Heavy Data, Light Measurements: Network troubleshooting in a distributed computing environment

Domenico Vicinanza

DANTE, Cambridge, UK

perfSONAR MDM Product Manager

domenico.vicinanza@dante.net

QUESTnet 2013 Conference – 2-5 July 2013

Gold Coast, Queensland, Australia

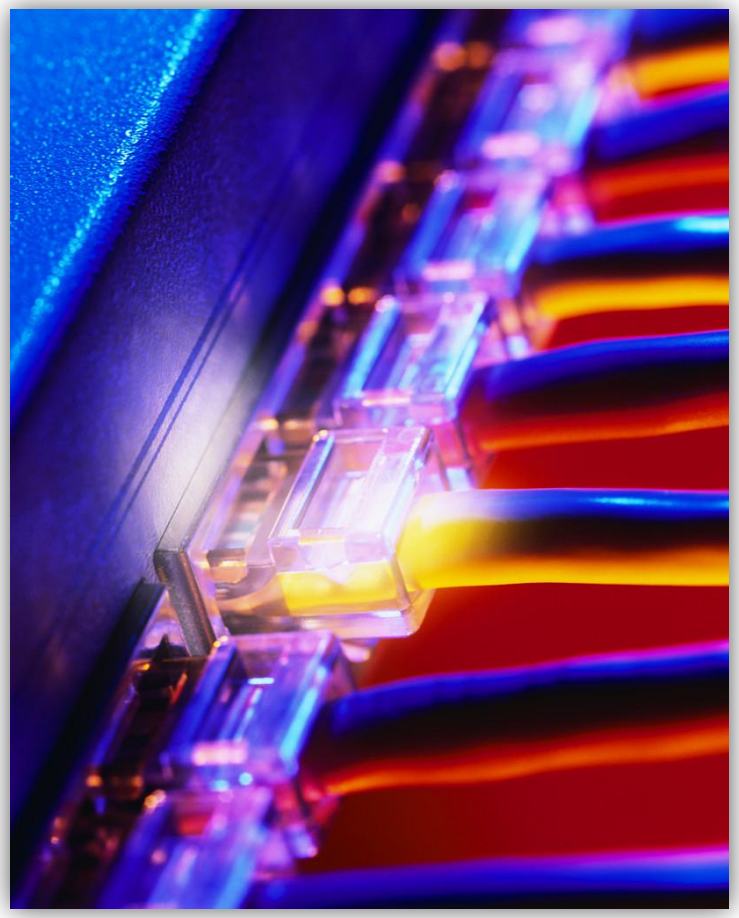
Collaboration today

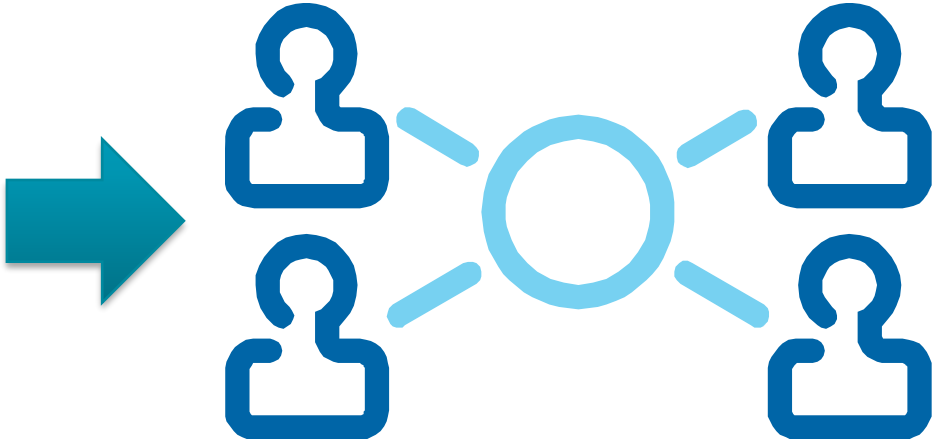
Distributed resources
Distributed computation

Grid computing and Clouds
Videoconferencing
Remote collaborations
Virtual teams

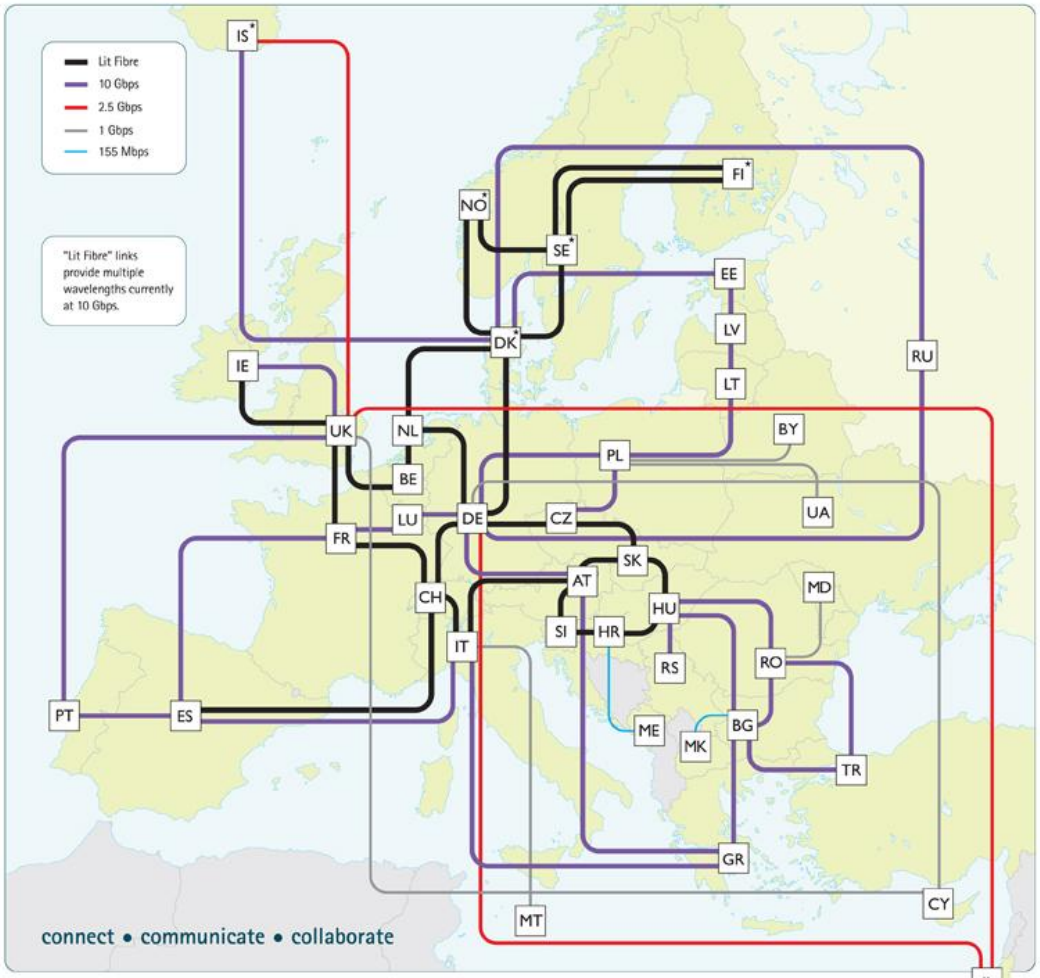
Picture courtesy of
Koniclab, GridCafe

How?
Networking ☺





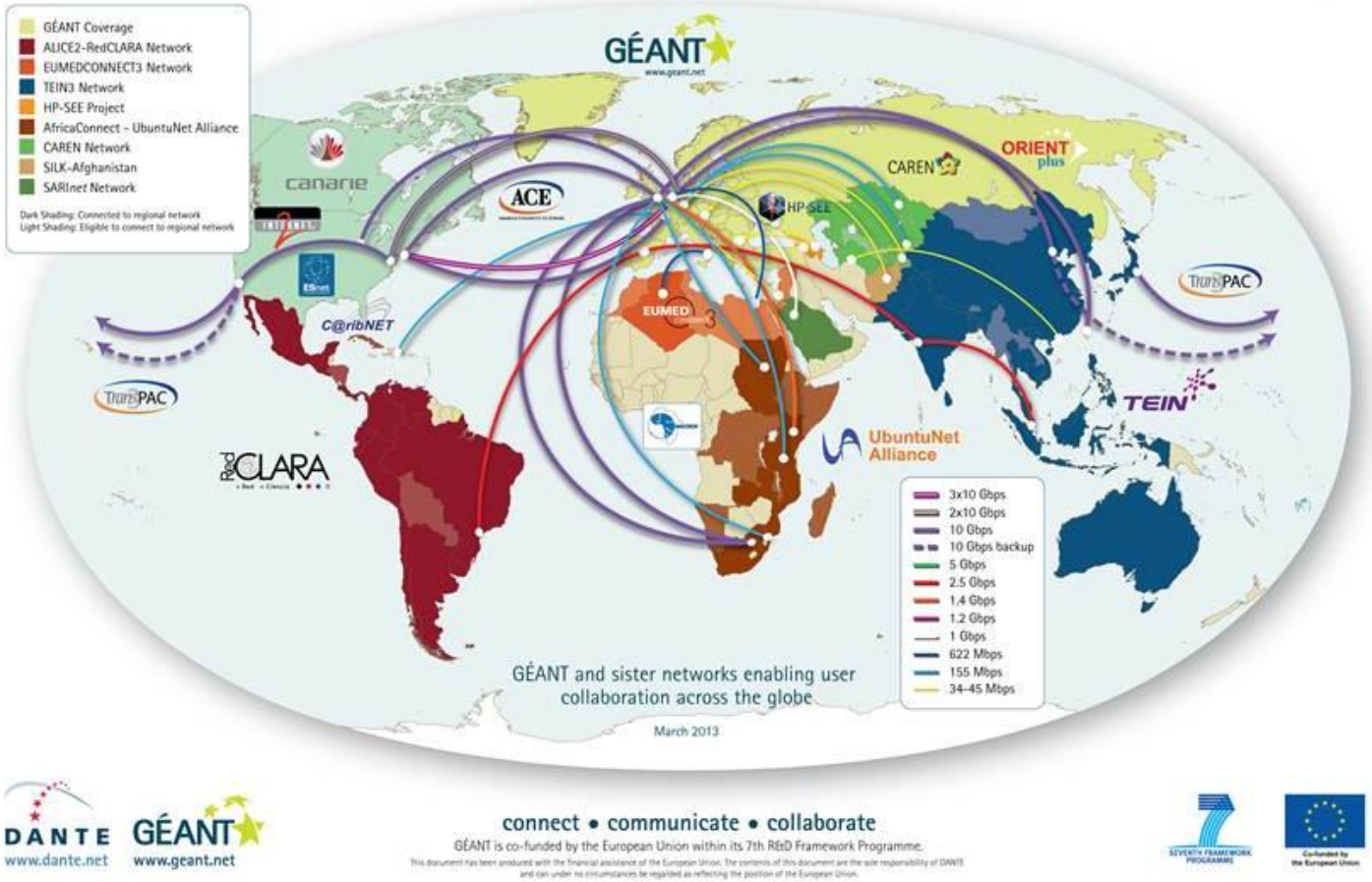
Europe's 100Gbps Network
- *e-Infrastructure for the "data deluge"*



Backbone topology as at March 2012. GÉANT is operated by DANTE on behalf of Europe's NRENs.

- Latest transmission and switching technology
- Routers with 100Gbps capability
- Optical transmission platform designed to provide 500Gbps super-channels
- 12,000km of dark fibre
- Over 100,000km of leased capacity (including transatlantic connections)
- 28 main sites covering European footprint

GÉANT Global Connectivity
- at the heart of global research networking



DANTE www.dante.net GÉANT www.geant.net

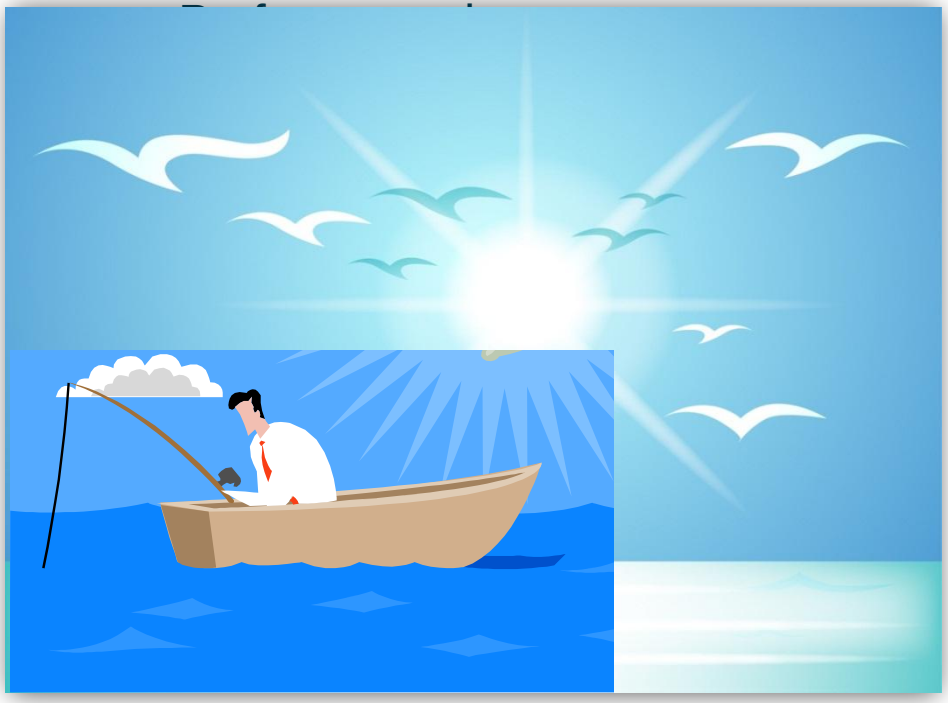
connect • communicate • collaborate

GÉANT is co-funded by the European Union within its 7th R&D Framework Programme.
This document has been produced with the financial assistance of the European Union. The contents of this document are the sole responsibility of DANTE and can under no circumstances be regarded as reflecting the position of the European Union.

