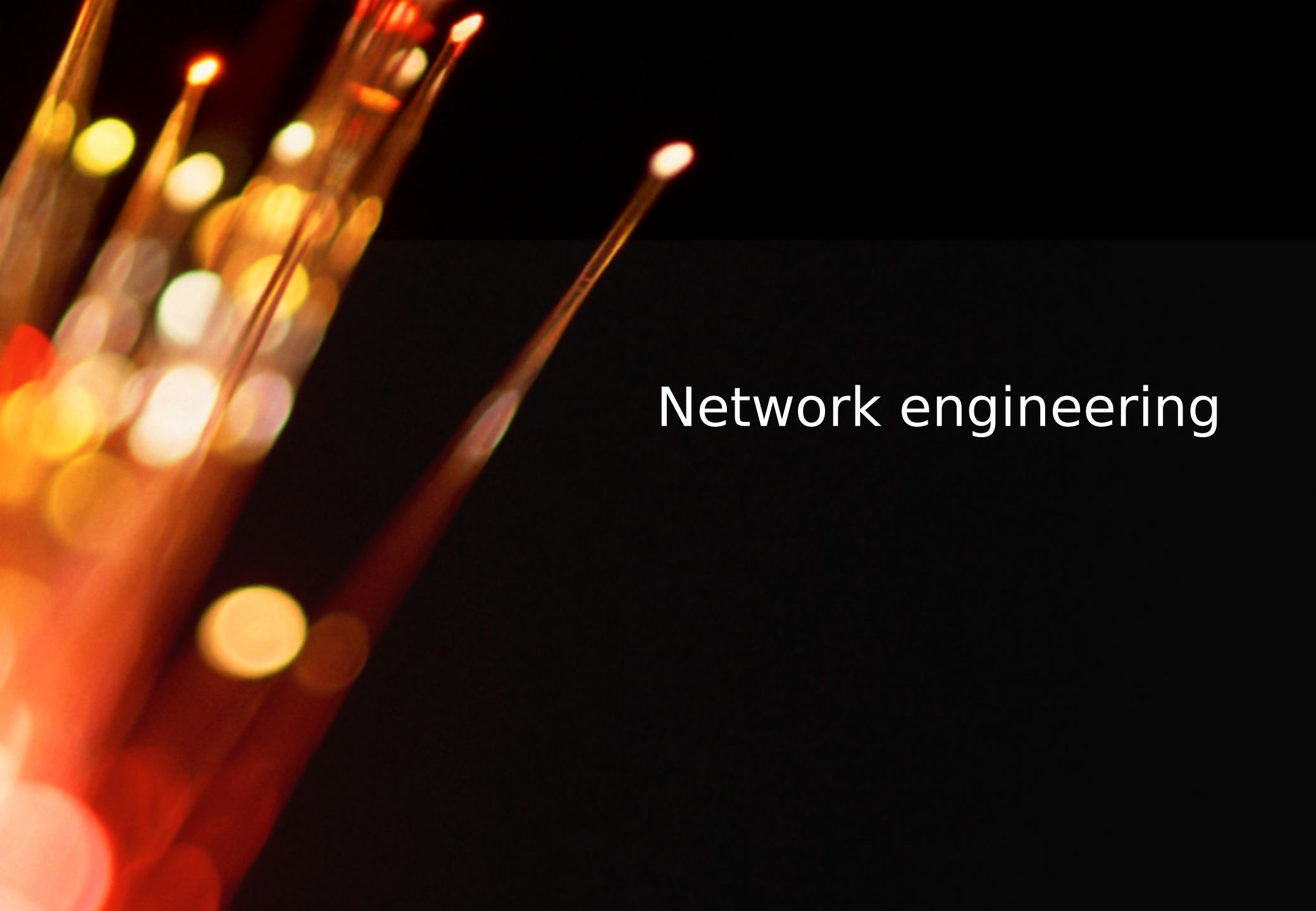
- Slow start
- Congestion avoidance



Inter-flow fairness

- Connections don't share bandwidth evenly
  - The size of the effect varies greatly by TCP algorithm
- So adding more connections may not improve the amount of data transferred
- Connections from further away always lose




