



UiO • University of Oslo

Shared Bottleneck Detection evaluations with NorNet

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

[**simula** . research laboratory]



REDUCING INTERNET TRANSPORT LATENCY



Rough Outline

Shared Bottleneck Detection — David

- ▶ Problem and motivation
- ▶ Our algorithm
- ▶ Simulation tests

NorNet Experiments — Simone

- ▶ Experiment setup
- ▶ Traffic generation
- ▶ Results
- ▶ Lessons learned

Conclusions and further work

D. A. Hayes, S. Ferlin, and M. Welzl, “Practical passive shared bottleneck detection using shape summary statistics,” in Proceedings of IEEE LCN, Sep. 2014, to appear.

Flows that Share Bottlenecks

Flow 1

Flow 2

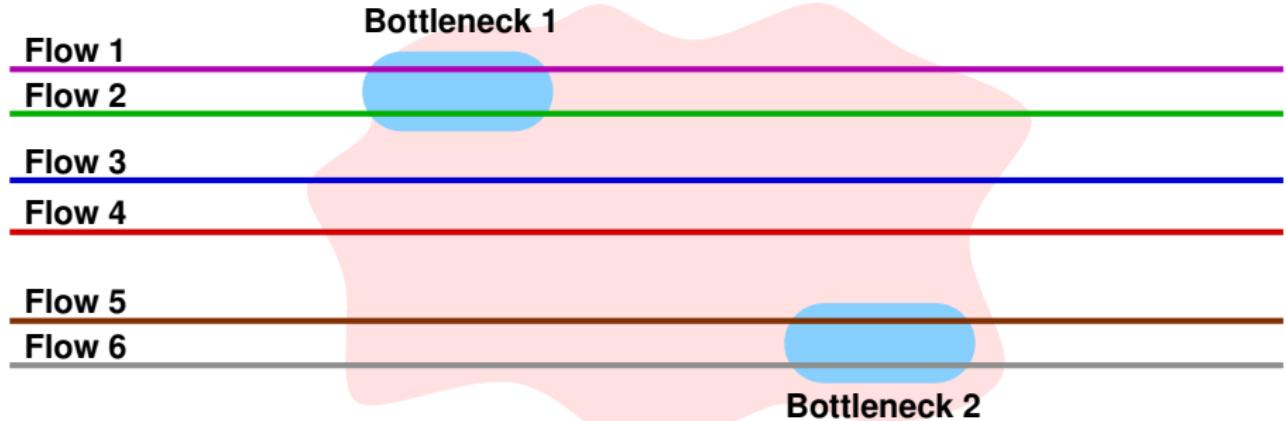
Flow 3

Flow 4

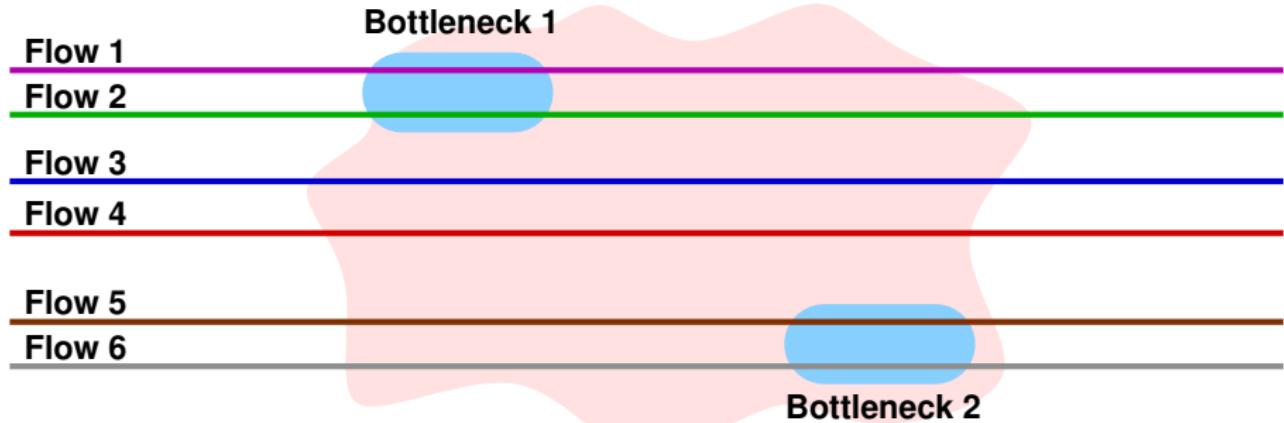
Flow 5

Flow 6

Flows that Share Bottlenecks



Flows that Share Bottlenecks



Why detect bottlenecks?

- ▶ Endpoints can manage competing flows for:
 - ▶ priority
 - ▶ latency
 - ▶ fairness, etc
- ▶ Applications include:
 - ▶ MPTCP
 - ▶ SCTP
 - ▶ RTCWEB, etc.

Shared Bottleneck Detection (SBD)

What does it rely on?

- ▶ flows that share a bottleneck are similar in a measurable way

Shared Bottleneck Detection (SBD)

What does it rely on?

- ▶ flows that share a bottleneck are similar in a measurable way

Why is it hard?

- ▶ delay and loss measurements are noisy
 - ▶ noise from rest of the path
 - ▶ noise at the bottleneck
 - ▶ each packet sees a different queueing delay
- ▶ different path delays
 - ▶ time correlations are lost

Shared Bottleneck Detection (SBD)

What does it rely on?

- ▶ flows that share a bottleneck are similar in a measurable way

Why is it hard?

- ▶ delay and loss measurements are noisy
 - ▶ noise from rest of the path
 - ▶ noise at the bottleneck
 - ▶ each packet sees a different queueing delay
- ▶ different path delays
 - ▶ time correlations are lost

It *has* been regarded as too unreliable and computationally intensive for practical use.

Summary

We argue

Estimates of higher order summary statistics are sufficient to detect shared bottlenecks among small to moderate numbers of flows

- ▶ Variance, Skewness, and a frequency measure

Summary

We argue

Estimates of higher order summary statistics are sufficient to detect shared bottlenecks among small to moderate numbers of flows

- ▶ Variance, Skewness, and a frequency measure

We show

Shared Bottleneck Detection (SBD) working in:

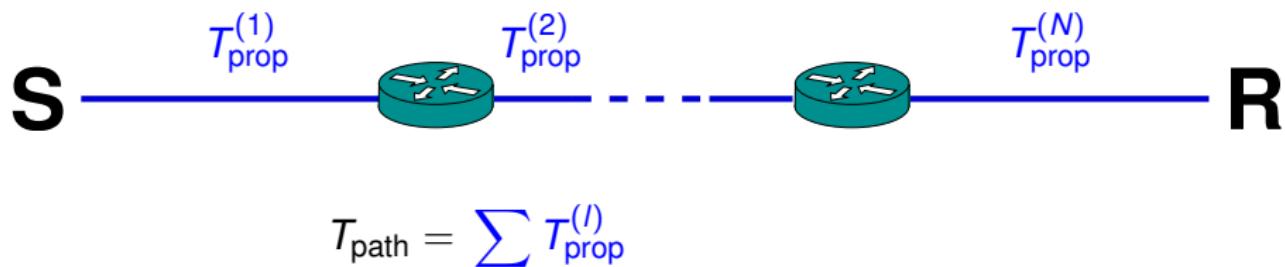
- ▶ a difficult (for SBD) simulation scenario,
- ▶ Nornet tests

