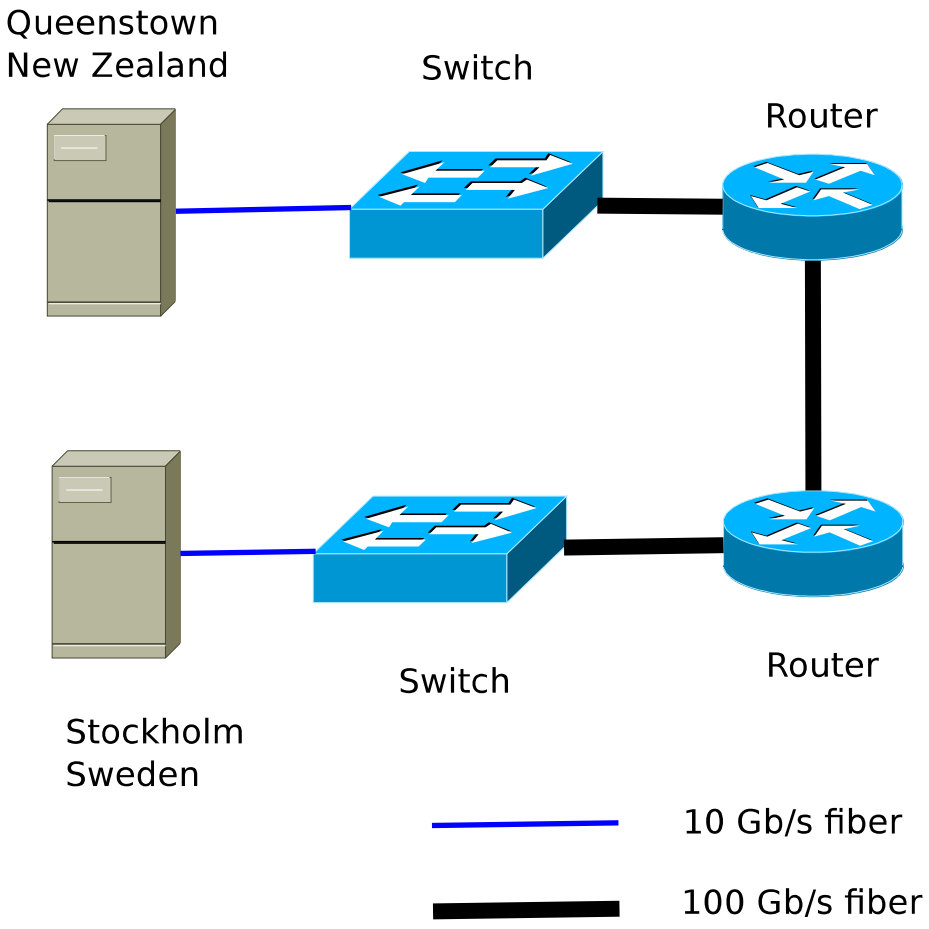
# Characterizing Performance and Fairness of Big Data Transfer Protocols on Long-haul Networks

*Nevil Brownlee <n.brownlee@auckland.ac.nz>*
Se-young Yu <syu051@aucklanduni.ac.nz>
Aniket Mahanti <a.mahanti@auckland.ac.nz>

# Transferring over Long Fat pipes

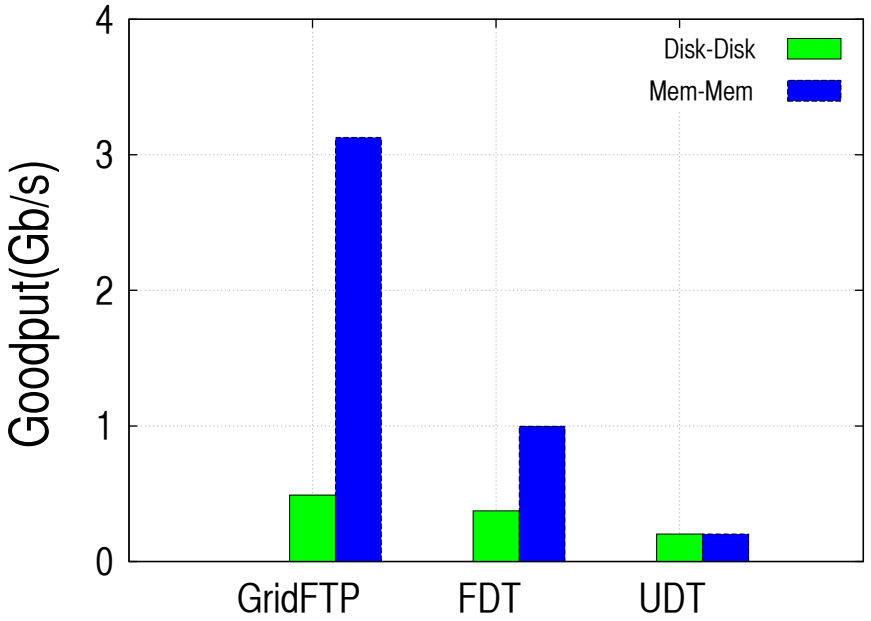
- SKA, LHC, ITER, etc., generate PB of data!
- Transferring data over distance is hard
- TCP is not efficient , UDP is not reliable
  - TCP uses small buffers by default
  - UDP doesn't provide congestion control
  - Neither protocol uses parallel transfer streams
- We tested systems that provide these improvements