GÉANT connects 65 countries outside of Europe, reaching all continents through international partners

Network monitoring

- Functioning network is the foundation for smooth operation
- Failure of mail servers, archives, or performance problems often cause tremendous damage.
- Network monitoring looks at



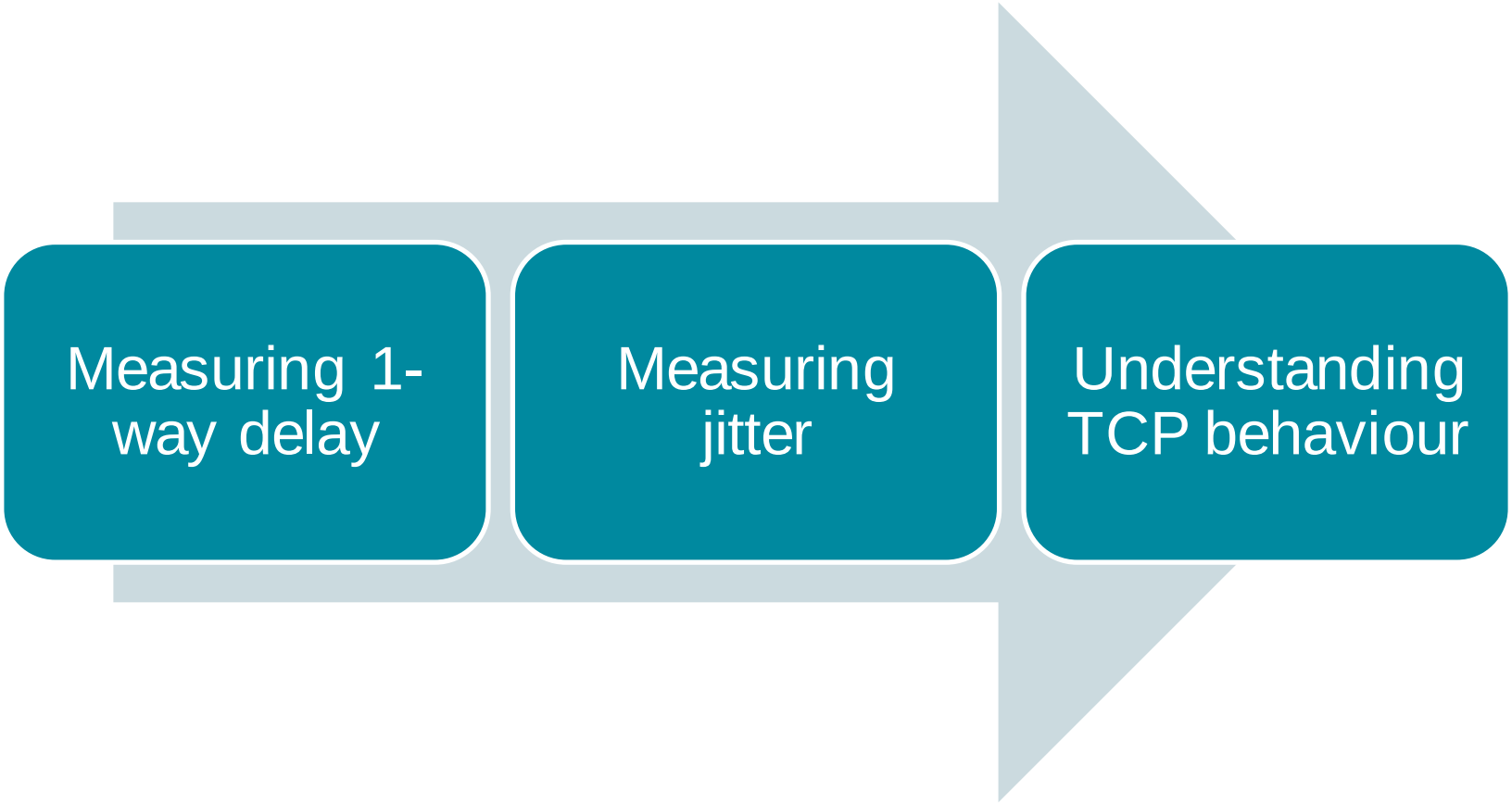
Where do we start from?

What to measure?

The problem

- In many practical situation (iperf or similar) bandwidth tests are the first tests to be done
- Occasionally regular bandwidth tests are carried out
  - To check the end-user bandwidth
  - To verify the links are healthy
  - Even to check if there is any suspicious losses
- Fairly accurate reproduction of the end-user experience but…
- Heavy, consistent test traffic is generated and transported
  - Competing with “real” end-user traffic
  - Contributing to congestion
    - *In particularly in highly distributed environments*
- **Is there a way to have the same info with lightweight measurements?**

The strategy



The solution!

- Network troubleshooting can make excellent use of lightweight network measurements
  - Measurement not competing with the production traffic
  - Measurement not competing with network resources
- Examples:
  - 1-way delay/RTT
  - 1-way delay variation (jitter)
  - Packet loss

The tools

perfSONAR MDM

UDPMon

Traceroute

OWAMP

Ping

First step: ping

- Monitor the RTT and packet loss at regular intervals
- Highlight systematic effects:
  - Rerouting
  - Systematic losses
- Methodology: 30 pings and calculating average RTT and % loss
  - every minute
- Example result:

Timestamp	RTT (ms)	(% loss)
0402_2109	115.408	0%
0402_2110	115.909	0%
0402_2111	**98.751**	**4%**
0402_2112	98.188	0%
0402_2113	98.345	0%

- RTT time difference → link length difference

The RTT monitoring script

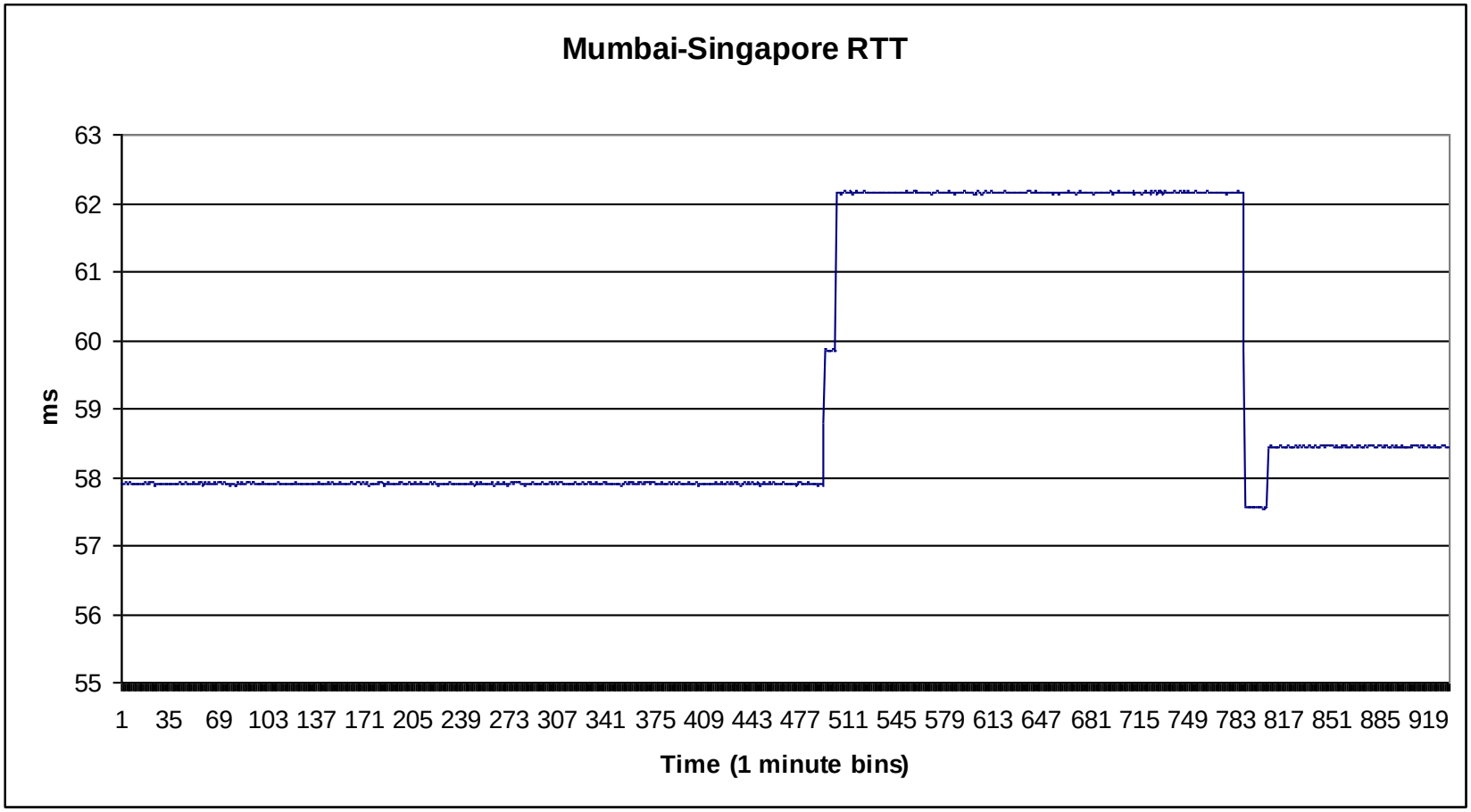
```
#!/bin/bash
host="<host_ip_address>"
counts="30"
logfile="rttlog.txt"
now=`date '+%d%m_%H%M'`
ping -q -c$counts $host > /tmp/ping
rtt=`grep "rtt" /tmp/ping|sed "s/\// /g"|awk '{ print
$8 }'`
loss=`grep "%" /tmp/ping|awk '{ print $6 }'`

printf "%s \t %s \t %s \n" $now $rtt $loss
printf "%s \t %s \t %s \n" $now $rtt $loss >> $logfile
```

RTT script: the results

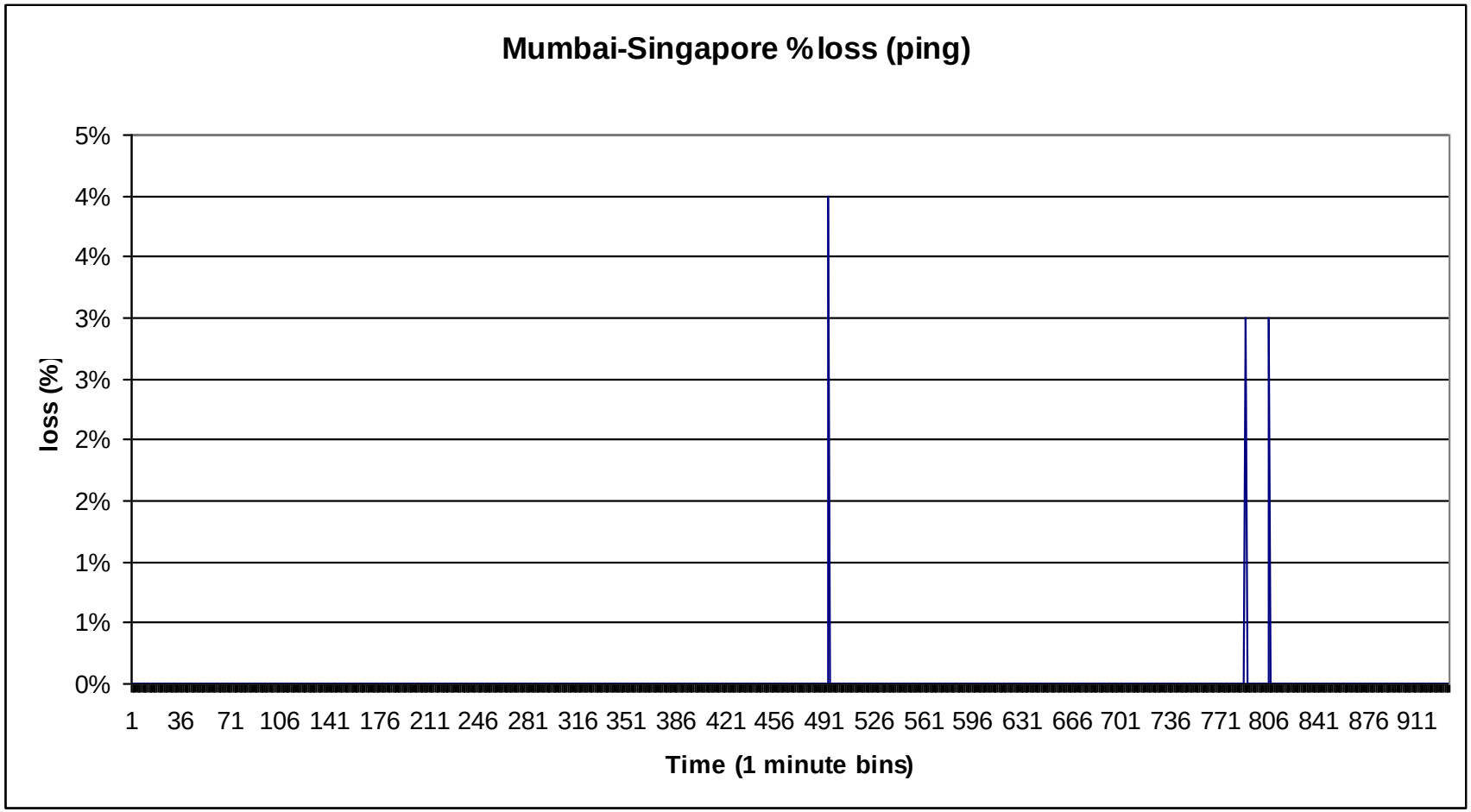
- The output file on a real link:

Timestamp	RTT (ms)	(% loss)
0402_2107	115.964	0%
0402_2108	115.365	0%
0402_2109	115.408	0%
0402_2110	115.909	0%
0402_2111	**98.751**	**4%**
0402_2112	98.188	0%
0402_2113	98.345	0%
0402_2114	98.679	0%

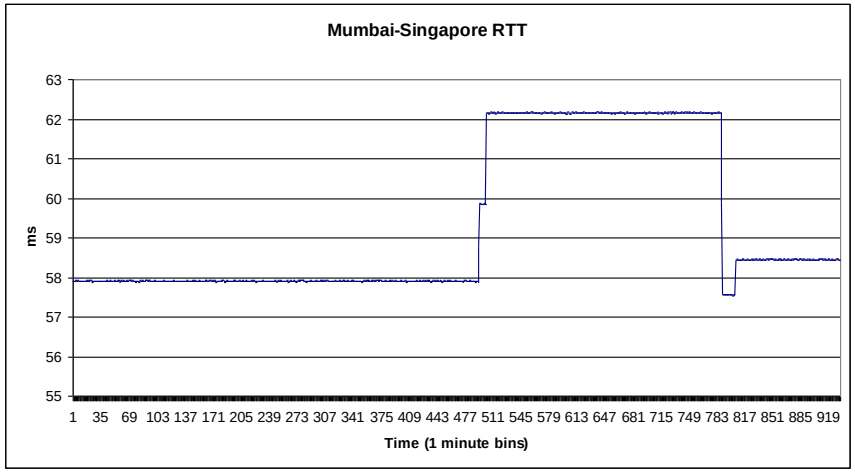
Visualising the results

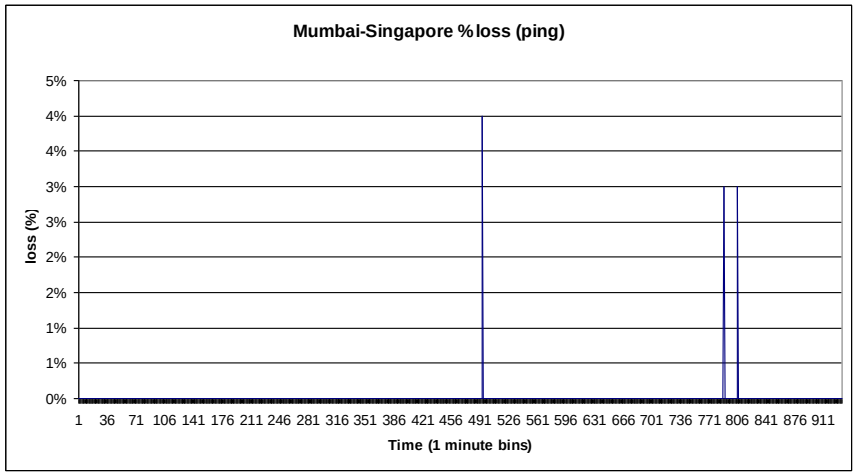


Packet loss


Mumbai-Singapore %loss (ping)
loss (%)
5%
4%
4%
3%
3%
2%
2%
1%
1%
0%
1 36 71 106 141 176 211 246 281 316 351 386 421 456 491 526 561 596 631 666 701 736 771 806 841 876 911
Time (1 minute bins)

Summarising the ping results





- RTT and %loss on a real link
- RTT changed 4 times during the test
  - 5 different RTTs
    - *5 different routes*
  - Traffic been routed on different SDH rings with different lengths
  - Non negligible packet loss during switchovers

1-way delay measurements RFC2679

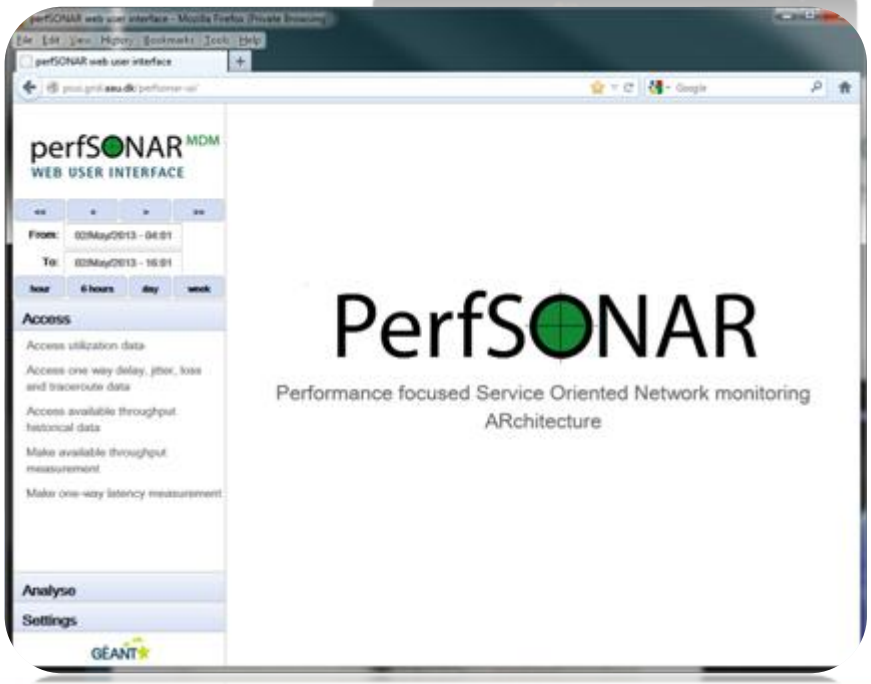
- Advantages of 1-way delay measurements over round-trip time
  - Spotting router asymmetry
    - *Highlighting performance differences between the two paths which may traverse*
  - Asymmetric queuing
    - *Even when the two paths are symmetric*
  - Performance of an application may depend mostly on the performance in one direction.
    - *Ex: A file transfer using TCP may depend more on the performance in the direction that data flows, rather than the direction in which acknowledgements travel.*

1-way delay measurements RFC2679 (cont.)

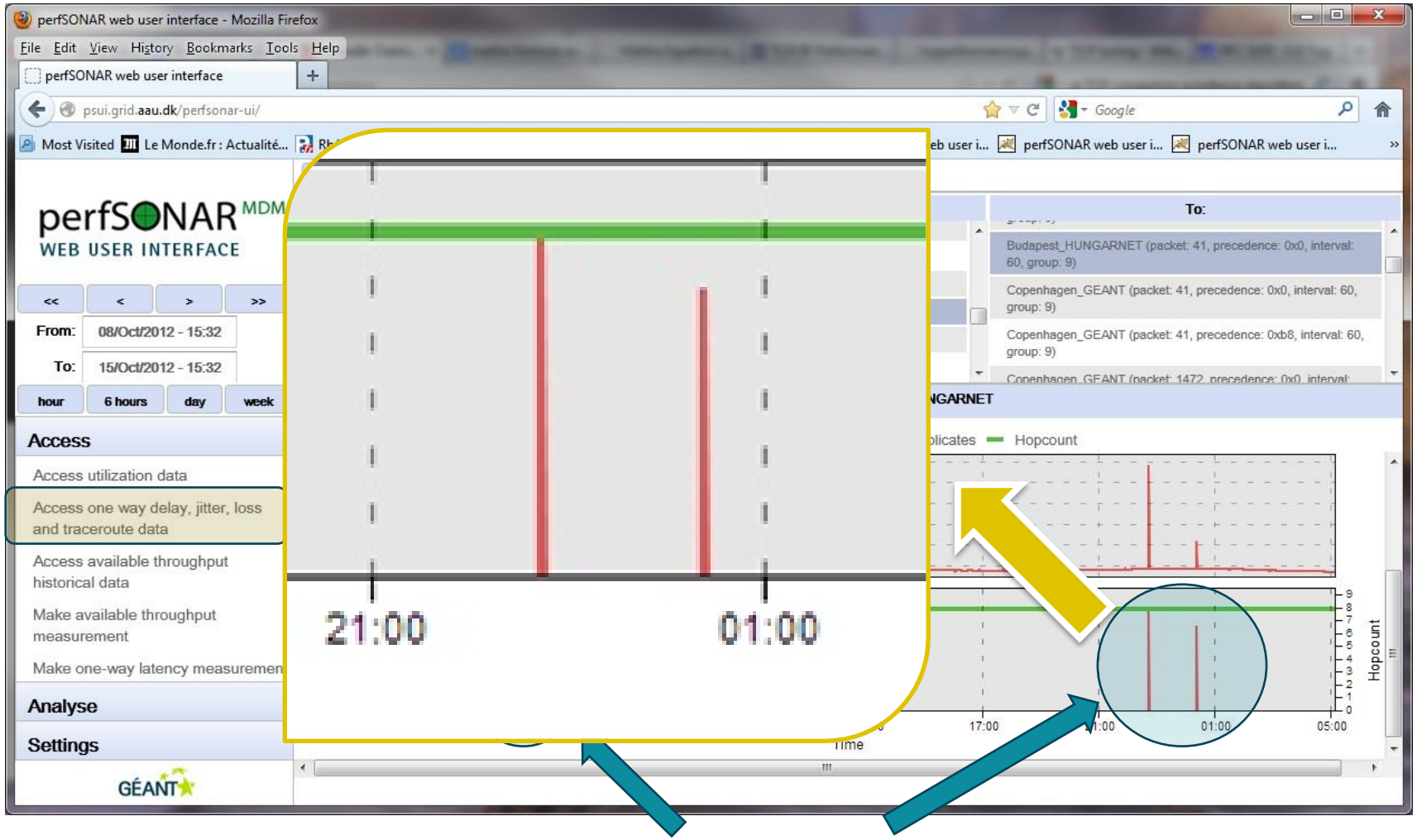
- Quality-of-service (QoS) enabled networks:
  - *Different provisioning in one direction than provisioning in the reverse direction,*
    - *thus the QoS guarantees differ.*
  - *Measuring the paths independently allows the verification of both guarantees.*

perfSONAR MDM

- Multi-domain monitoring service
- Based on monitoring probes installed in the network
- Based on a standard (perfSONAR) protocol
- Hundreds of deployments around the world
- Web interface!