Path Delay



$$T_{\text{path}} =$$

Path Delay



Path Delay



$$T_{\text{path}} = \sum T_{\text{prop}}^{(l)} + \sum T_{\text{queue}}^{(l)}$$

Path Delay



$$T_{\text{path}} = \sum T_{\text{prop}}^{(l)} + \sum T_{\text{queue}}^{(l)} + \sum T_{\text{misc}}^{(l)}$$

Path Delay



$$T_{\text{path}} = \sum T_{\text{prop}}^{(l)} + \sum T_{\text{queue}}^{(l)} + \sum T_{\text{misc}}^{(l)}$$

$$T_{\text{b_signal}} = T_{\text{queue}}^{(b)} + T_{\text{misc}}^{(b)}$$

Path Delay



$$T_{\text{path}} = \sum T_{\text{prop}}^{(l)} + \sum T_{\text{queue}}^{(l)} + \sum T_{\text{misc}}^{(l)}$$

$$T_{\text{b_signal}} = T_{\text{queue}}^{(b)} + T_{\text{misc}}^{(b)}$$

$$T_{\text{path_noise}} = \sum T_{\text{prop}}^{(l)} + \sum_{l \neq b} T_{\text{queue}}^{(l)} + \sum_{l \neq b} T_{\text{misc}}^{(l)}$$

Path Delay



$$T_{\text{path}} = \sum T_{\text{prop}}^{(l)} + \sum T_{\text{queue}}^{(l)} + \sum T_{\text{misc}}^{(l)}$$

$$T_{\text{b_signal}} = T_{\text{queue}}^{(b)} + T_{\text{misc}}^{(b)}$$

$$T_{\text{path_noise}} = \sum T_{\text{prop}}^{(l)} + \sum_{l \neq b} T_{\text{queue}}^{(l)} + \sum_{l \neq b} T_{\text{misc}}^{(l)}$$

$$T_{\text{RTT_noise}} \approx 2 T_{\text{path_noise}}$$

Summary Statistics

One Way Delay (OWD) metrics

- ▶ variance
- ▶ skewness
- ▶ oscillation

Packet loss is also used when it can be.

Summary Statistics

One Way Delay (OWD) metrics

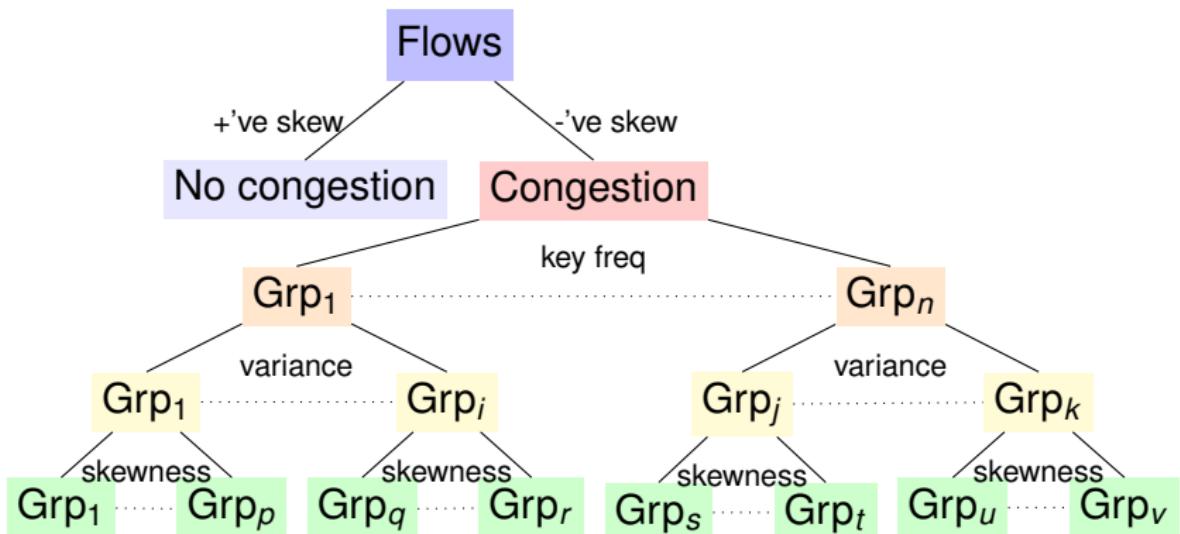
- ▶ variance
- ▶ skewness
- ▶ oscillation

Packet loss is also used when it can be.

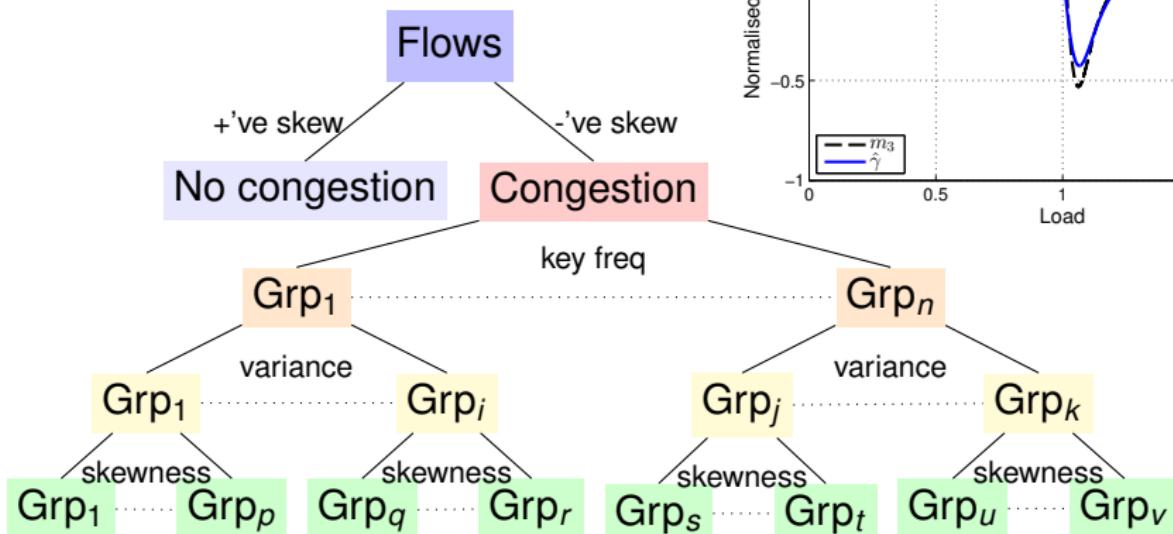
Metric Estimates

- ▶ Packet Delay Variation (PDV) RFC 5481
- ▶ counter based skewness estimate
- ▶ normalised significant mean crossings