Network engineering

Media bit error rate

- Looks like congestion to TCP
- If bad enough TCP never leads slow start

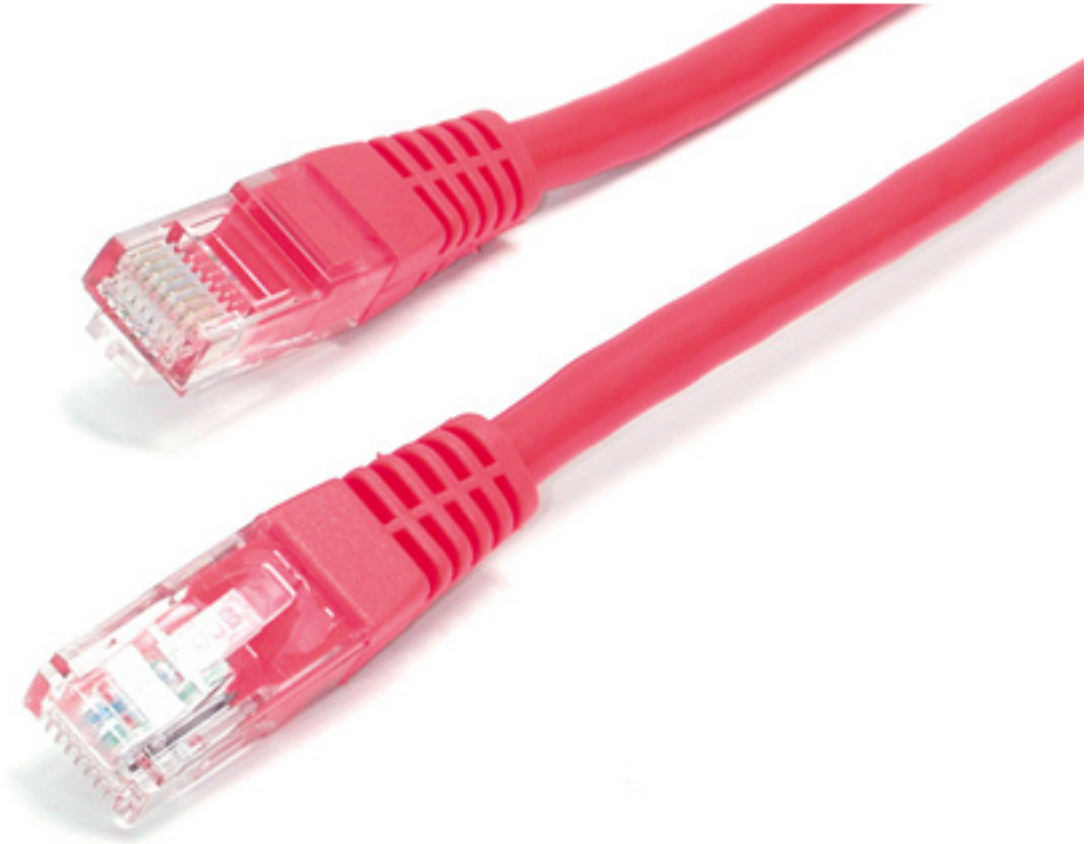
Pick the bottleneck

54Mbps WLAN with 3% loss

12Mbps ADSL with 0% loss

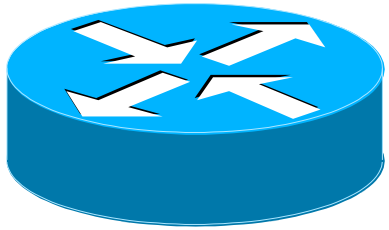
Configuration errors

- 90% fixed by not touching the speed and duplex knob



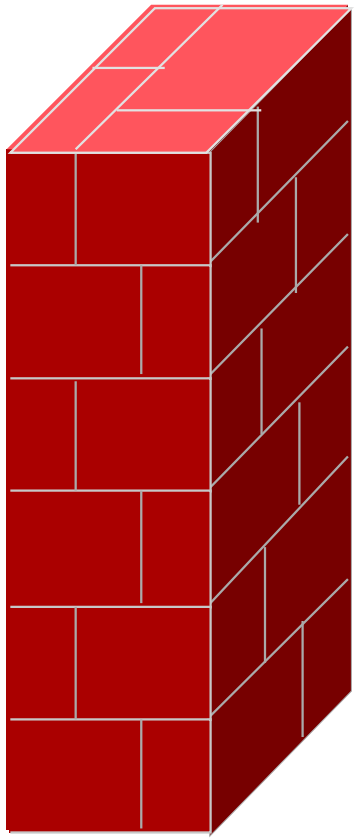
Routers are good

- Forwarding plane is in hardware
  - Low jitter
  - Accurate policing and shaping
  - Treating one flow doesn't effect other flows
- Big buffers
- Considered choice of queue algorithms
- Improving defaults