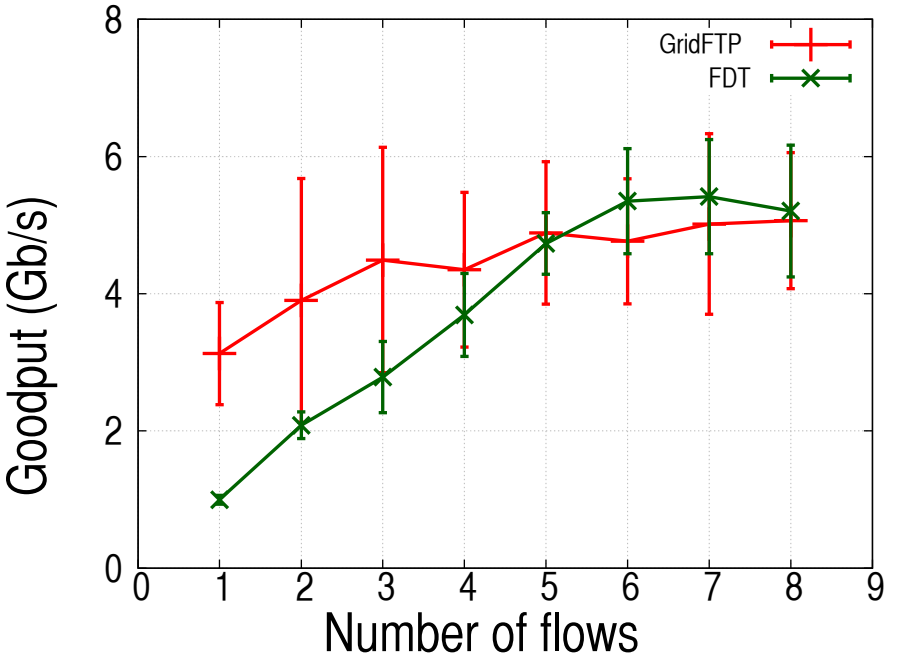
# Our 'International' Testbed




- 10 Gb/s, 320 ms RTT international link
- Tested GridFTP, FDT (TCP-based) and UDT (UDP-based)
- Measured goodput, RTT and bytes in-flight