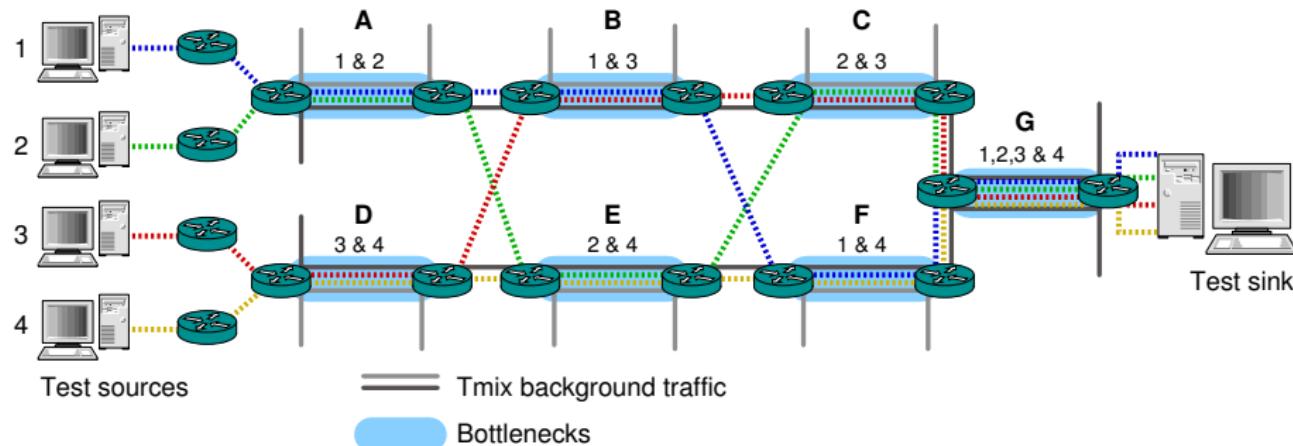
Simple grouping overview



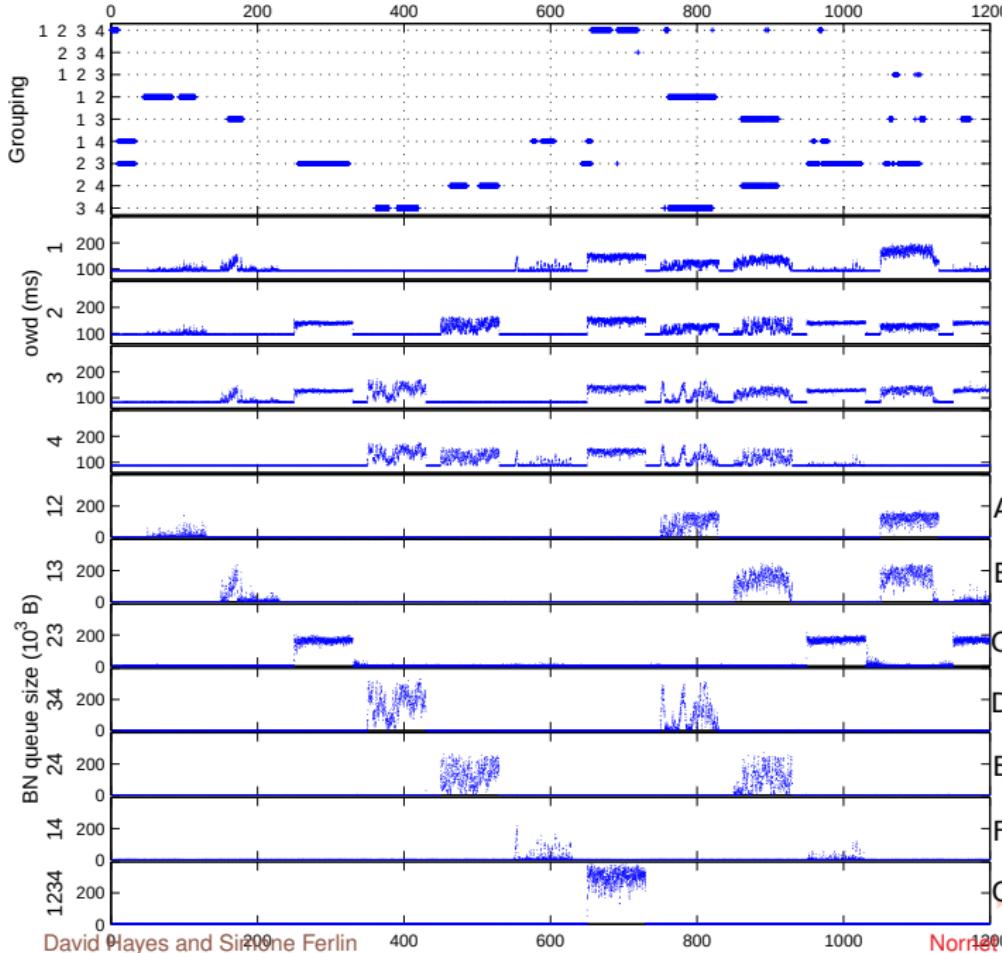
Simple grouping overview



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 over simulation.