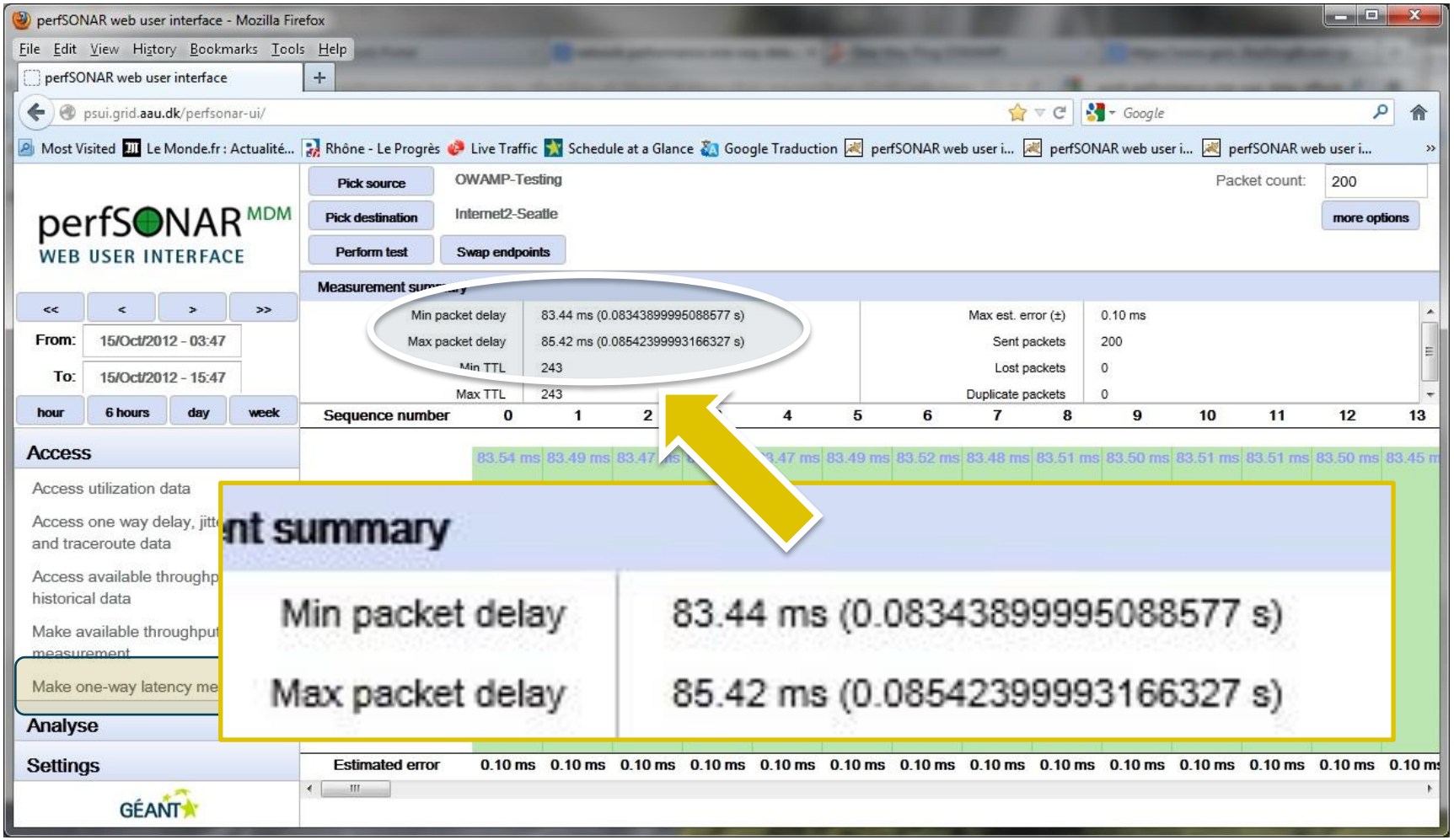


Example of packet loss: from regularly scheduled measurements



Look at here for packet loss

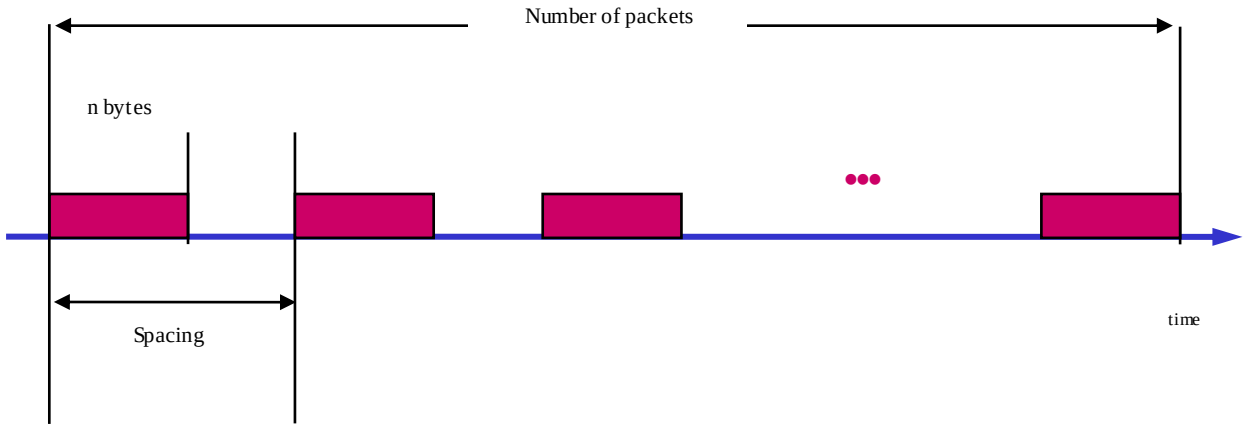
1-way delay on-demand (OWAMP) through perfSONAR MDM interface



Look at here for 1-way delay

UDPMon

- Developed by R. Hughes-Jones (DANTE & Univ. Manchester)
- http://www.hep.man.ac.uk/u/rich/net/index.html
- Client-server
- It works by sending a stream of carefully spaced packets

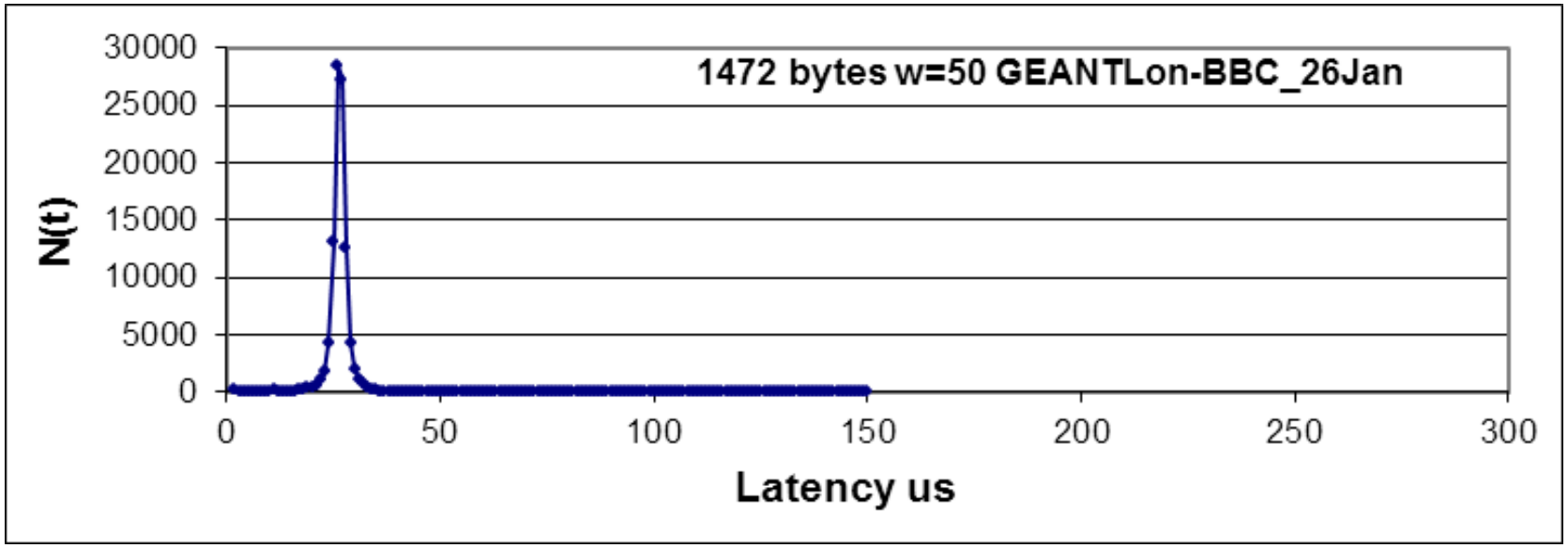


- And recording a set of metrics

What it records

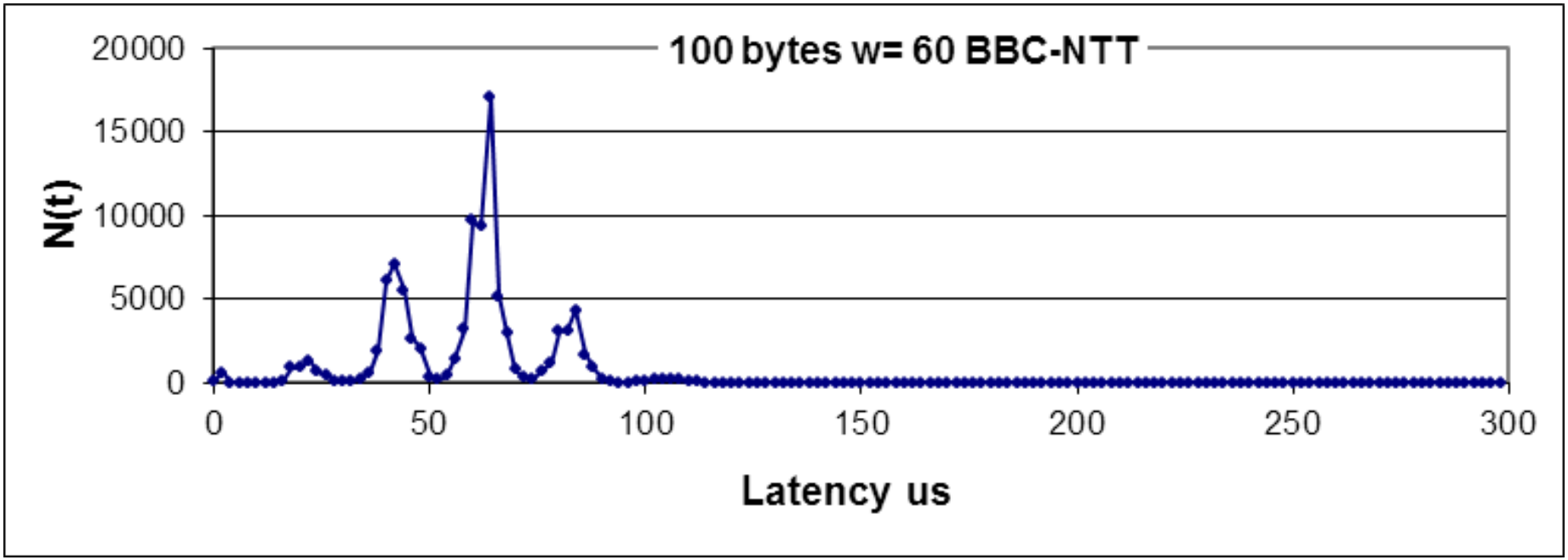
- Timestamp
- Number of packets received
- Packets lost (total)
- Packets arrived in bad order
- **Packet lost in the network**
- Bytes Received and Bytes/frame rate
- Elapsed time (microseconds)
- Time per received packet
- Receiver data rate and wire rate (Mb/s)

Jitter plots



Histograms of inter-packet arrival time for equally spaced packets (1472 Bytes packets, with 50µs spacing in this case)
This is a really good jitter plot, really narrow and no side bands

Jitter to discover queuing

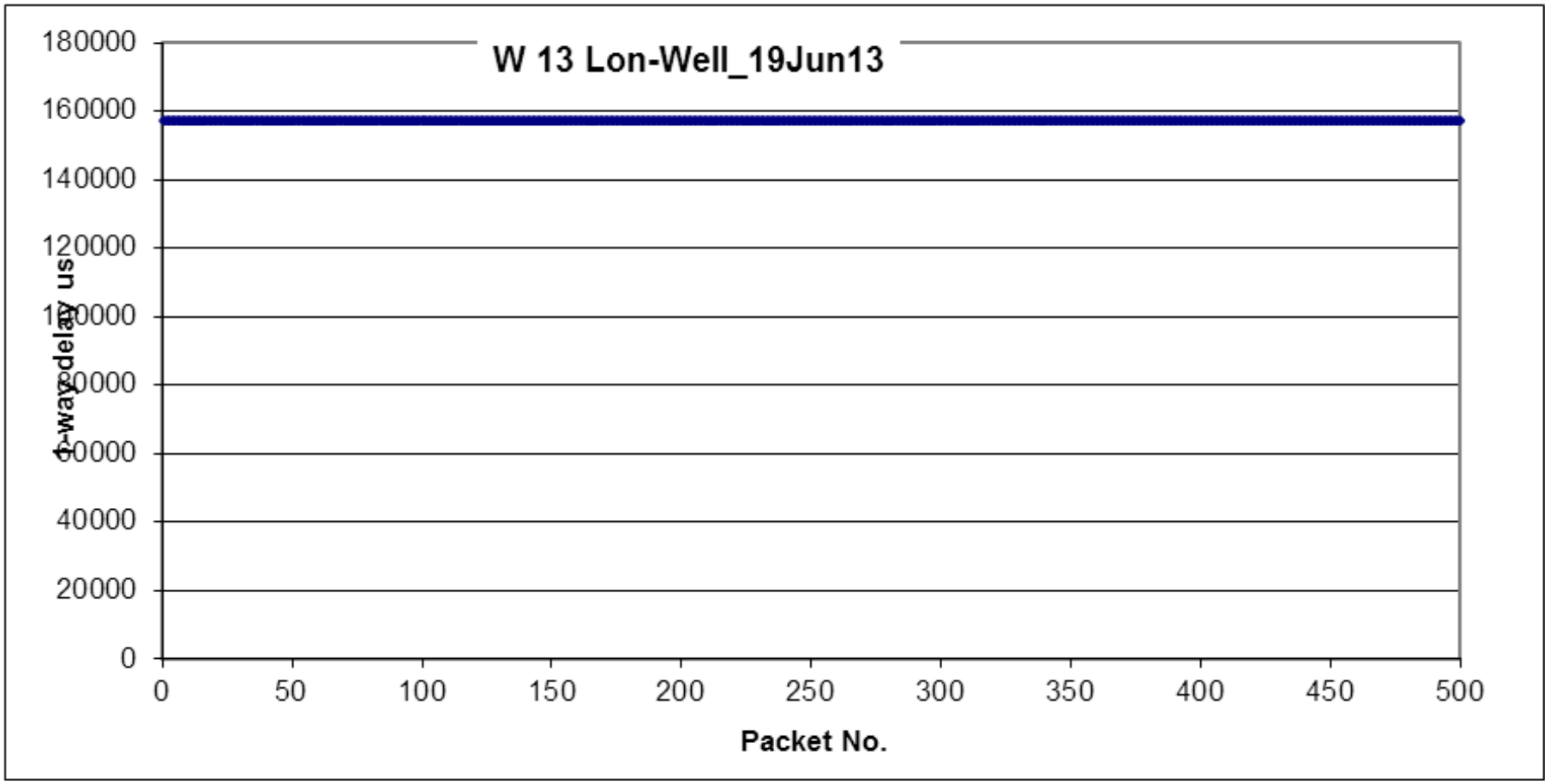


This is how queuing along the path is shown on a jitter plot:

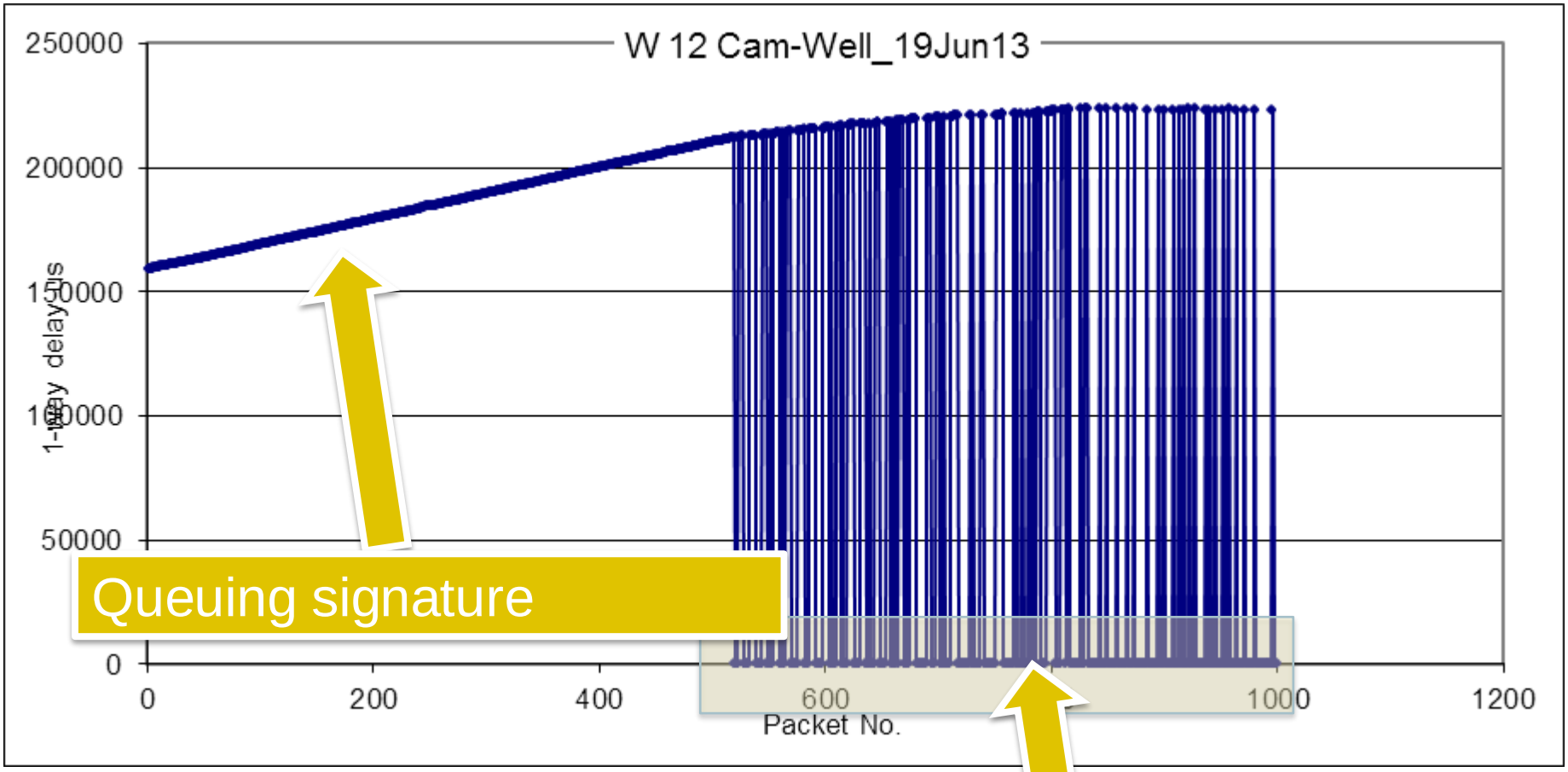
- Side bands
- Multiple peaks

This is a typical shape of a busy link with cross traffic

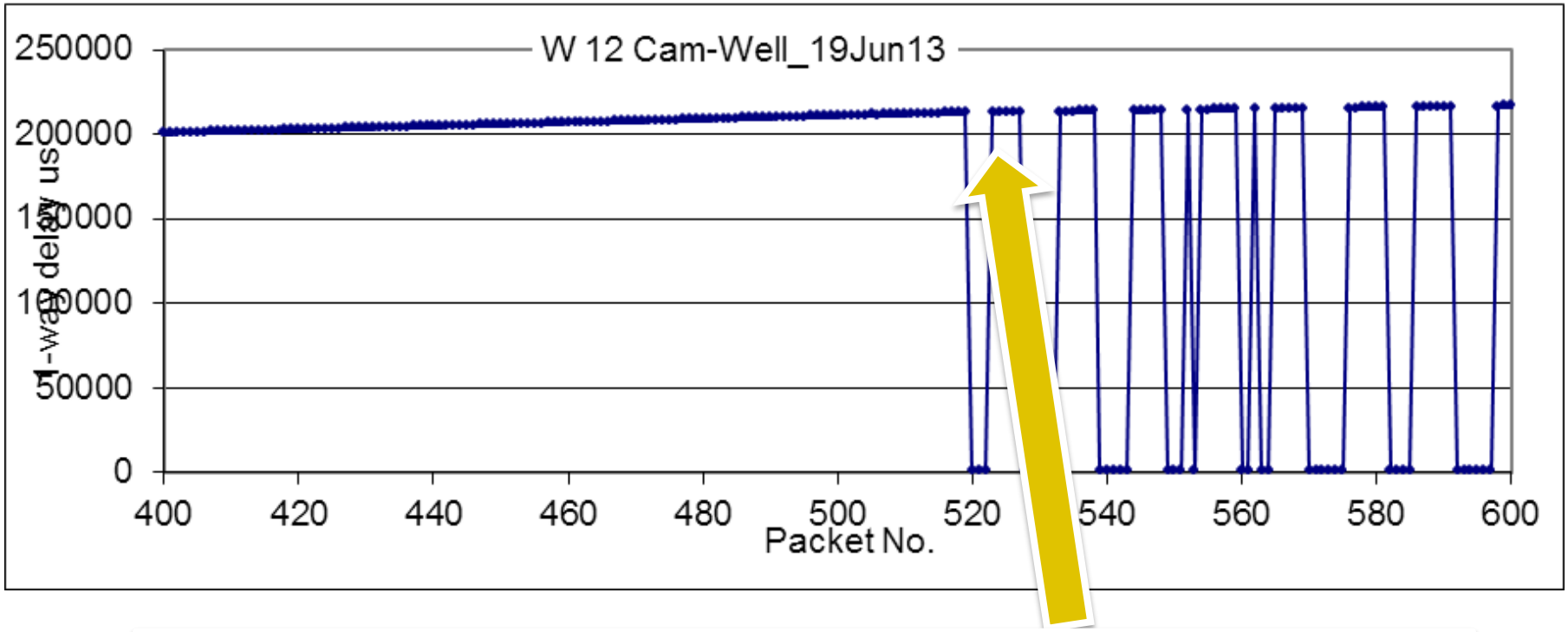
One-way delay graph



One way delay on a link with queuing and losses



Limiting buffer



There is clearly a buffer too small here!
Knowing the size of packets and the number of packets (=number of points between losses) it is possible to know the limiting buffer size!

TCP performances with packet loss

- TCP performances depends heavily on packet loss
- Because of the way TCP works, waiting for acks
  - Losing packets rapidly leads to performance degradation
- Mathis formula:

$$\text{Max rate} < \frac{MSS}{RTT * \sqrt{p}}$$

- where:
  - Max Rate: is the TCP transfer rate in bps
  - MSS: is the maximum segment size (e.g.1460) in bytes
  - RTT: is the round trip time in seconds
  - p: is the packet loss rate or loss probability (fraction)

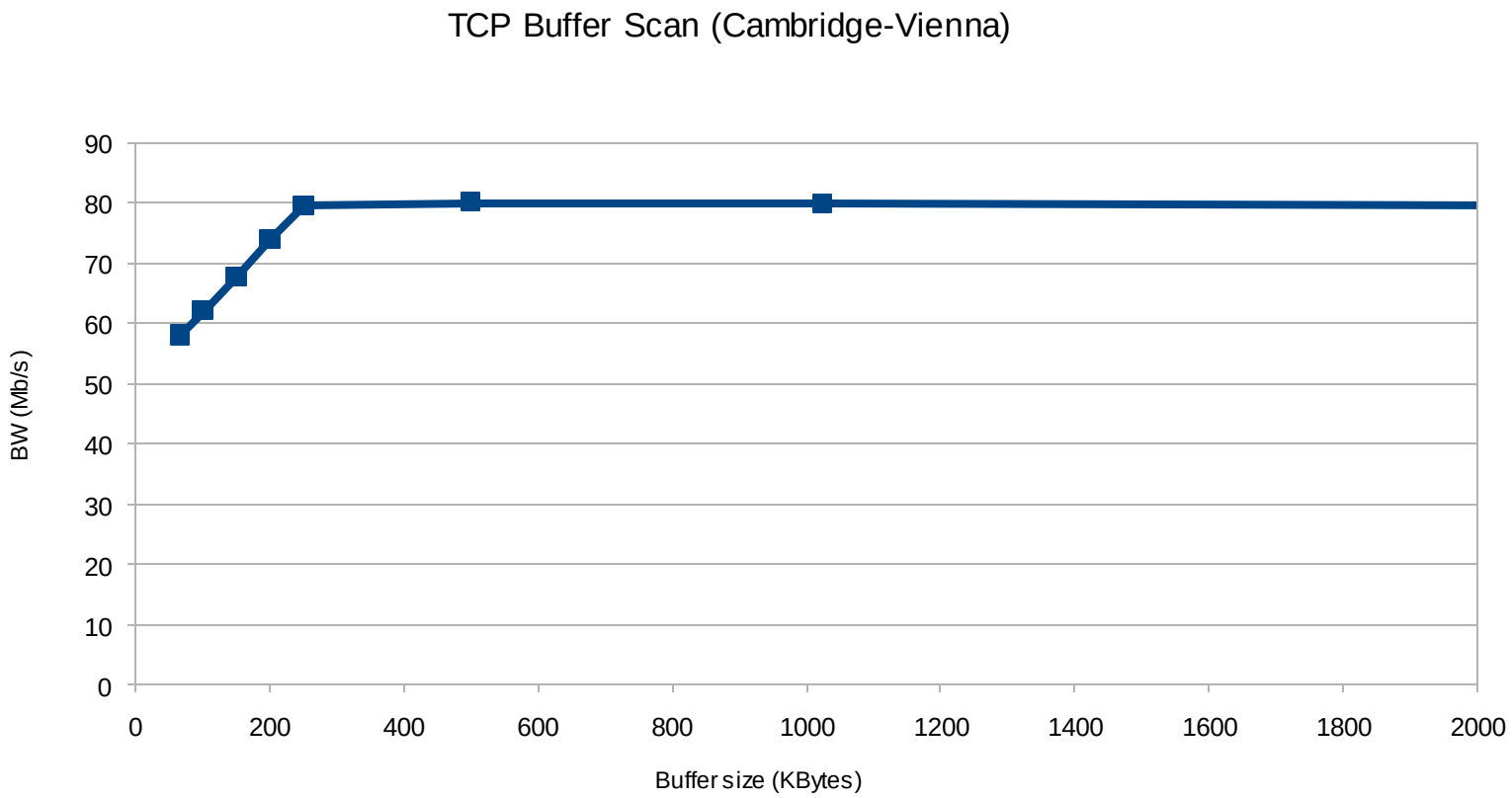
Formula from Mathis, et al. *The macroscopic behavior of the TCP congestion avoidance algorithm.* CCR, 27 (3), July 1997

