



Extending the IPFIX protocol for
better QoS monitoring

22.05.07

Olav Kvittem, Arne Øslebø

End to end QoS measurements

- Goal : Knowing the QoS by measurements
- Traditional SNMP/netflow measure volumes of packets and errors on network components - not quality of service
- End to end active measurement does not scale
- Both engineering and customer oriented statistics
- User deserve end-to-end inter domain view
- Passive probes see the flows quality

Scampi/Lobster Software

- MAPI - Measurement API – in C
 - ◆ Interfaces passive measurement cards(DAG, Combo6)
 - ◆ Abstraction, sharing, branching, anonymization, efficiency(0-copy)
 - ◆ Functions – filtering hw/sw, counters, flow analysis
- Applications
 - ◆ Service detection - appmon
 - ◆ Polymorphic attack detector
 - ◆ Extended flow analysis – Stager
 - ★ SubSecond Bandwidth measurement(SSB)

Flow characterization

- MAPI with passive monitoring cards
 - ◆ Investigate flows per packet with microsec clock
- Extended IPFIX flow records with statistics for intensity, intervals, sizes
 - ◆ Will allow us to assess quality by service and location (AS or IP prefix).
 - ◆ Research actual quality of services

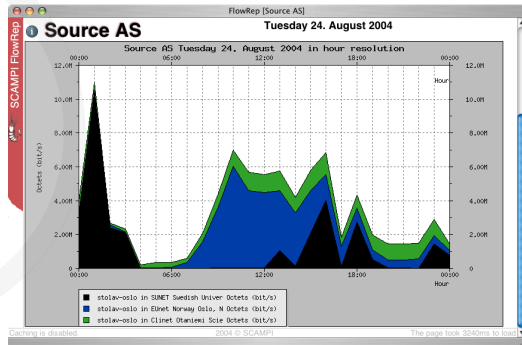
IPFIX flow

- IPFIX is an IETF standardization inspired by netflow v9 (Cisco)
- Extensible : custom defined record types and parameters
- Flow is defined by a flow key :
 - ◆ std. 5-tuple : IP-adresses, transport protocol and ports
 - ◆ end by FIN, time gap or max
 - ◆ Parmes : bytes, octets, time, AS, protocol, ..

Extended flow parms.

- Compute cheap numbers :
count, sum, sum of squares =>
standard deviation
- Packet size distribution
- Interarrival time distribution
- Bit rate vs time – 1, 10, 100,
1000 ms (max/min)
- RTP time-stamps/payload-type
- TCP properties : windows,
retransmissions/out-of-seq
- direction - initiator
- Service classification

Framework



Stager user interface



<http://www.ist-lobster.org>



- Flow collector based on NERD
- Stager backend

- Passive monitoring card
- MAPI

7

Stager DB

Collector

Exporter

Splitter



Destination IP report

Stager - Firefox

File Edit View Go Bookmarks Tools Help

Destination IP table Standard 20 Show

Add

All interfaces ??? In none 1

Destination IP Monday 18. September 2006, 15:00
??? (in, 1/1)

Pie chart Plot graph

Select	Dst IP	Octets	Rate 1 second		Rate 100 milliseconds		Rate 10 milliseconds	
	<input type="checkbox"/> IP address	<input type="checkbox"/> Total	<input type="checkbox"/> Max	<input type="checkbox"/> Min	<input type="checkbox"/> Max	<input type="checkbox"/> Min	<input type="checkbox"/> Max	<input type="checkbox"/> Min
<input type="checkbox"/>	w.x.y.z	18.8M	10.5M	780k	15.8M	0	53.8M	0
<input type="checkbox"/>	w.x.y.z	8.91M	5.21M	583k	5.37M	0	32.4M	0
<input type="checkbox"/>	w.x.y.z	7.75M	195k	75.3k	1.33M	163k	4.72M	5.78k
<input type="checkbox"/>	w.x.y.z	8.79M	143k	140k	937k	731k	7.62M	315k
<input type="checkbox"/>	w.x.y.z	7.48M	45.9k	2.38k	143k	1.88k	1.22M	7.34k