# Disk vs Memory



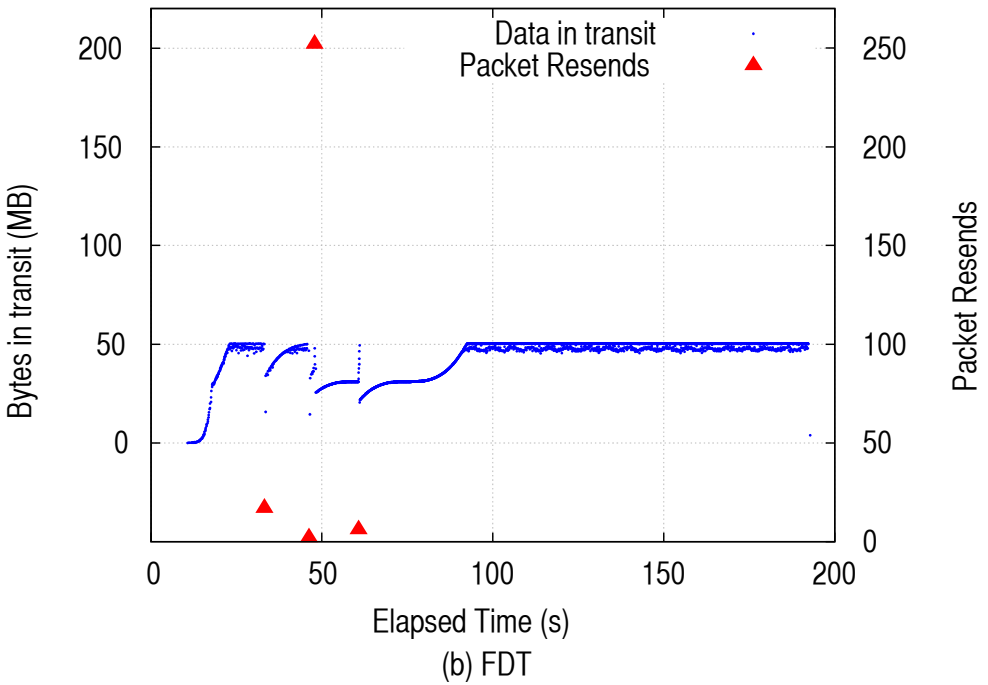




- Disk speed limits performance
- UDT does not improve mem-mem transfer rate

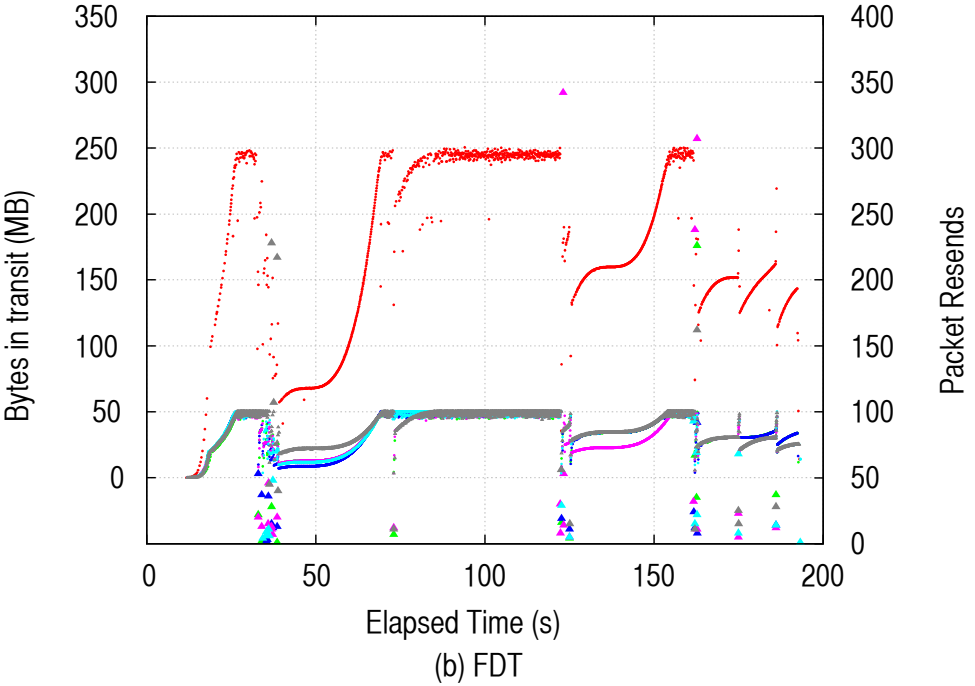
- Multiple flows improve performance for FDT (which uses TCP)