Simulation results

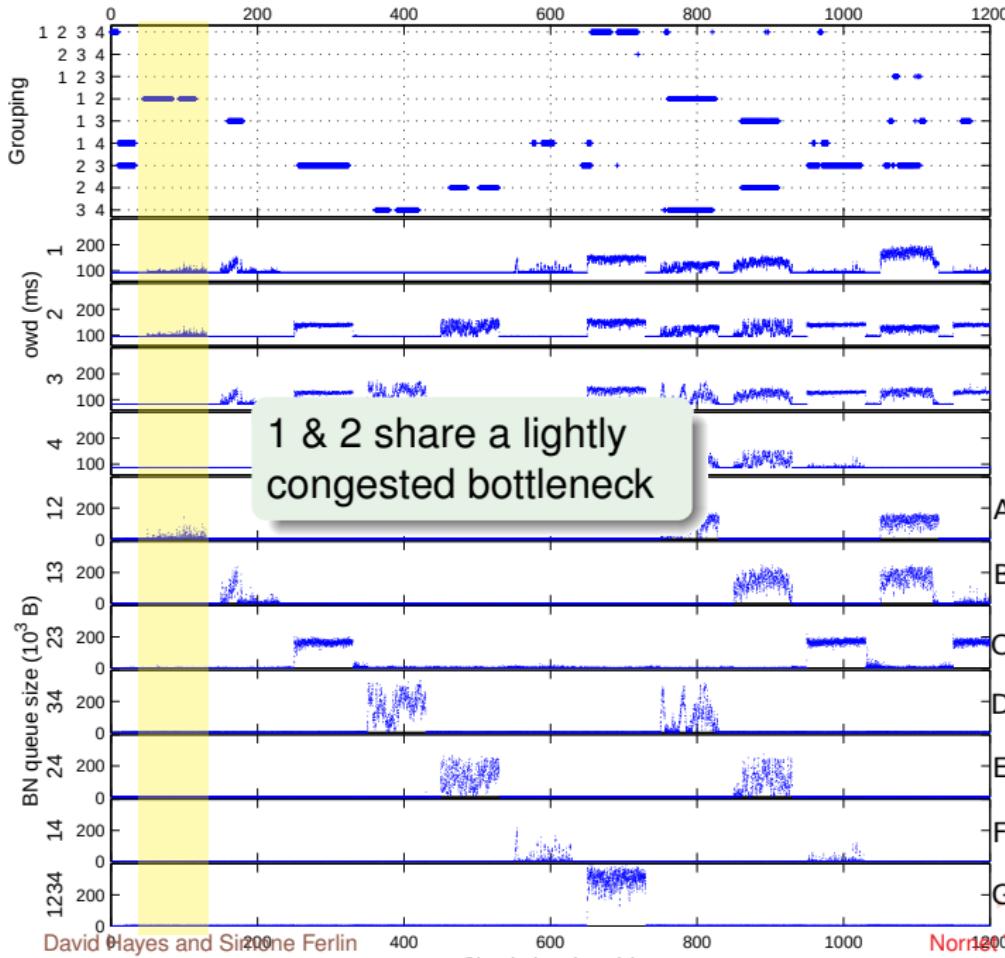
OWD –
One Way Delay

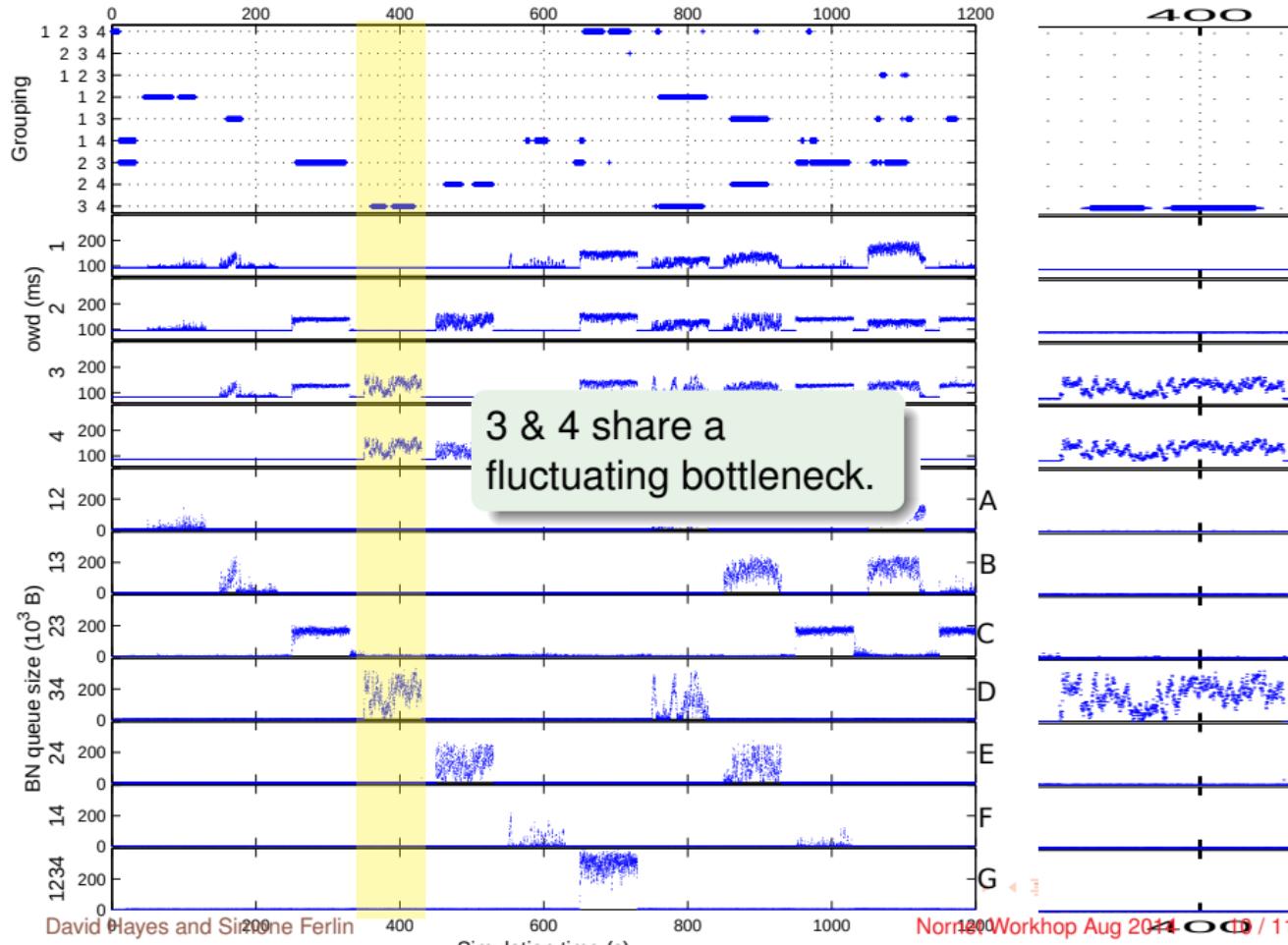
BN – BottleNeck

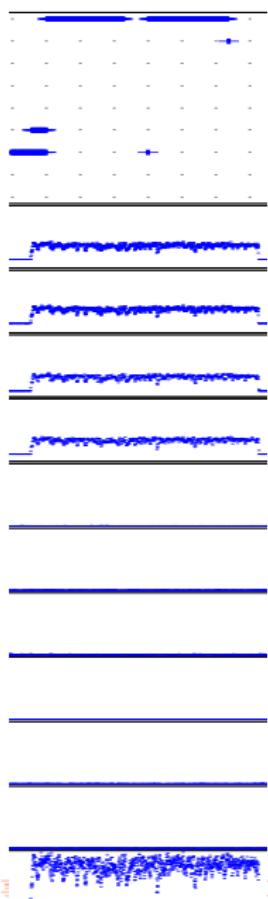
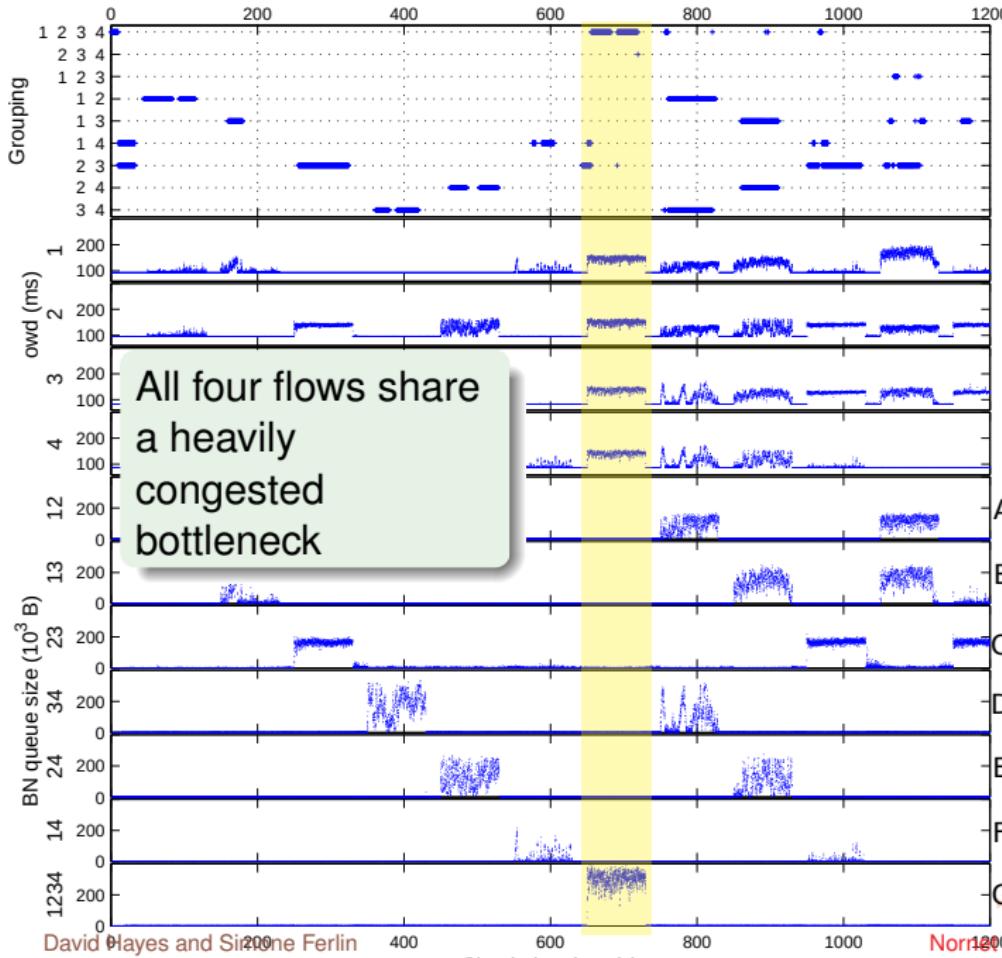
link delay
variation std 2.5.

