UiO : University of Oslo

Shared Bottleneck Detection for Coupled Congestion Control for RTP Media Update (draft-ietf-rmcat-sbd-01)

David Hayes (UiO)
Simone Ferlin (SRL) and Michael Welzl (UiO)

[simula . research laboratory]

RITE
REDUCING INTERNET TRANSPORT LATENCY

SEVENTH FRAMEWORK PROGRAMME

Mechanism based on Summary Statistics

Why summary statistics?

- To limit feedback from receivers
- To deal with noise
- To deal with differing path lags

Mechanism based on Summary Statistics

Why summary statistics?

- To limit feedback from receivers
- To deal with noise
- To deal with differing path lags

Statistics Used

- a measure of delay variability (**var_est**)
- a measure of delay skewness (**skew_est**)
- a measure of delay oscillation (**freq_est**)
- a measure of packet loss (**pkt_loss**), a supplementary measure.
- *not a closed list*

Overview of work so far

The mechanism has been demonstrated using:

- Simulation experiments with multiple hops, changing bottlenecks, and realistic background traffic.
- Real network tests over the Internet and 3G mobile using NORNET (https://www.nntb.no/)
- Robustness tweaks

Publication

D. A. Hayes, S. Ferlin, and M. Welzl. Practical passive shared bottleneck detection using shape summary statistics.
In *Proc. of the IEEE Local Computer Networks (LCN)*, pages 150–158, Sept. 2014.
URL http://dx.doi.org/10.1109/LCN.2014.6925767

Key changes in WG-01

Revisions

- Minor terminology improvements
- Moved unbiased skew section to replace skew estimate section
- Removed clock skew estimation text (An improved version in the next revision).

Additions

- Description of key parameters and the influence they have on performance
- A more robust variability estimator

These are minor changes that can improve the performance in certain circumstances.

PDV as a variability estimator

PDV

- Simple and generally good
- Relies on either max_T(OWD) or min_T(OWD)
- During congestion (ie when there is a bottleneck)
  - max_T(OWD) is well sampled
  - min_T(OWD) is not well sampled (so not recommended)
- max_T(OWD) is affected by noise along the path and OS noise.
- We have found it works well, except when there are:
  - orders of magnitude differences in flow send rates combined with light congestion (maximum is harder to sample)

A more robust variability estimator

MAD — Mean Absolute Deviation

var_base_T = sum_T (| OWD – E_T(OWD) |)

where E_T(OWD) is from the previous T

var_est = MAD_MT = sum_MT (var_base_T) /num_MT(OWD)

A more robust variability estimator

MAD — Mean Absolute Deviation

$$var\_base\_T = sum\_T\,(\,|\,OWD - E\_T(OWD)\,|\,)$$

where E_T(OWD) is from the previous T

$$var\_est = MAD\_MT = sum\_MT\,(var\_base\_T)\,/num\_MT(OWD)$$

Qualities

- less affected by path and OS noise
- less affected by extreme values
- less affected by orders of magnitude different flow rates
- still quite simple to calculate

Parameters

- grouping threshold p_mad=0.1 (less noisy so test can be tighter)
- for freq_est p_v=0.7 (MAD is a smaller number then PDV)

Conclusions and plans

- Short paper on online clock skew estimation
- Short paper on online clustering
- Journal
  - algorithm refinements
  - quantitative tests
- Define sender receiver interaction
- Evaluate the effect of time resolution
- Extend tests to wifi scenarios

