UNIVERSITY OF OSLO

A Practical Evaluation of Load Shedding in Data Stream Management Systems for Network Monitoring

Jarle Søberg, Kjetil Hernes, Matti Siekkinen,
Vera Goebel, Thomas Plagemann
University of Oslo, Department of Informatics
P.O. Box 1080, Blindern, N-0316 Oslo
{jarleso, kjetih, siekkine, goebel, plageman}@ifi.uio.no

- Outline
  1. Overview
  2. Motivation
  3. Experiment setup
  4. Tasks and results for TelegraphCQ
  5. Conclusion and future work

UNIVERSITY OF OSLO

Overview

- TelegraphCQ as a network monitoring tool
- Investigate load shedding in TelegraphCQ
  - "Black box"-testing
  - Two tasks
    - Design
    - Results

UNIVERSITY OF OSLO

Motivation: Passive network monitoring with DSMSs

- Capture data packets from the network
  - No insertion of own packets, just listen!
  - Obtain information from the captured data
- Send packets as *tuples* to the DSMS
  - A data packet can be viewed as a *tuple*
  - The packet's header fields can be viewed as *attributes*

- Real-time
  - Intrusion detection
  - Traffic engineering

UNIVERSITY OF OSLO

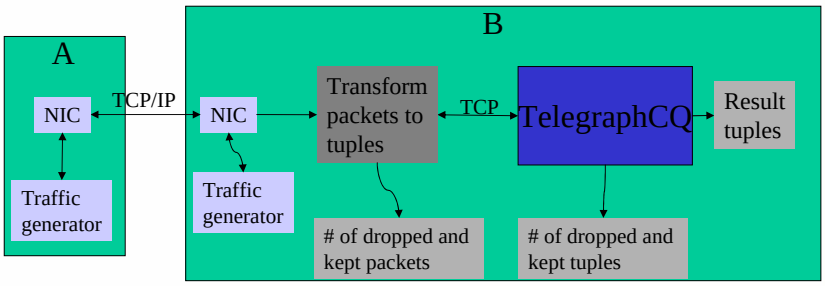
Motivation: Passive network monitoring and load shedding

- Reduces number of packets to be processed
- Network characteristics
  1. A considerable amount of traffic data
     - Gbit/s backbones used by ISPs
  2. Peak periods
     - Low activity at night
  3. Non-interesting data
- Several techniques
  - Simplest: Packet dropping

UNIVERSITY OF OSLO

Experiment setup

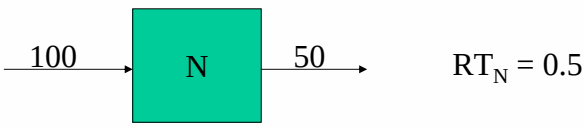
- How does the tuple dropping start?
  - Stress the DSMS
- Generate TCP/IP traffic between two machines
  - Linux SuSE 9.2 (vanilla installation)
  - Two 3 GHz Intel Pentium 4 processors
  - 1 GB of memory



UNIVERSITY OF OSLO

Experiment setup: definitions

1. *Network load* is specified by the number of observed bits per second on a link.
2. The *relative throughput* of node N, denoted RT_N, is the relation between the network load node N receives and the network load it manages to handle.

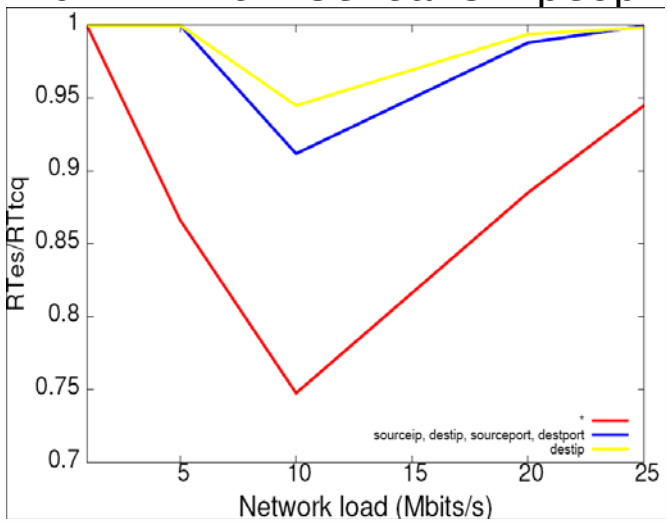


$RT_N = 0.5$

UNIVERSITY OF OSLO

TCQ load shedder correctness:

`SELECT X FROM streams.iptcp`



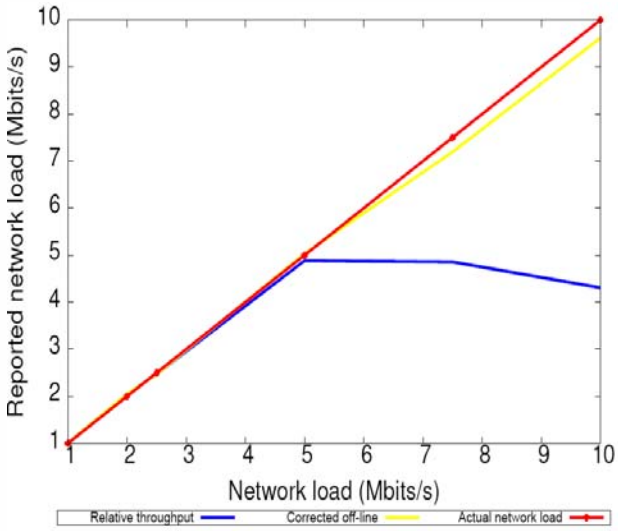
UNIVERSITY OF OSLO

Average Load of Packets and Network Load

```
SELECT
  COUNT(*)/60, AVG(s.totalLength)*8
FROM
  streams.iptcp s [RANGE BY '1 minute'
  SLIDE BY '1 second']
```

UNIVERSITY OF OSLO

Average # of Packets and Network Load



UNIVERSITY OF OSLO

Conclusion and future work

- TelegraphCQ uses a simple and declarative query language
- TelegraphCQ only manages low network loads
  - Starts dropping tuples at 5 Mbits/s
- TelegraphCQ seems to report higher relative throughput than what is actually reached

- Currently, we run tasks on the Borealis stream processing engine
- We will extend the packet-to-tuple translator with more sophisticated load shedding techniques

- Questions?