Except for one odd type of router

- These are usually PCs
  - Quantisation of output
    - Shaping leads to huge bursts upon clock ticks
  - Treating one flow adds jitter, or even loss, to another
- Small buffers
- Defaults are wrong
  - Queuing EF data in with BE data
  - FIFO rather than RED queues



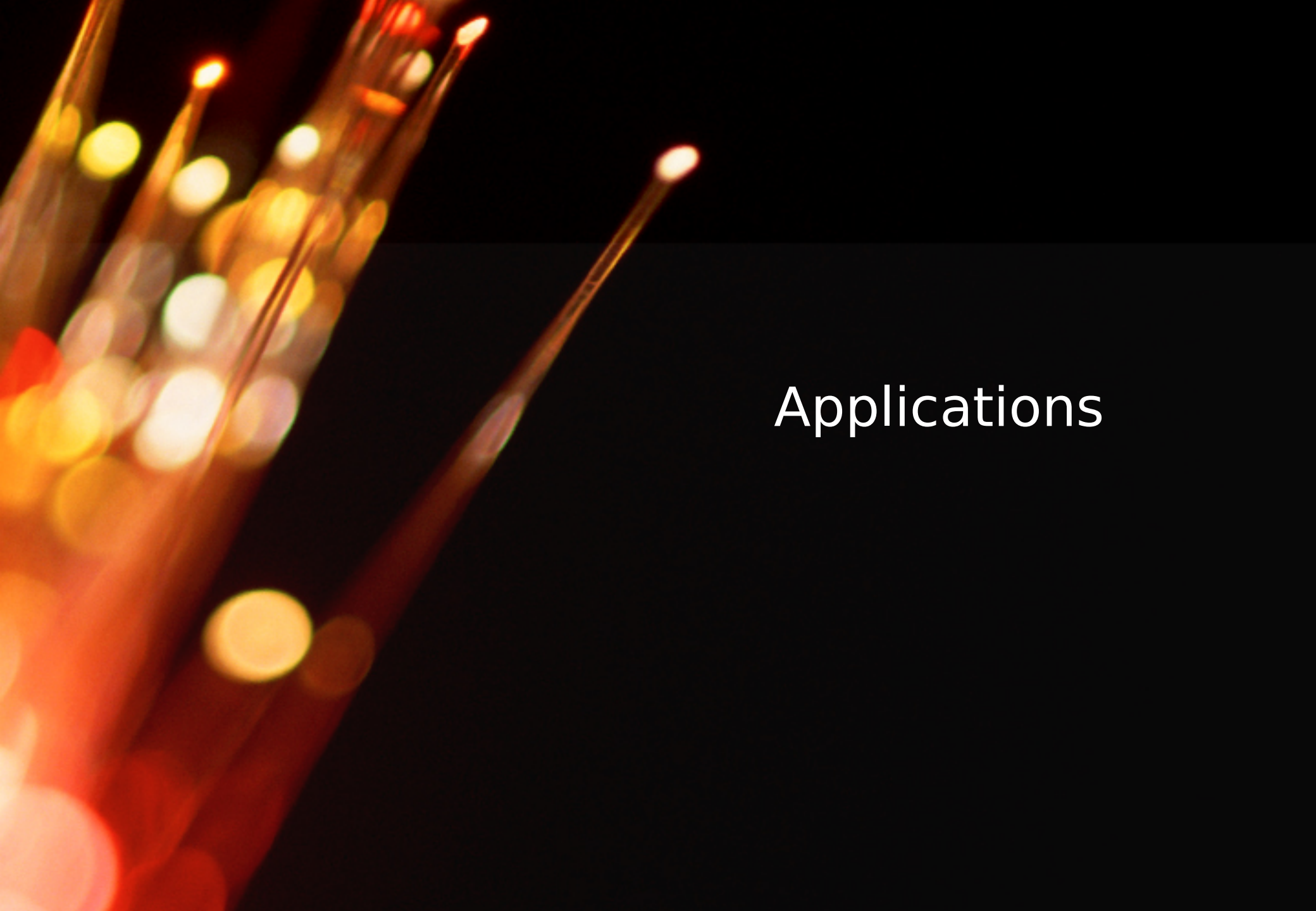
Operating systems are better

- Good TCP implementations
  - Modern TCP algorithm
- TCP buffer auto-tuning is improving
- 64-bit operating systems make more than 0.5GB available to networking
- This is a systems administration challenge
  - You want to run Fedora Core in x86-64 mode, maybe with the Web100 kernel patch
  - You want to run Windows Vista, it has no server version
  - They are the very last place sysadmins want a production service

Computers are cheap

- In the worst meaning of the term
  - Gigabit ethernet adapters not running at 1Gbps
  - Disk I/O buses that won't bus, working with only one drive
  - RAID hardware that runs in BIOS software
  - Interrupt controllers that drop interrupts
- By the time you benchmark all this, your test platform is outdated

As seen on TV!


Applications

Make round trip times count

- We can increase bandwidth but not the speed of light

Avoid application windows

- Application windows need to be larger than the bandwidth-delay product too.
- Writing one disk block or one window of disk blocks.
- Using a windowing mechanism

Loss of TCP estimates

- When your program is not sending data the TCP estimate is degrading
- When it degrades enough, slow start mode is entered
- ```
  while (running) {
    get_data(block);
    process_data(block);
  }
  ```
- Use a reader thread and a writer thread with a BDP-sized buffer between them.
  ```

What TCP wants to do

- Stream data continuously from disk to the network
- Operating systems optimise this to one system call

```
sendfile(file, socket)
```

- If you do something different expect to fix performance problems, perhaps expensively
  - SSL accelerators for HTTPS
  - TCP offload adapters for iSCSI

Caches

- If moving data a long way is hard, let's cache it close to the user
- Has the performance problem simply moved to populating the cache?
- The cache needs to be closer to the user, measured using milliseconds, not kilometres
  - Seattle is often closer than Indonesia in round-trip time milliseconds

Systems design and loading

- TCP sender makes the decisions