Examples

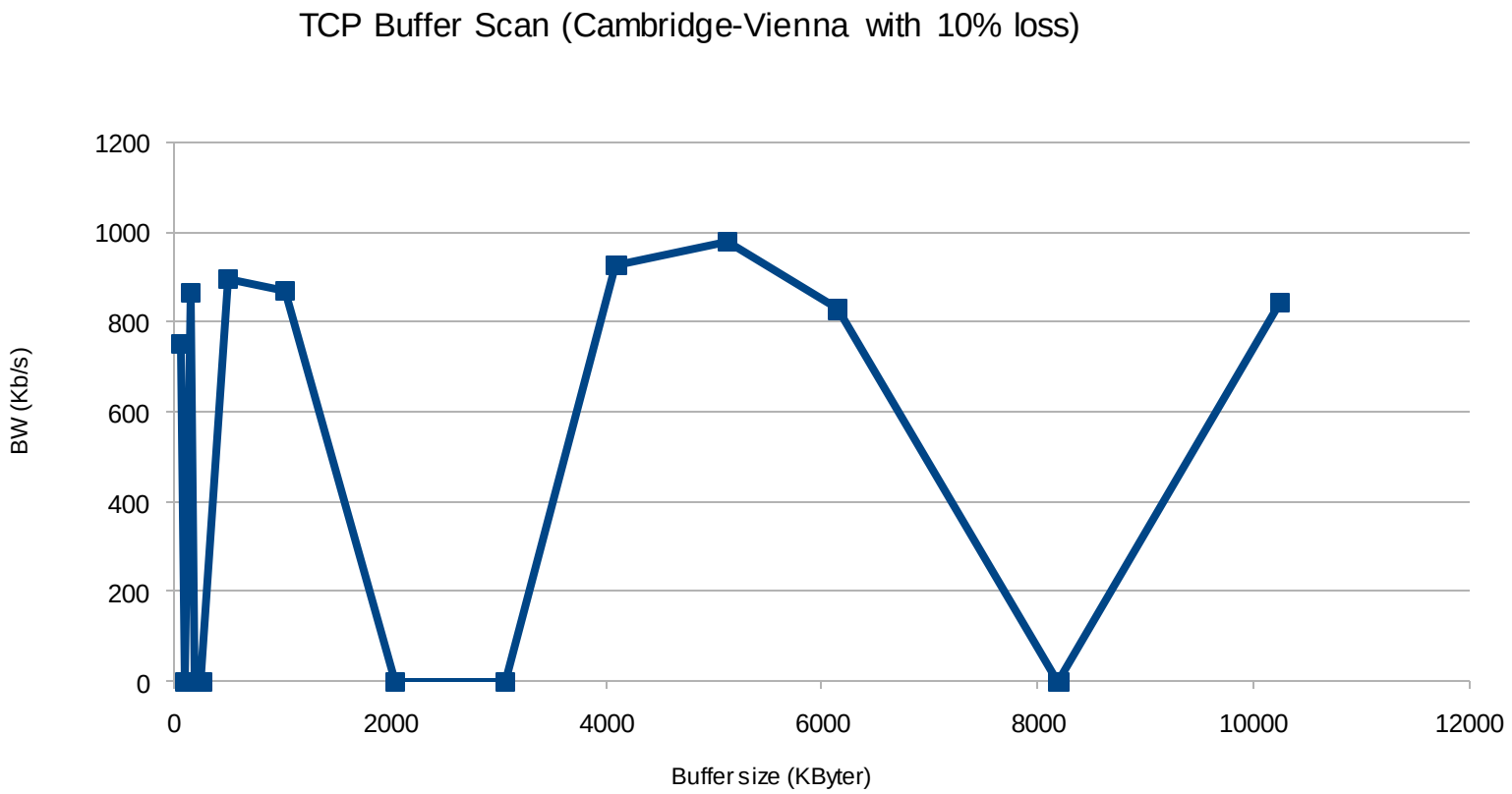
- The following examples assume MSS=1460 Bytes
- Example 1:
  - RTT= 88 ms
  - Loss probability: 1%

  → Max BW=**1.33 MB/s**
- Example 2:
  - RTT= 20 ms
  - Loss probability: 0.1% → Max BW=**18.5 MB/s**
- Example 3:
  - RTT= 20 ms
  - Loss probability: 1E-6 → Max BW=**5.8 GB/s**

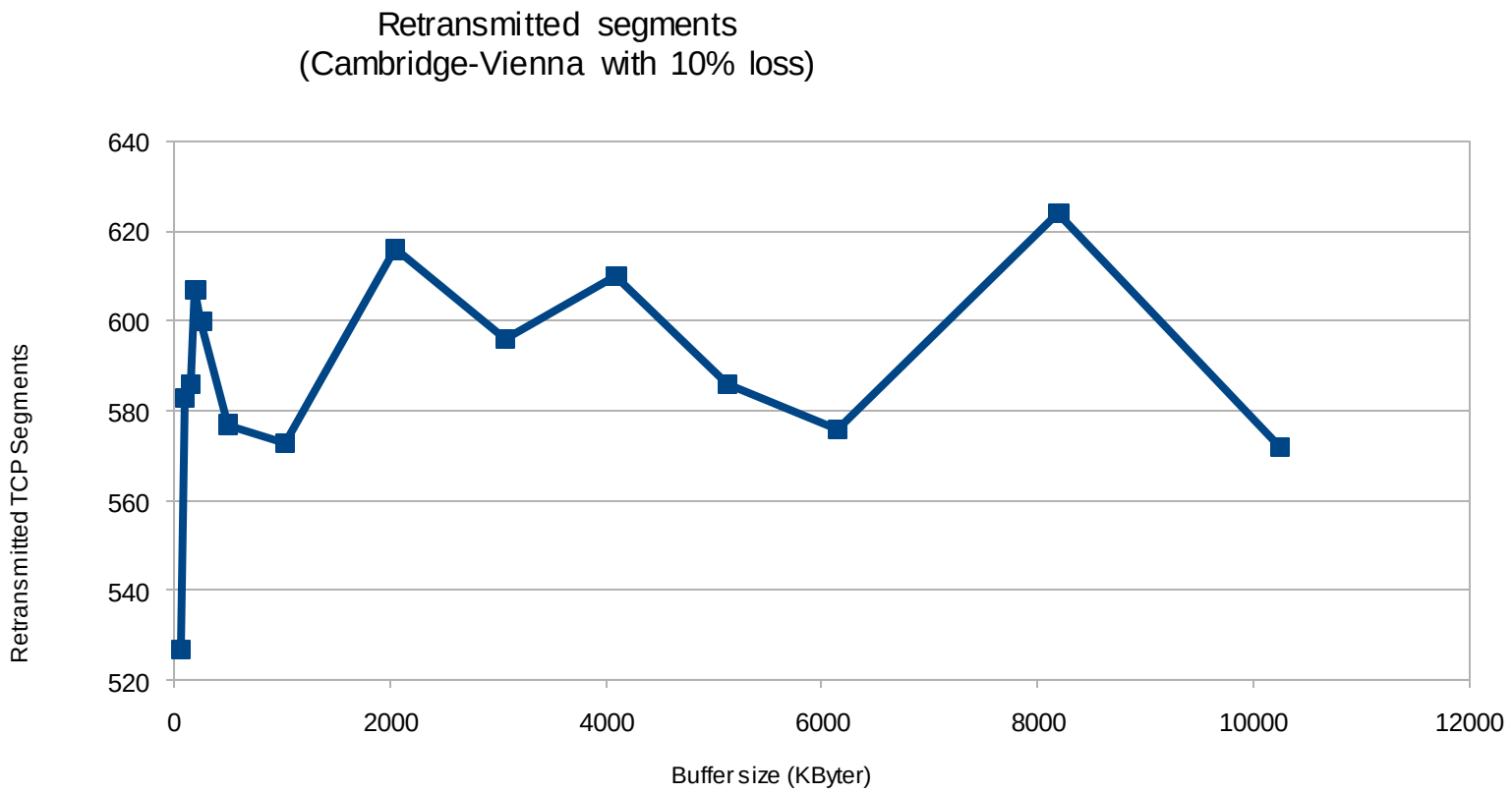
TCP Buffer scan
Cambridge-Vienna (no packet loss)



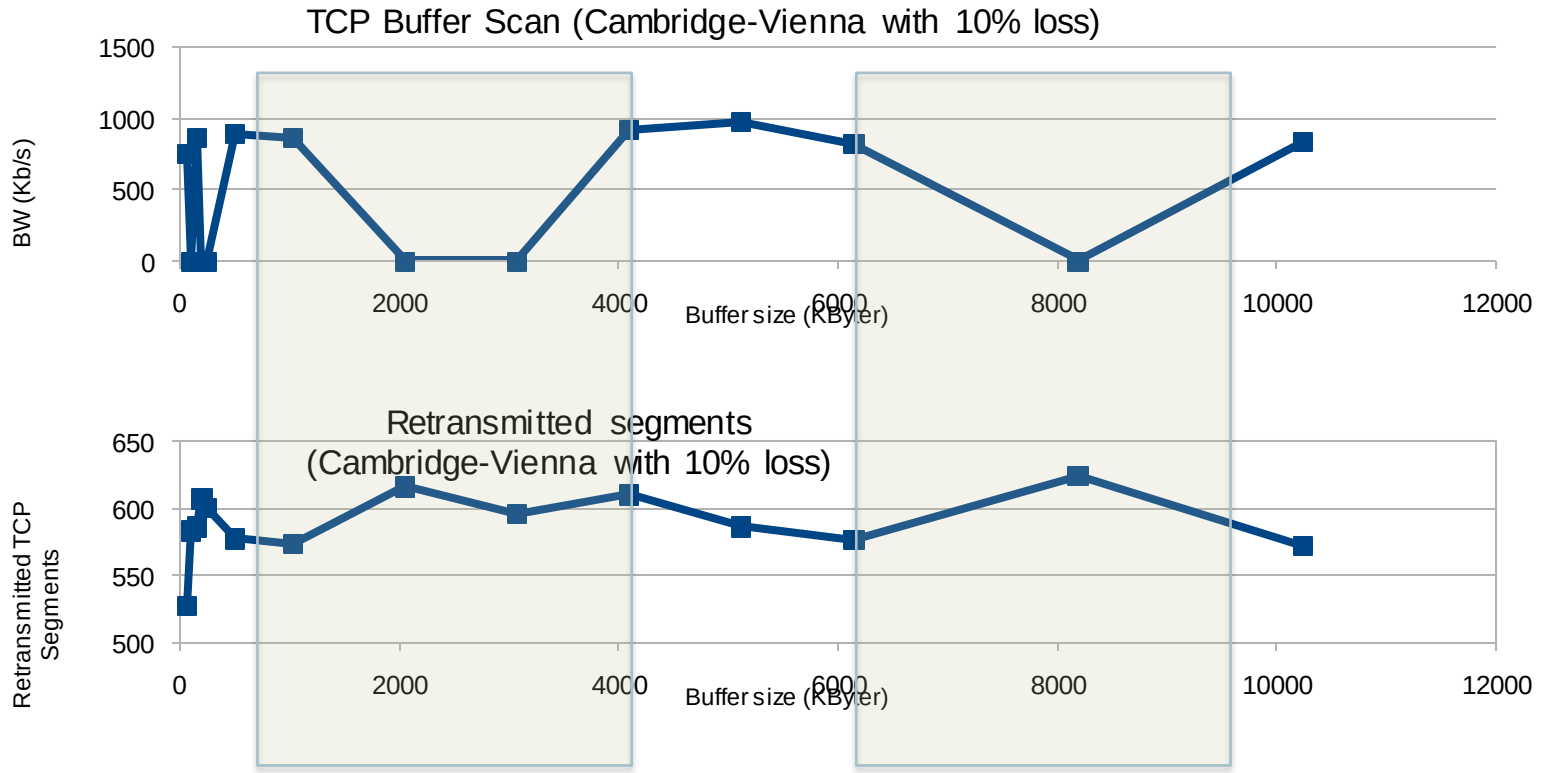
TCP Buffer scan
Cambridge-Vienna (10% loss)



TCP Retransmitted segments Cambridge-Vienna (10% loss)



Packet loss → Low TCP Bandwidth



Packet loss dominates TCP behaviour

Conclusions

- Lightweight measurements are extremely useful for network troubleshooting purposes
  - They do not disturb the network traffic
  - They do not take vital resources
  - Yet they provide precious information about the health of the network
- Most end user application uses TCP
- Testing TCP can be intrusive/expensive
- TCP behaviour can be deducted from lightweight measurements
- perfSONAR MDM and UDPmon have full support to use them
  - Retrieve and visualise regularly scheduled measurements
  - Run on-demand measurements when needed
  - From the new web user interface
    - *http://psui.geant.net*

perfSONAR MDM. Be part of it.