Stager, 2004-2006 © UNINETT AS Processing the report took 131.8ms

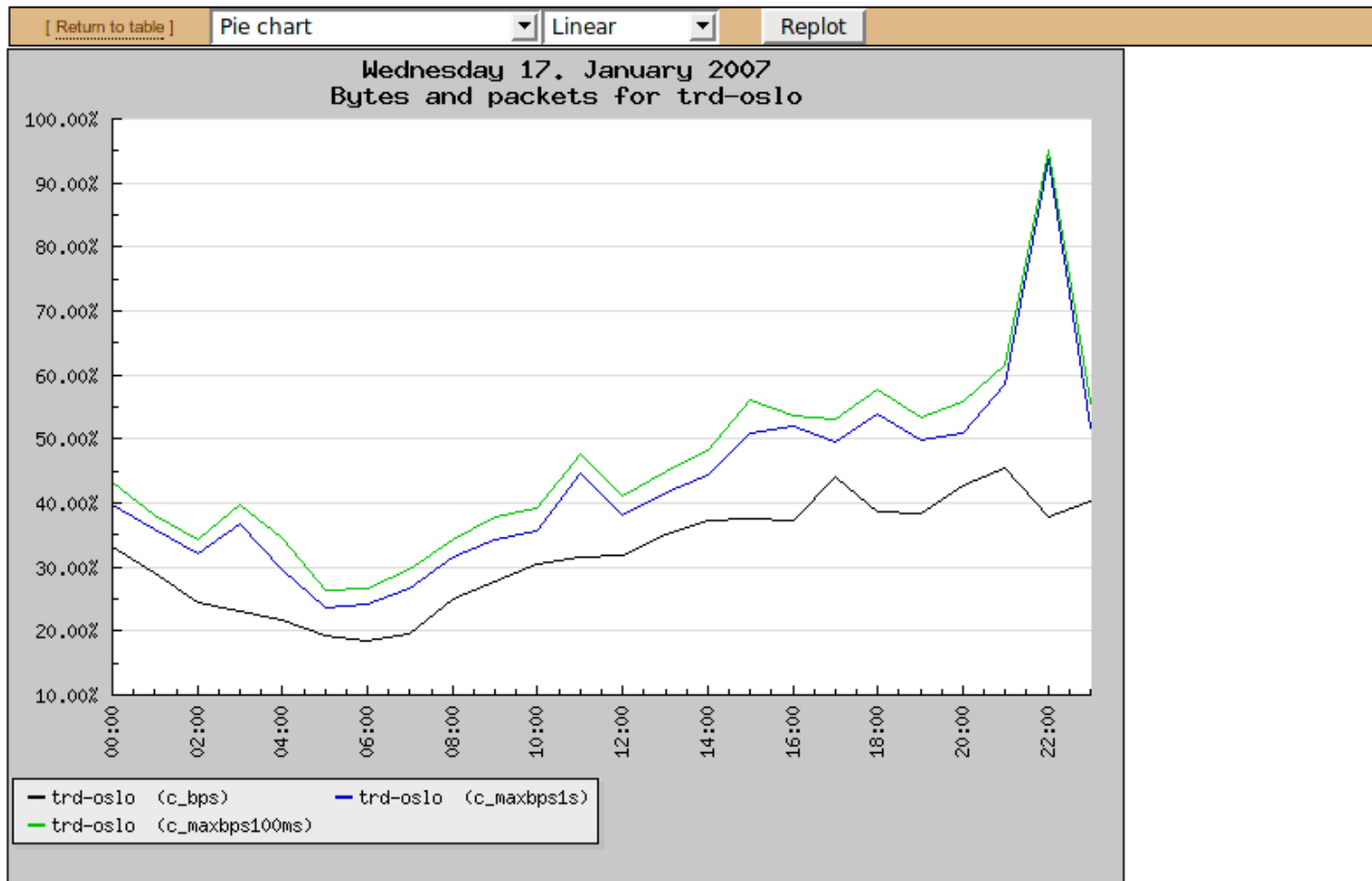
SSB

(Sub-Second Bandwidth)