- TCP receiver does the work
- So where do I place the application's load?
  - HTTP chose the client, since the server would then be simple enough to serve huge numbers of people
  - A server with just one client should take the load

Is your application a DoS attack?

- Does it use UDP?
  - Does it send more data than it receives prior to authentication?
  - Is it congestion aware?
  - What happens when the user holds the mouse down
    - Hint: don't open fifty 10Mbps connections

Constant bit rate sources

- Data acquisition devices run a analogue-digital converter and pump out *n* samples per second whatever else is happening
- Use UDP with congestion detection rather than TCP
  - TCP can enter slow start, which may have lower bandwidth than the sampling rate
- Not dropping the occasional sample is very difficult, even the operating system won't play a CD without drops when under load

Avoid negotiations

- TCP is trying to discover the available bandwidth, why is our application sending small, atypical packets.
- Compare
  - Telnet
    - Will, won't, Will, won't, ...
  - HTTP
    - This is what I want, do your best

Avoid minor transactions

- FTP
  - AUTH GSSAPI 530
    AUTH KERBEROS 530
    USER anonymous 331
    PASS … 230
    CWD pub 200
    BIN 200
    RETR afile.bin 150
    QUIT 221
- These could have been a single chained transaction, as CIFS does

Move up the stack

- Compare the effectiveness of round-trip times between ODBC and SOAP and REST.
  - ODBC issues a single SQL statement
  - SOAP issues a single transaction
  - REST issues a part of a business function

Anticipate the entire transaction

- HTTP sends HTML, which is interpreted, which then requests style sheets and images.
- Style sheets can be provided with the HTML, so do that
  - Use a content management system so that this “hardcoding” is manageable
- A simple feature allows images to accompany the HTML text but it not used
  - This feature is used for HTML e-mail

API and programmers

- Why do the network APIs allow applications programmers to do the wrong thing. File I/O is buffered, why isn't network traffic.
- RPCs look like a function call. Chaining RPCs makes sense from a network layer, but breaks the abstraction presented to the programmer (chained function calls?). We need a new abstraction for the RPC mechanism, perhaps similar to parallel programming monitors.


aarnet
Australia's Academic
and Research Network