Follow perfSONAR at:
http://twitter.com/#!/perfSONARMDM

- Website: http://perfsonar.geant.net
- Twitter: @perfSONARMDM
- Info: domenico.vicinanza@dante.net

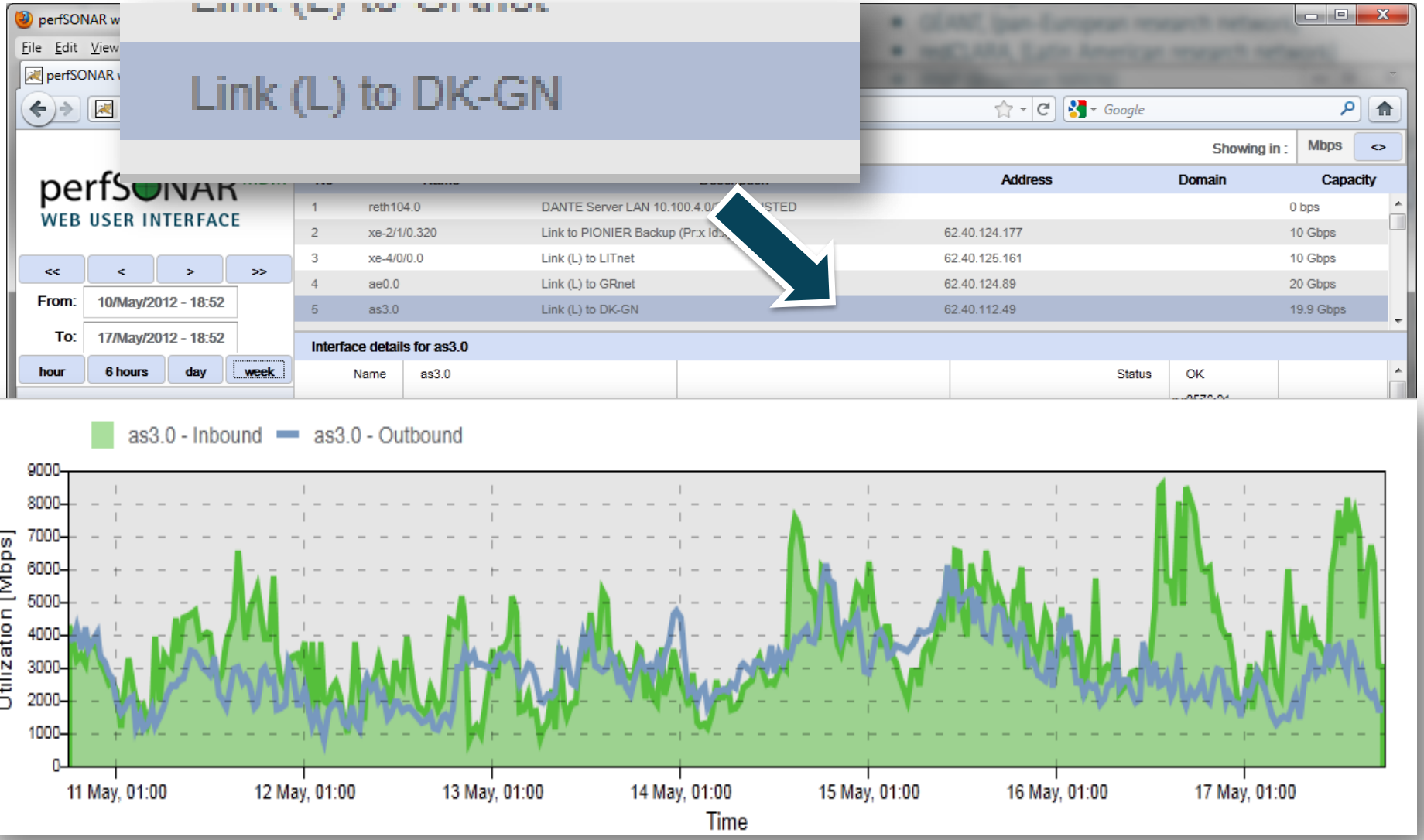
Thank you!

Connect | Communicate | Collaborate

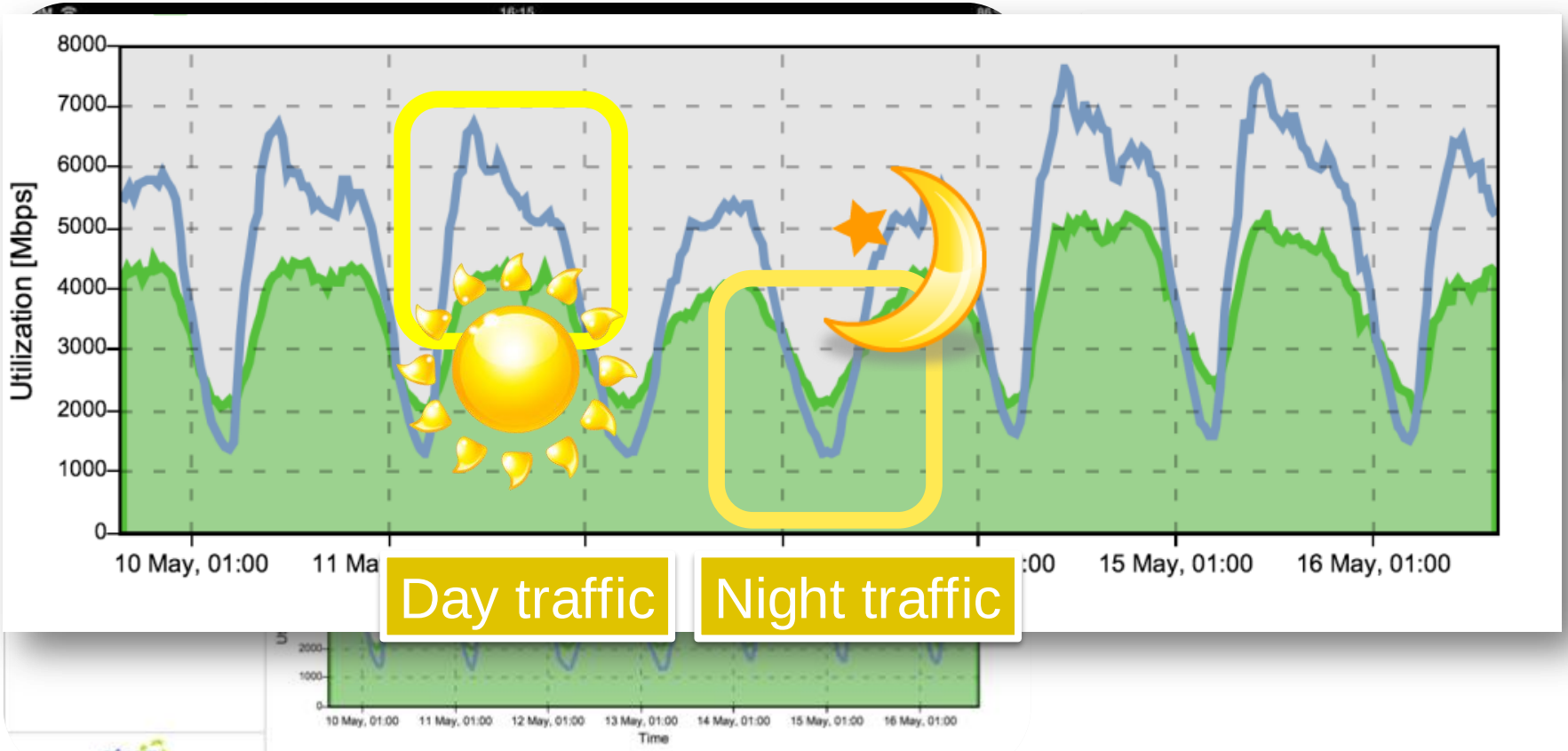
www.geant.net

www.twitter.com/GEANTnews | www.facebook.com/GEANTnetwork | www.youtube.com/GEANTtv