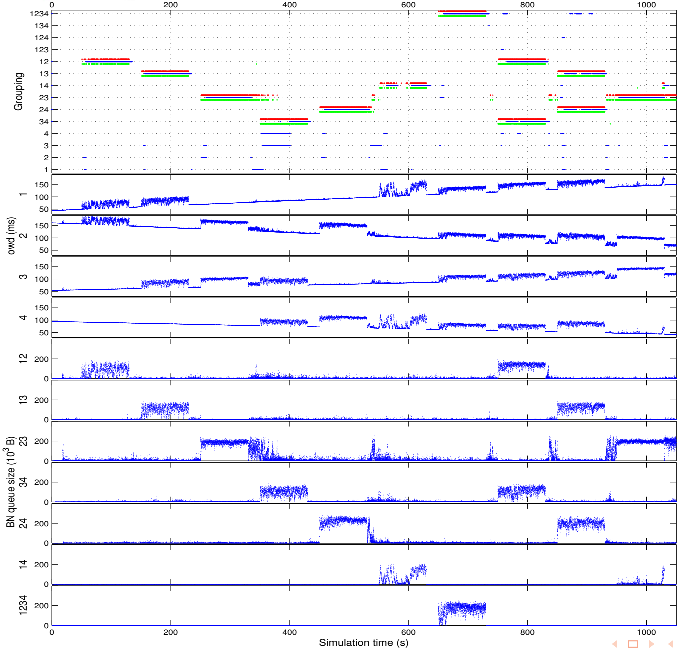
Acknowledgements

The authors are funded by the European Community under its Seventh Framework Programme through the Reducing Internet Transport Latency (RITE) project (ICT-317700). The views expressed are solely those of the authors.

UiO : University of Oslo

Extra slides

(Some experiments looking at noisy clock skewed scenarios)

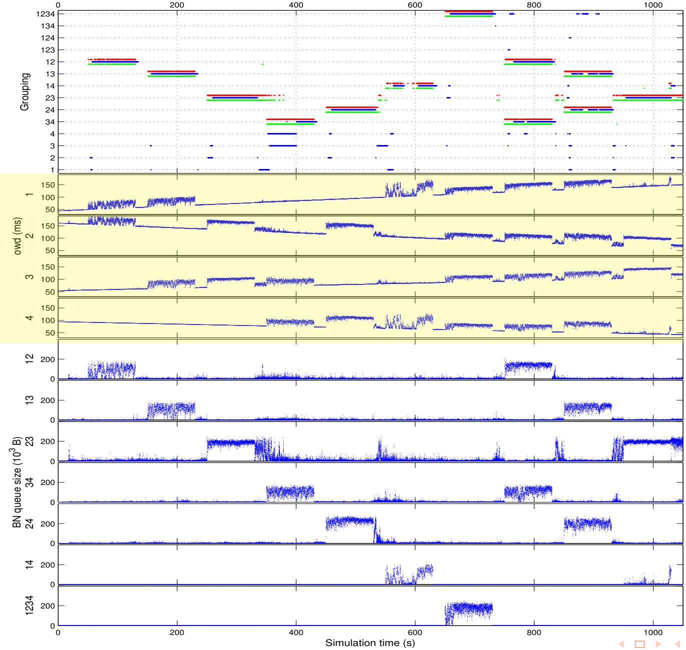


threshold based clustering

- high shared noise
- high clock skew

Subsampled:

- Queue sizes (1:200)
- OWDs (1:10)



threshold based clustering

- high shared noise
- high clock skew

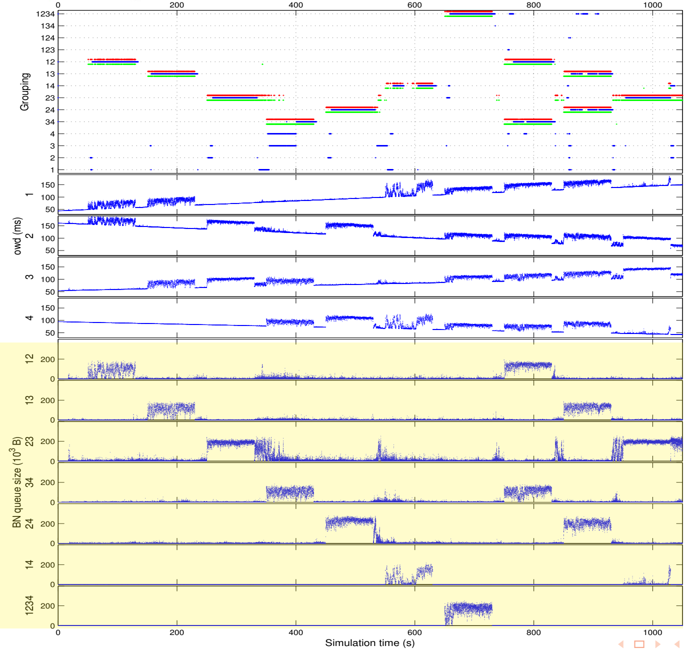
Subsampled:

- Queue sizes (1:200)
- OWDs (1:10)

Clock skews:

100ppm,-100ppm,

50ppm,-50ppm



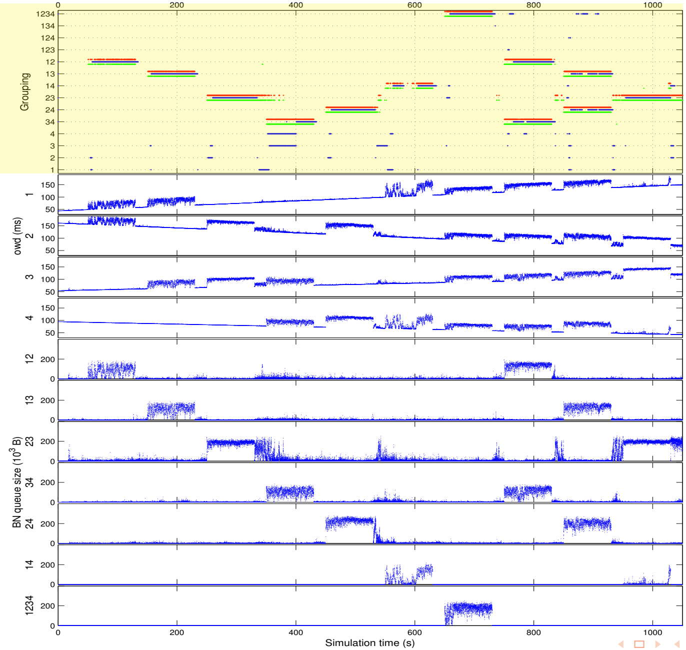
threshold based clustering

- high shared noise
- high clock skew

Subsampled:

- Queue sizes (1:200)
- OWDs (1:10)

Bottleneck queue sizes



threshold based clustering

- high shared noise
- high clock skew

Subsampled:

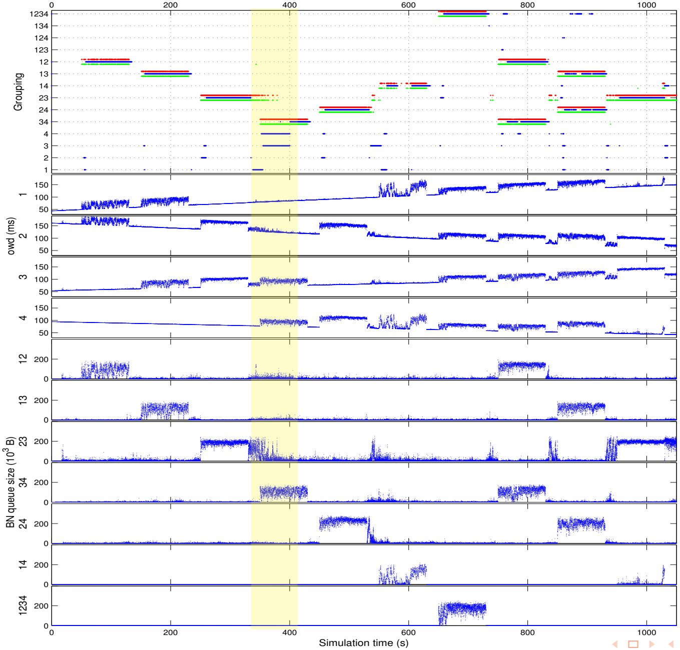
- Queue sizes (1:200)
- OWDs (1:10)

+ Average Q > 50000

+ Does not empty in T

+ SBD decisions

Grouping decisions



threshold based clustering

- high shared noise
- high clock skew

Subsampled:

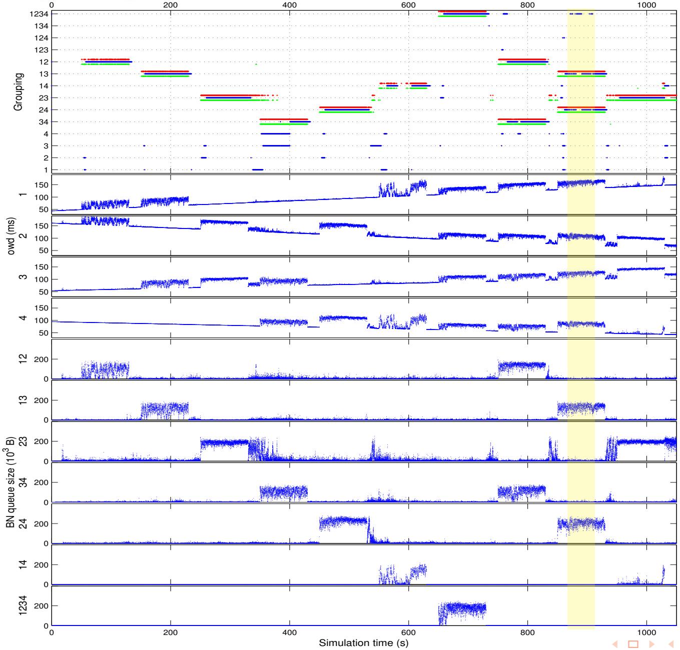
- Queue sizes (1:200)
- OWDs (1:10)