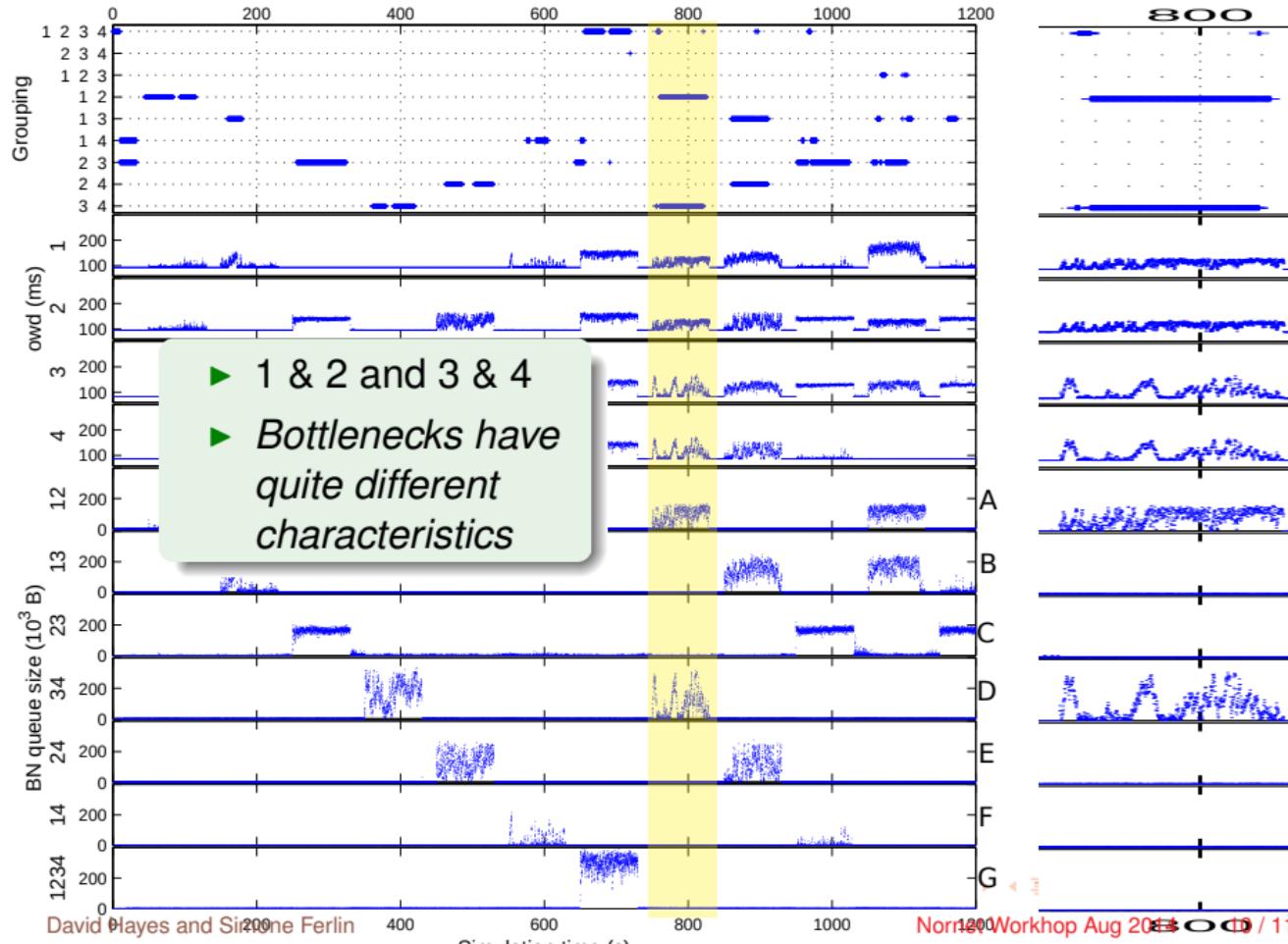
sub-sampled
queue sizes
(1:350)