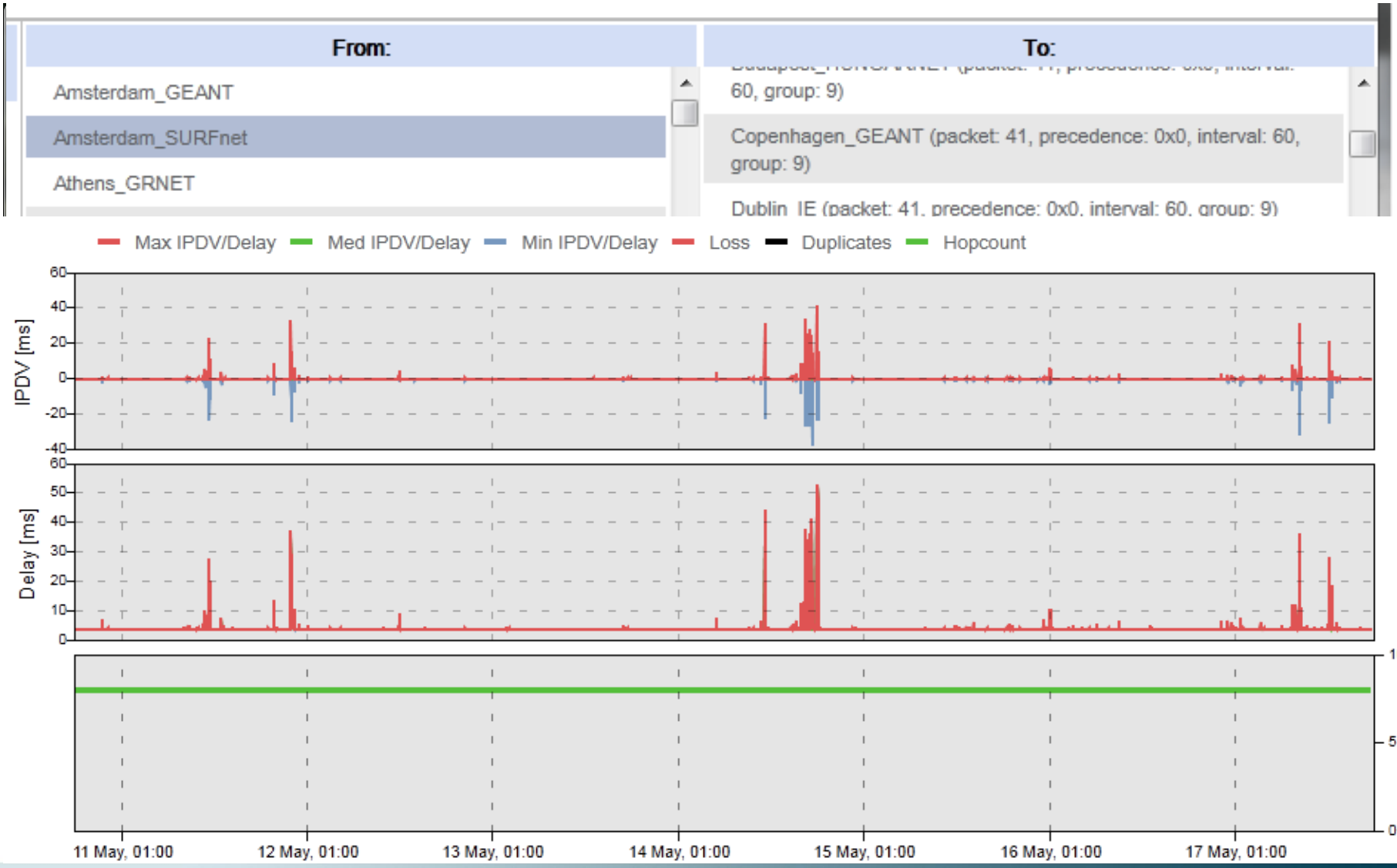
Link Utilisation User Interface



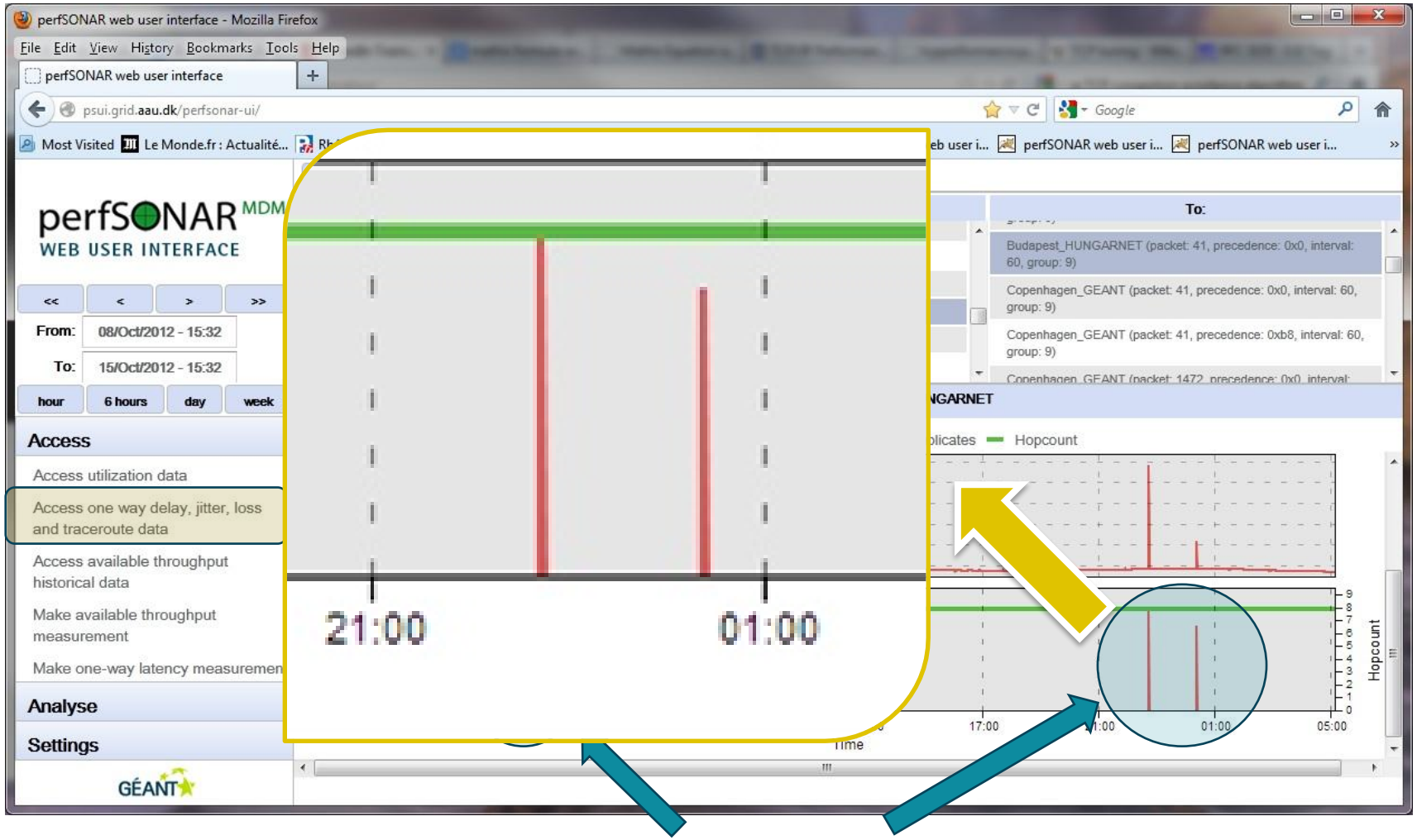
Link utilisation on an iPad



Web User Interface – OWD, jitter packet loss

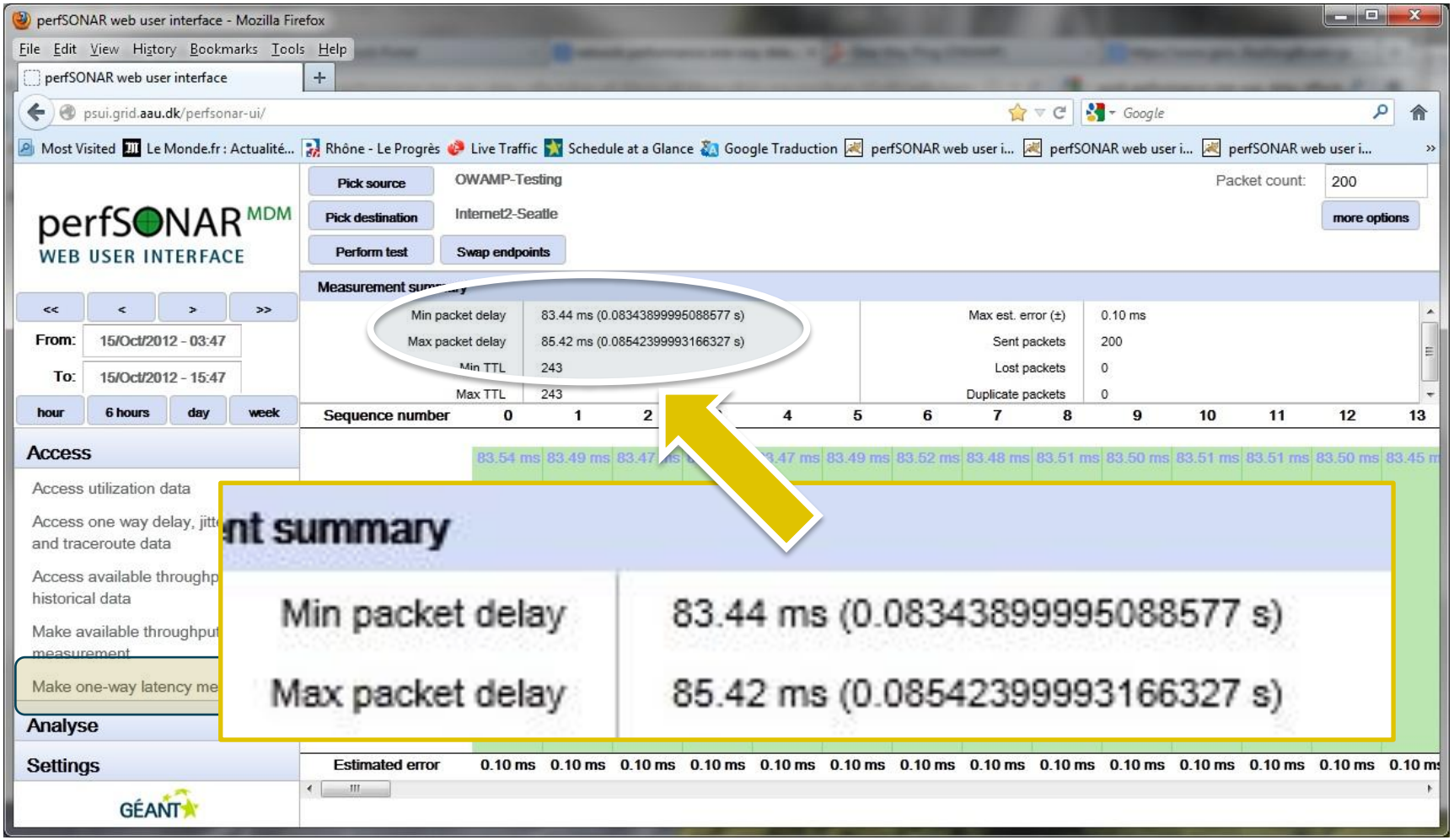


Example of packet loss: from regularly scheduled measurements



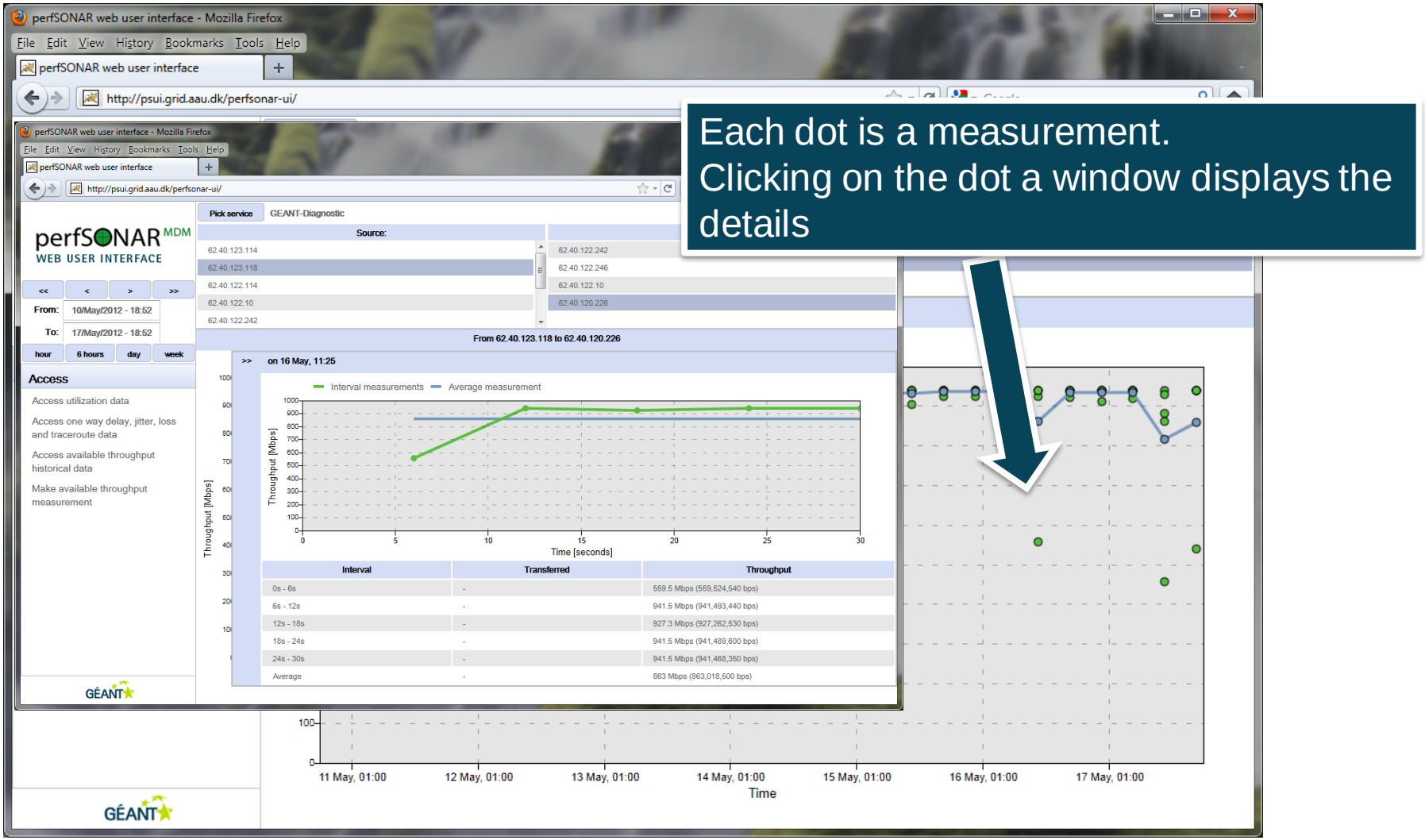
Look at here for packet loss

1-way delay on-demand (OWAMP)



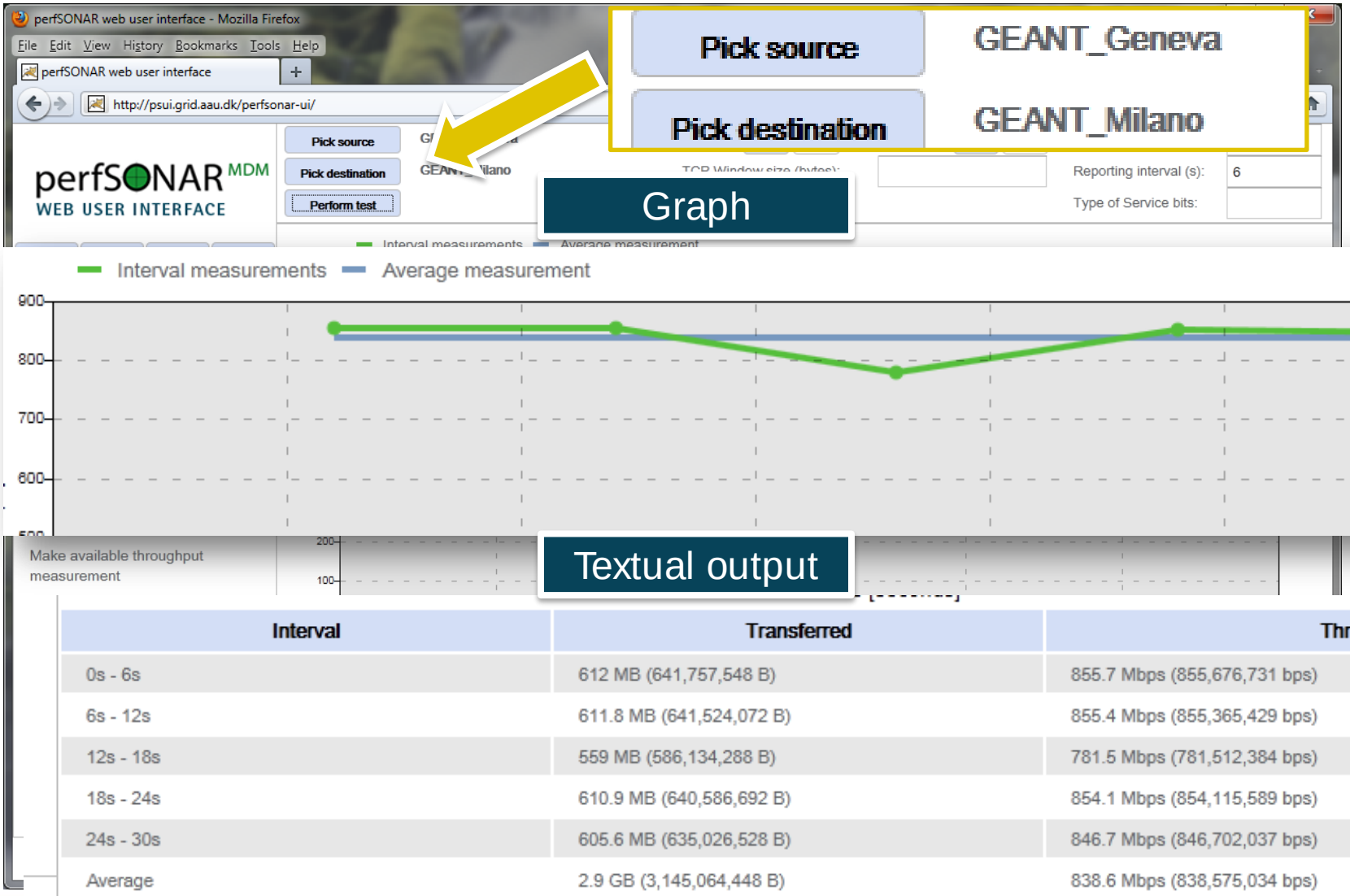
Look at here for 1-way delay

Accessing Historical Bandwidth Measurements



Each dot is a measurement.
Clicking on the dot a window displays the details

…and getting the results in two clicks from the web interface



Interval	Transferred	Throughput
0s - 6s	612 MB (641,757,548 B)	855.7 Mbps (855,676,731 bps)
6s - 12s	611.8 MB (641,524,072 B)	855.4 Mbps (855,365,429 bps)
12s - 18s	559 MB (586,134,288 B)	781.5 Mbps (781,512,384 bps)
18s - 24s	610.9 MB (640,586,692 B)	854.1 Mbps (854,115,589 bps)
24s - 30s	605.6 MB (635,026,528 B)	846.7 Mbps (846,702,037 bps)
Average	2.9 GB (3,145,064,448 B)	838.6 Mbps (838,575,034 bps)