

Pluggable TCP Congestion Avoidance Modules for eVLBI



Mark Kettenis, Arpad Szomoru
Joint Institute for VLBI in Europe
kettenis@jive.nl

Outline

- Introduction
- C-TCP
- Pluggable Congestion Avoidance Modules
- A simple module for eVLBI
- Measurements
- Conclusions

Datatransport for eVLBI at JIVE

Boundary conditions:

- Mark4 correlator
- Real time

Which means:

- Correlation within a second after observation
- Timely delivery of data is important

Exceedingly more difficult at higher data rates

- Packet loss increases
- Fixed size hardware buffers so timing gets more critical
- Power-of-two datarates

Datatransport for eVLBI at JIVE

Mark4 datarates

- 128Mb/s Routine for science observations
- 256Mb/s Possible in most "weather" conditions
- 512Mb/s If we get lucky
- 1Gb/s Impossible with std 1Gb/s ethernet

Bandwidth seems to be available, yet we can't always transfer data at 512Mb/s.

Using TCP for eVLBI

Probably not the optimal choice

- exponential backoff
- retransmission

Unfortunately...

- Mark5A UDP doesn't really work

C-TCP

Developed by CHEETAH project for circuit-switched networks

- Patches to Linux 2.6.11 that
 - ▷ Disable slow start completely
 - ▷ Fix the congestion window at the bandwidth delay product
- For connections marked by userland app
- Problems
 - ▷ Needs web100
 - ▷ Not available for other kernels
- Benefits
 - ▷ Application control possible

<http://cheetah.cs.virginia.edu/>

Linux Pluggable Congestion Avoidance Modules

- Minimally implement

- ▷ u32 ssthresh(struct sock *sk);
- ▷ u32 cong_avoid(struct sock *sk, u32 ack, u32 rtt, u32 in_flight, int good_ack);
- ▷ u32 min_cwnd(struct sock *sk);

- Optionally implement

- ▷ u32 rtt_sample(struct sock *sk, u32 usrtt);
- ▷ u32 undo_cwnd(struct sock *sk);
- ▷ And more...

- Access to "struct sock" with all details about the connection

- ▷ destination address & port
- ▷ source address & port

Linux Pluggable Congestion Avoidance Modules

Benefits

- Only the sender needs to be modified
- No need to reboot...but don't lock yourself in.
- Stable interface?

Disadvantages

- No application control
- Must run Linux 2.6.13 or later
- Documentation missing?

A simple module for eVLBI

- Based on the ideas from C-TCP
 - ▷ Very small slow start threshold
 - ▷ Fix congestion window
- Only activated when destination is Mark5A data port
- "Standard" Reno for other connections
- Bandwidth specified when module is loaded
- Less than 50 lines of code

Measurements

Bandwidth measurements using iperf between Amsterdam and JIVE

Mark5's directly connected to 1Gb/s links

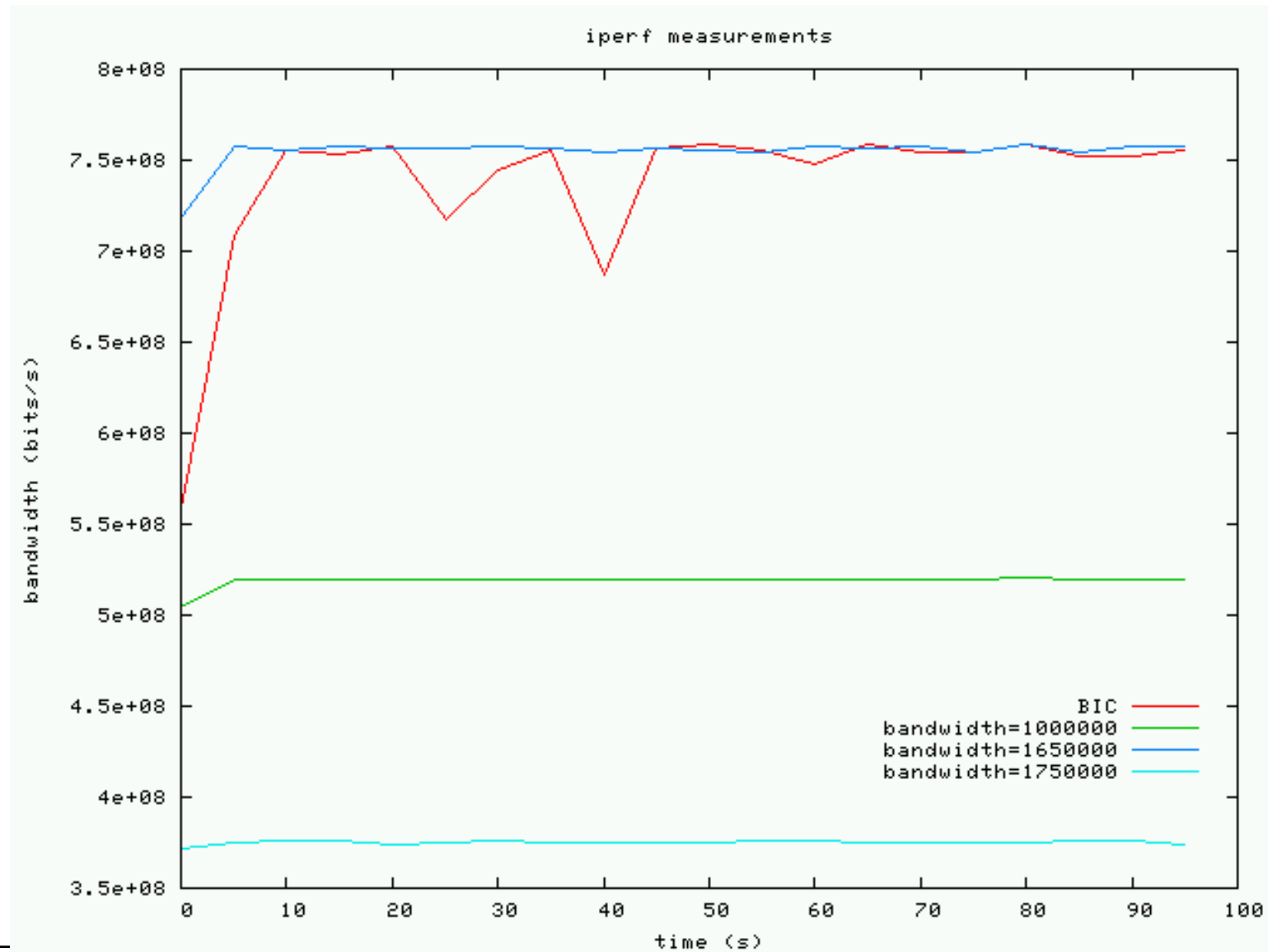
- Tests between two Mark5s
- With "background" traffic from a third Mark5

Conclusions

- Aggressive Congestion (non)-Avoidance can give an improvement
- Connectivity provided by SURFNet is too good
- The Netherlands is too small for meaningful tests

Measurements

Bandwidth measurements using iperf between Medicina and JIVE



Measurements

Bandwidth measurements using Mark5A failed

- Stable version of Mark5A code doesn't support Linux 2.6.13
- Development version seems to be broken

Datatransport up to 256Mb/s seemed to work

Conclusions

- Pluggable modules provide interesting flexibility
- Fluctuations in throughput are smaller
- Evaluation for real eVLBI remains to be done
- Application level changes needed

Thanks

Many thanks to

Giuseppe Maccaferri
Radiotelescopi di Medicina
Istituto di Radioastronomia