+ Average Q > 50000

+ Does not empty in T

+ SBD decisions

Noise from a bottleneck 2&3 destroy statistical commonality between flows 3&4



threshold based clustering

- high shared noise
- high clock skew

Subsampled:

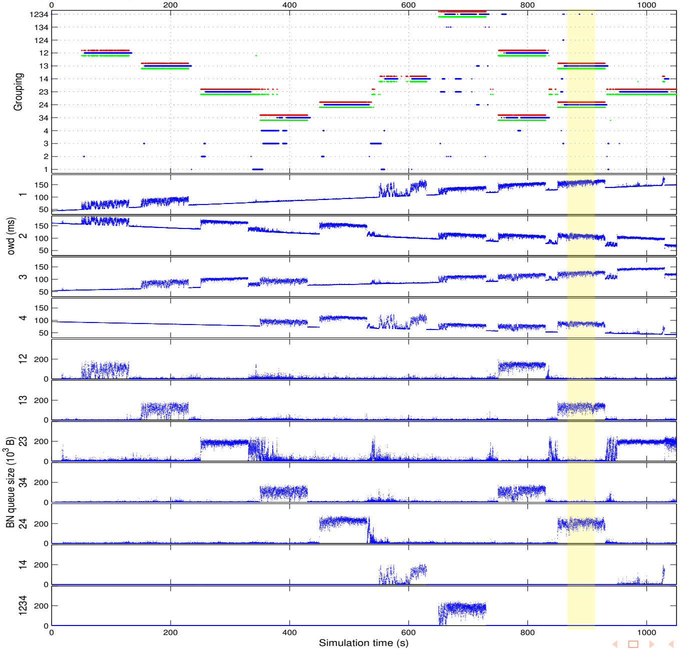
- Queue sizes (1:200)
- OWDs (1:10)

+ Average Q > 50000

+ Does not empty in T

+ SBD decisions

Flows on both bottlenecks are statistically close (within thresholds)



graph based clustering

- high shared noise
- high clock skew

Subsampled:

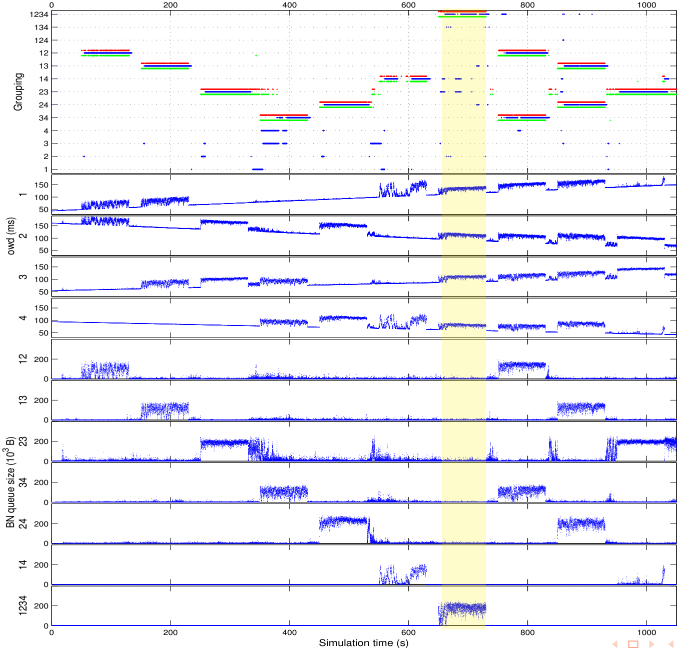
- Queue sizes (1:200)
- OWDs (1:10)

+ Average Q > 50000

+ Does not empty in T

+ SBD decisions

Graph based clustering can more finely group flows



graph based clustering

- ▶ high shared noise
- ▶ high clock skew

Subsampled:

- ○ Queue sizes (1:200)
- ○ OWDs (1:10)