# What Happens in the Network

- Single Flow



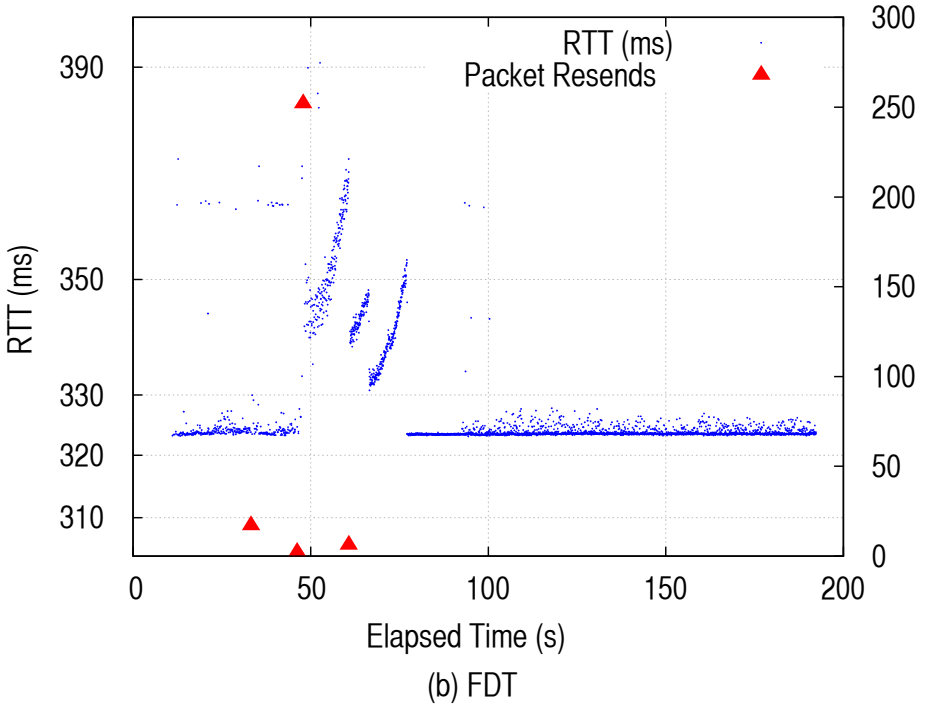

- Five Flows



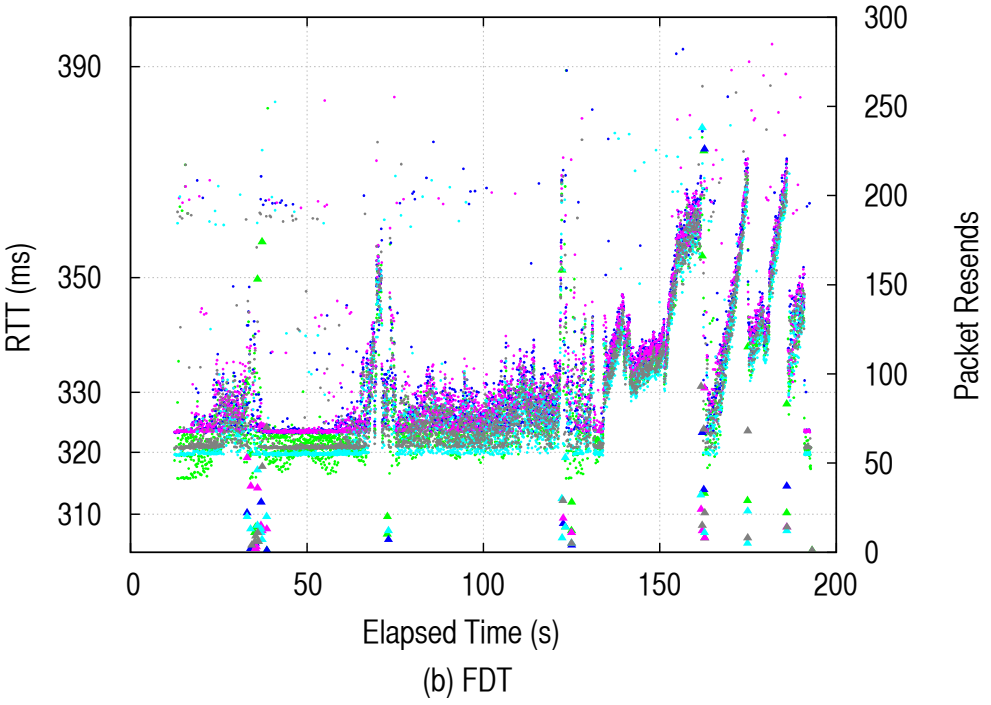

- Multiple flows increase aggregated *cwin*
- Faster recovery time with multiple flows

# Multiple Flows have Side Effects

- Single Flow




- 5 Flows




- Plots are for FDT, GridFTP plots are similar
- RTT increases for all flows
- More packets are lost

# Recommendation

- Avoid using UDP-based protocol
- Allocate enough socket buffer in OS/Kernel and force applications to use larger socket buffers
- Keep the number of multiple flows low

- Use TCP-based protocol with recent congestion control e.g. CUBIC, Scalable TCP
  - There are many possible definitions of network “fairness”
  - We want to use *most* of the available capacity
  - We consider our file transfers to be “fair”