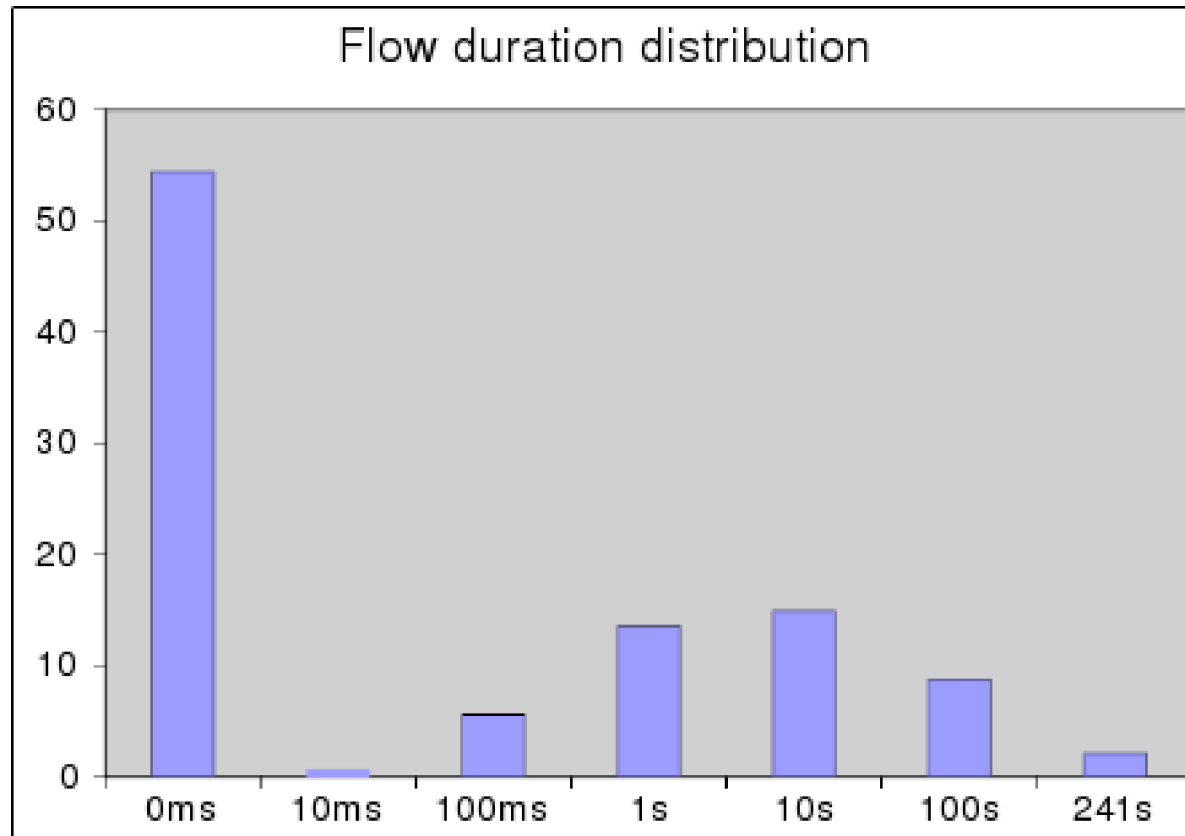
Setup Bytes and packets summary Capacity 20 Show Add << < > >> >>> ^ hour...
All interfaces trd-oslo In none 1

Bytes and packets

Wednesday 17. January 2007



Flow length



Web performance

- Flows > 100 ms to measure 100ms bursts
- TCP Src port 80 = web servers
- who initiated the flow – TCP SYN
 - request 9.1 mflow 63.2 GB
 - response 6.2 mflow 213.3 GB
 - uncertain 7.8 mflow 729.9 GB
-

AS - Web server performance

- 1ms 10ms 100ms eff win
- Kbps Kbps Kbps Byte
- 52535 7330 1582 5743
- 37056 4793 1729 14456
- 56092 8710 2147 2762
- 41778 5975 2193 13628
- 54324 7476 2197 14513
- average flows per AS, > 100 ms src port 80

Out of sequence TCP

- Count of TCP sequence $<$ previous
- does not distinguish :
 - retransmission of lost packets
 - reordered by network
 - cost CPU for sequencing out of fast path
- Some networks reorder
- distinguish by retransmission time \gg
reordered packet distance ?

Most out of sequence dst-AS

- out 10ms 100ms eff win
- % Kbps Kbps Byte
- 0.9 8449 3482 7105
- 1.4 4584 502 5210
- 1.5 5903 1007 8814
-
- 11.9 1917 204 3081
- 13.6 1054 112 1650
- 15.1 1445 156 2467
- 21.4 1973 208 3120

Scaling MAPI

- MAPI/ipfixlib copes on 2.5Gbps
- 10Gbps demands
 - ◆ parallelism - splitting captured data on more cpus for processing
 - ◆ offloading processing to hardware - flow generation too complex ?
- Faster report-generator – in parallel - ipfix/netflow v6 ?

Summary

- Passive flow measurements can
 - measure performance and indicate quality
 - show bursts in flows – 1ms too low
 - TCP effective window and out of sequence indicates bad performance
 - RTP measurements for further study
- application recognition demanding
- More work to be done
-
- Thanks to Scampi and Lobster partners
-
- Our software : <http://software.uninett.no>
- Lobster : <http://www.ist-lobster.org>