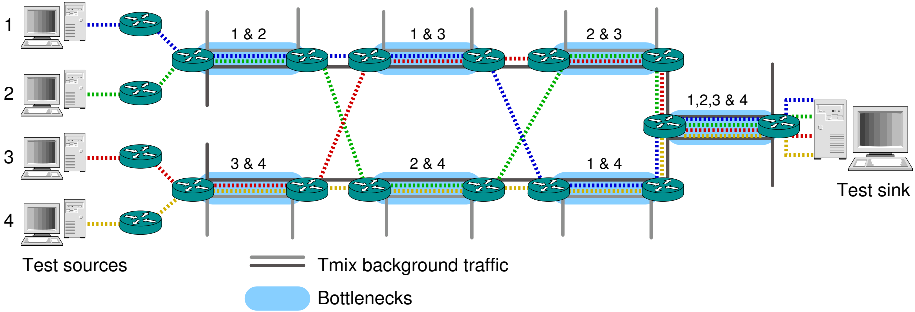
+ Average Q > 50000

+ Does not empty in T

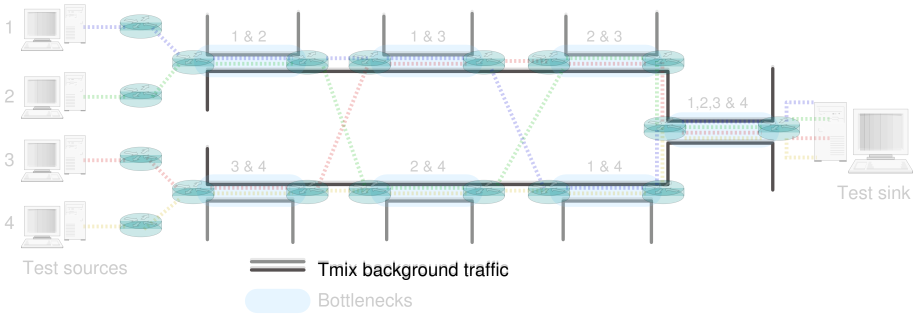
+ SBD decisions

- ○ Subgroups due to other shared noise.
- ○ More an artefact of the simulation setup.
  - ○ flows sharing in different combinations

Simulation setup

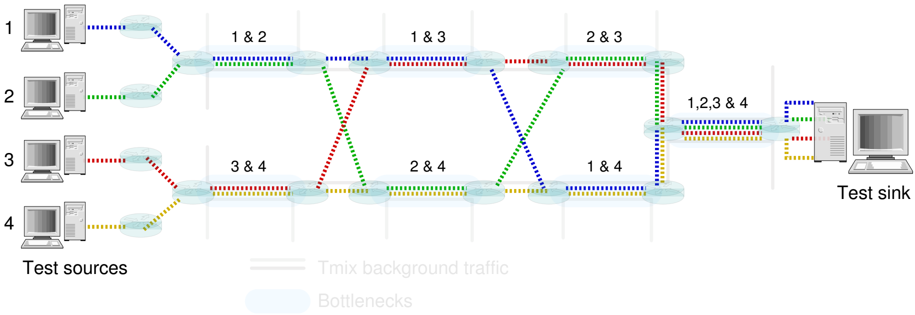


Simulation setup



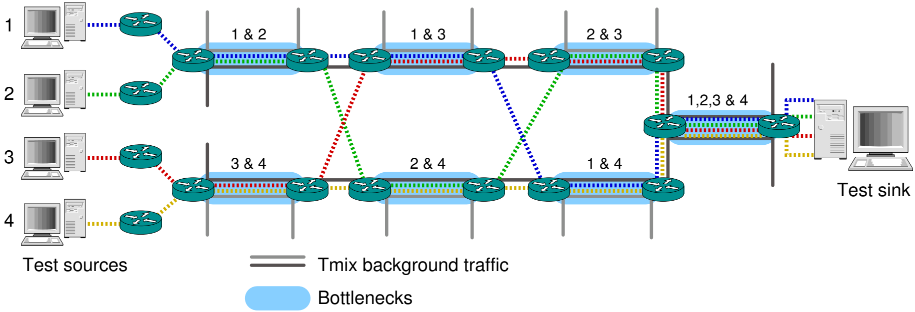
- Background traffic based on real traffic traces
  - > 90%

Simulation setup



- Background traffic based on real traffic traces
  - > 90%
- Flows 1 & 2 send at twice the rate of 3 & 4

Simulation setup



- Background traffic based on real traffic traces
  - > 90%
- Flows 1 & 2 send at twice the rate of 3 & 4
- Various combinations of bottlenecks activated