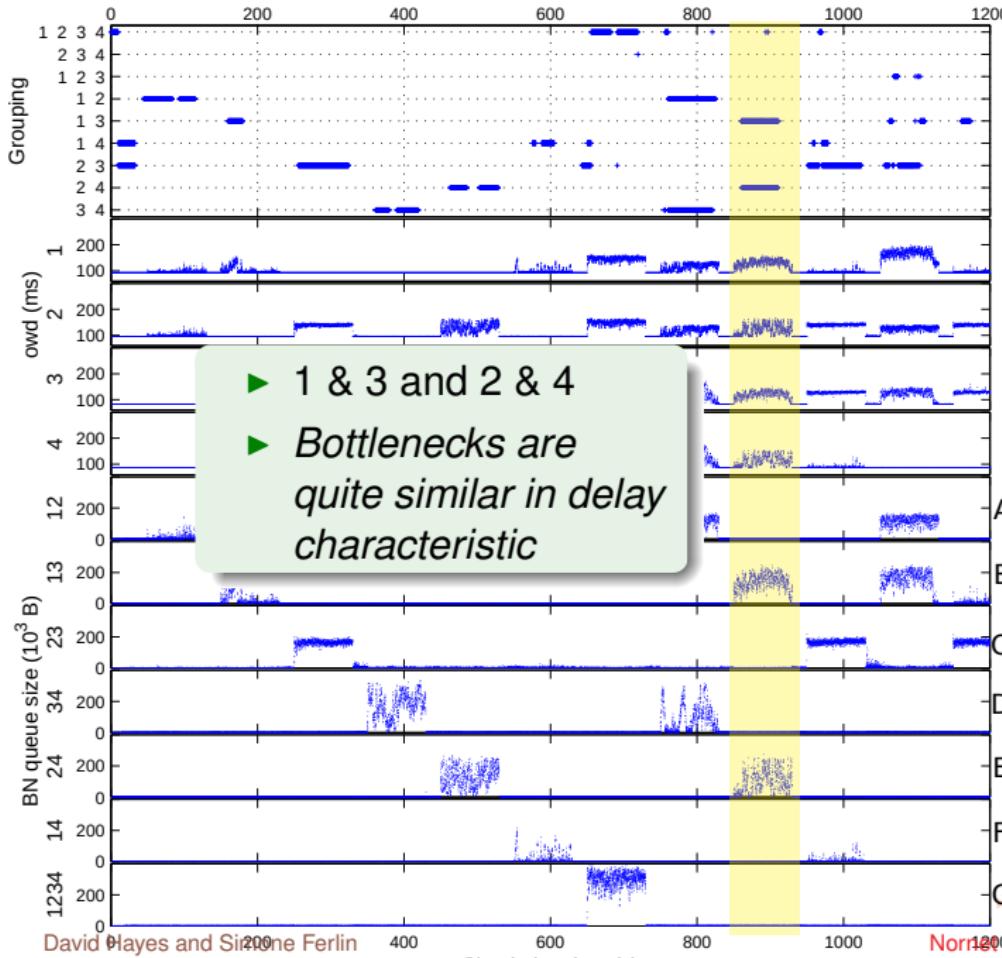
sub-sampled
OWDs (1:20)











NORNET tests

Simone



UiO • University of Oslo

Extra slides



Related Work

Active and passive techniques based on:

- ▶ base metrics:
 - ▶ throughput, loss, delay
- ▶ Similarity metrics:
 - ▶ cross correlation
 - ▶ clustering
- ▶ Noise handling
 - ▶ wavelet de-noising
 - ▶ Singular value decomposition
 - ▶ averaging

Related Work

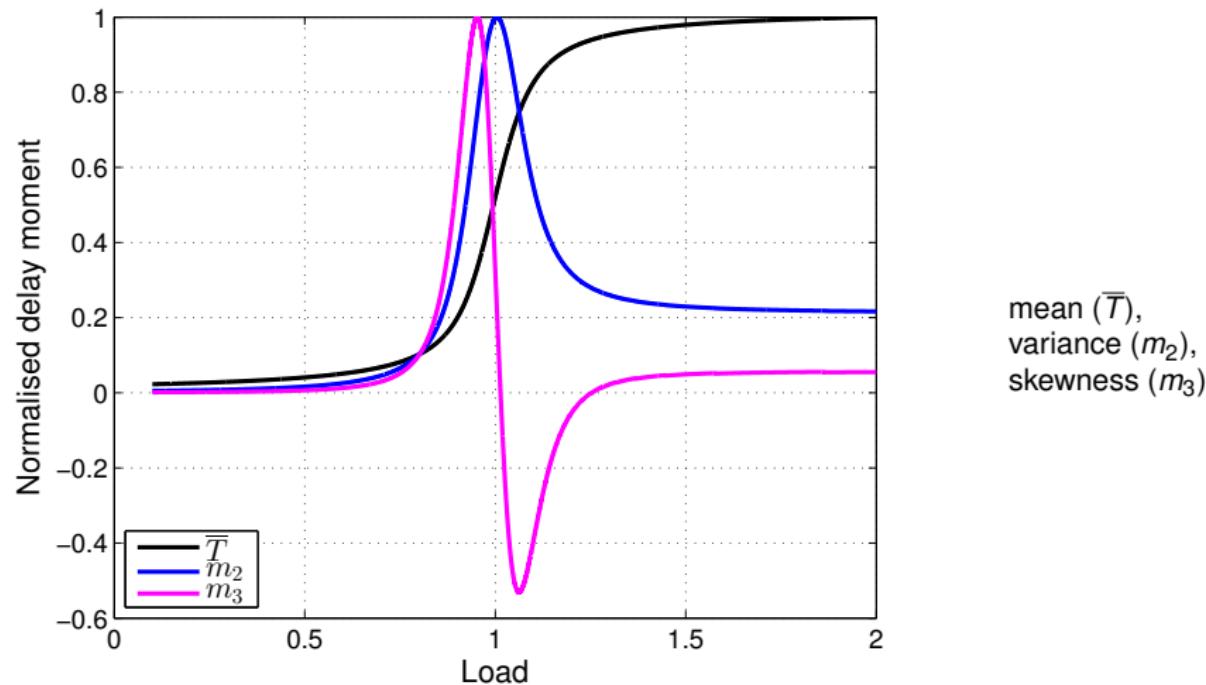
Active and passive techniques based on:

- ▶ base metrics:
 - ▶ throughput, loss, delay
- ▶ Similarity metrics:
 - ▶ cross correlation
 - ▶ clustering
- ▶ Noise handling
 - ▶ wavelet de-noising
 - ▶ Singular value decomposition
 - ▶ averaging

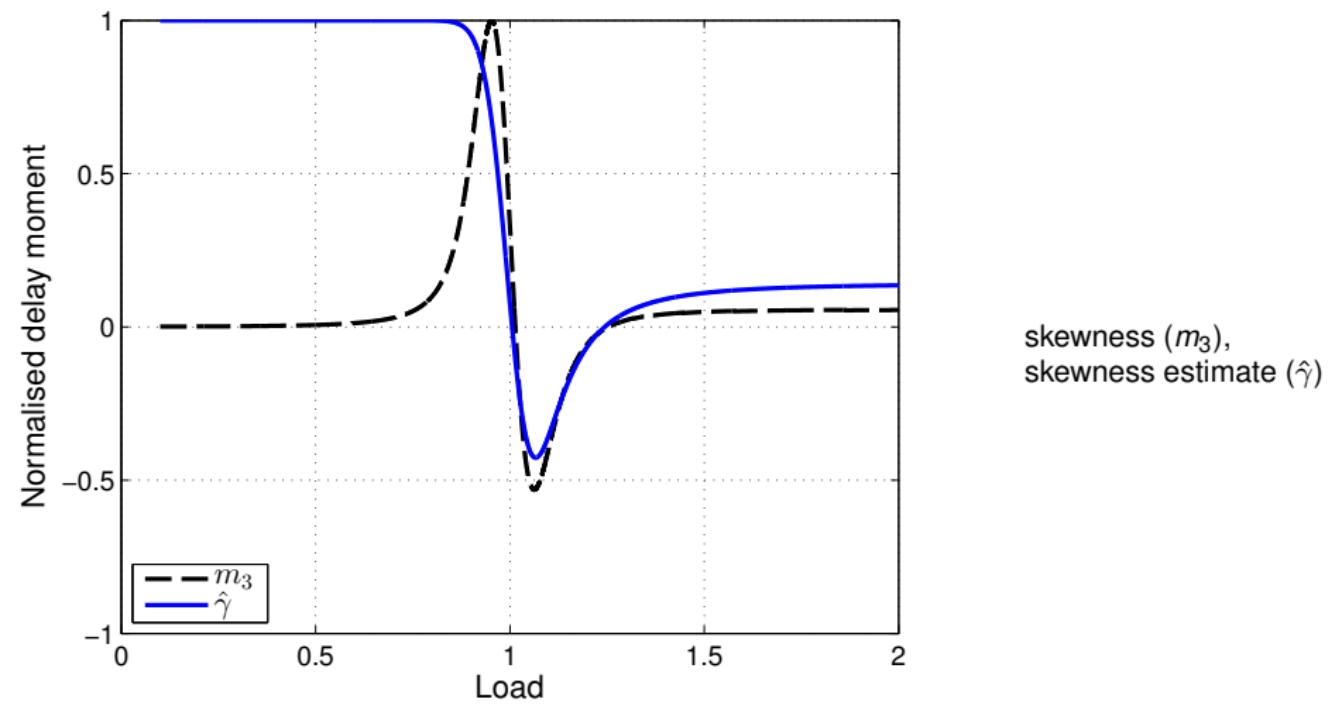
Common issues:

- ▶ testing scenarios
- ▶ ground truth
- ▶ dynamic bottlenecks
- ▶ practicality

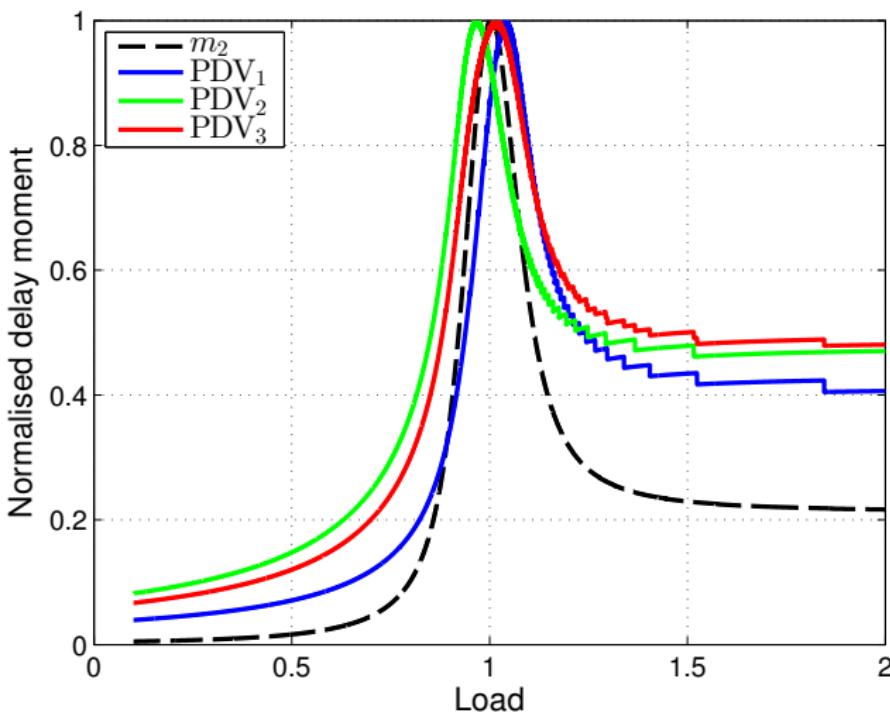
Time domain OWD summary statistics



Practical estimation of skewness



Practical estimation of variance



variance (m_2)

RFC 5481

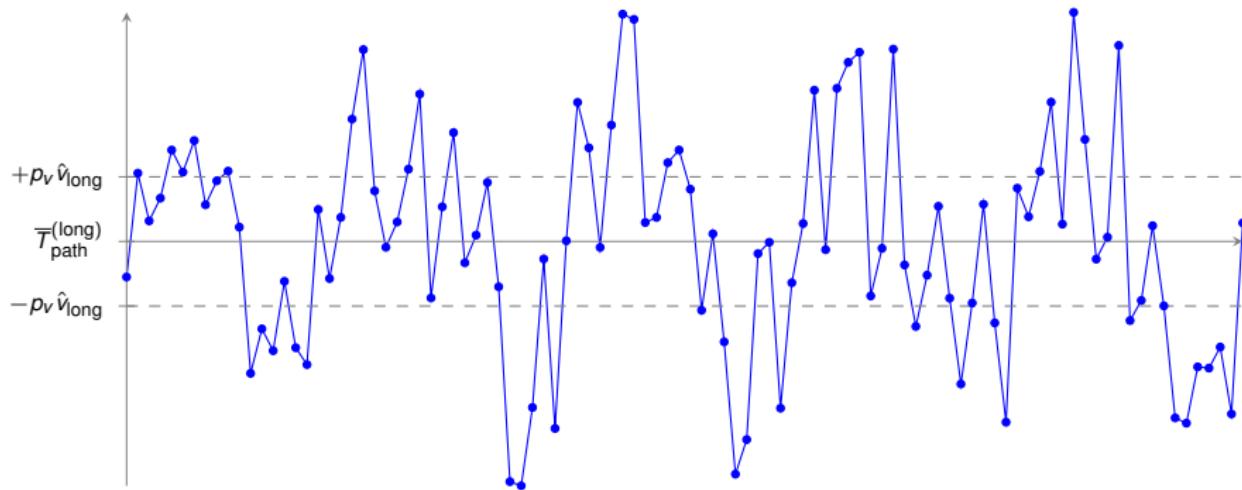
Packet Delay Variation (PDV):

$$\text{PDV}_1 = \bar{T}_{\text{path}} - \min(T_{\text{path}})$$

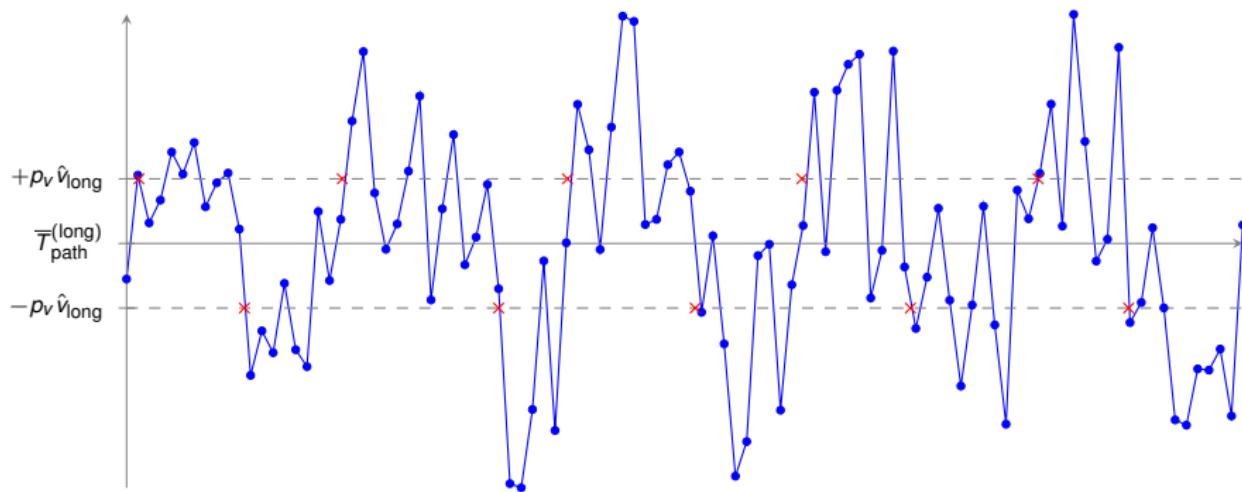
$$\text{PDV}_2 = \max(T_{\text{path}}) - \bar{T}_{\text{path}}$$

$$\text{PDV}_3 = \max(T_{\text{path}}) - \min(T_{\text{path}})$$

Practical estimation of key frequency \hat{f}



Practical estimation of key frequency \hat{f}



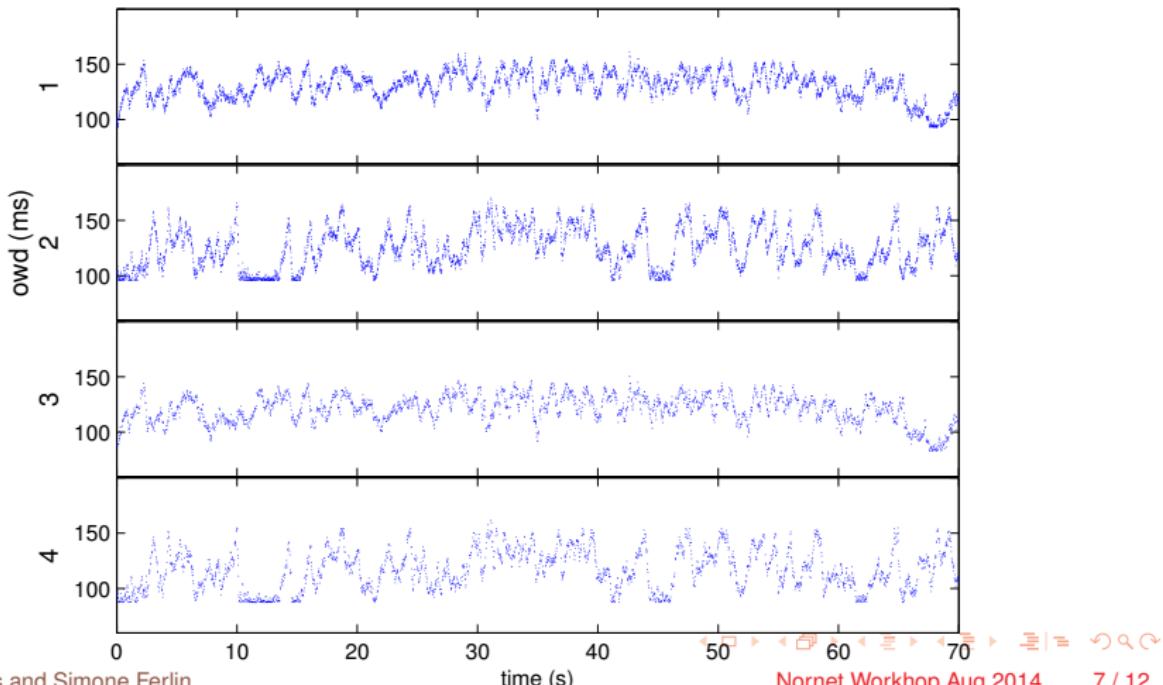
Normalised key bottleneck frequency: $\hat{f} = 10/100 = 0.1$

An observation

- ▶ Plots of a flows' packet delays are:
 - ▶ noticeably different when they traverse different bottlenecks
 - ▶ noticeably similar when they traverse the same bottleneck.

An observation

- ▶ Plots of a flows' packet delays are:
 - ▶ noticeably different when they traverse different bottlenecks
 - ▶ noticeably similar when they traverse the same bottleneck.



Problem

- ▶ Flows traversing different paths through a network may share a common congested link — a bottleneck
- ▶ Detecting which flows share a bottleneck and coupling their congestion control can provide performance advantages.

SBD design objectives

- ▶ Reliable
- ▶ Practical (not CPU nor network intensive)
- ▶ Small numbers of bottlenecks (< 10)
- ▶ Timely stable bottleneck detection (< 10 s)

Notes on results

- ▶ Decisions made every 300 ms, but based on 6 – 15 s statistics.
- ▶ Decision “points” are large for legibility, but it can tend to magnify errors.
- ▶ Results illustrates what can and can’t be done.

Classic cross correlation techniques

Pairwise flow cross correlation of delay samples

- ▶ delay signal is noisy
 - ▶ filter
- ▶ delay distribution is often skewed
 - ▶ sophisticated filter
- ▶ different path delays
 - ▶ incrementally shift and cross correlate to find lag of maximum correlation.

Dealing with the signal noise

- ▶ Remove half the noise from other links by using OWD instead of RTT
- ▶ Only using difference statistics
 - ▶ removes queueing delay estimate errors due to inaccurate estimate of OWD_{min}
- ▶ Mitigate lag and sample noise by:
 - ▶ relatively large statistic gathering periods
 - ▶ relax thresholds (no need to distinguish between 1000 bottlenecks)
 - ▶ use multiple measures

Working with Coupled Congestion Control

- ▶ Summary statistics are gathered at the receivers
- ▶ Shared bottlenecks to a receiver
 - ▶ Receiver does grouping and sends information to senders
- ▶ Shared bottlenecks from a sender:
 - ▶ Receivers send summary statistics for grouping at sender.
- ▶ Can provide the necessary information for a future multi-sender multi-receiver coupled congestion control.

Who am I?

[**simula**.research laboratory]



Simone Ferlin

PhD Student

Mobile: +4740720702

E-mail: ferlin@simula.no

Affiliations: [Simula School of Research and Innovation AS Networks](#)

Outline

Shared Bottleneck Detection — David

- ▶ Problem and motivation
- ▶ Our algorithm
- ▶ Simulation tests

NorNet Experiments — Simone

- ▶ Shared bottleneck with real experiments
- ▶ Experiments: NorNet setup and traffic generation
- ▶ Results
- ▶ Lessons learned

Shared bottleneck detection (SBD) with real experiments

Why is it hard?

- ▶ Bottleneck “ground truth” cannot be determined with 100 % certainty

Shared bottleneck detection (SBD) with real experiments

Why is it hard?

- ▶ Bottleneck “ground truth” cannot be determined with 100 % certainty

Method:

- ▶ Find thinnest link using STAB^a and traceroute
- ▶ Load thinnest link with other sources to create a known bottleneck

^a<http://spin.rice.edu/Software/STAB>

Shared bottleneck detection (SBD) with real experiments

Why is it hard?

- ▶ Bottleneck “ground truth” cannot be determined with 100 % certainty

Method:

- ▶ Find thinnest link using STAB^a and traceroute
- ▶ Load thinnest link with other sources to create a known bottleneck

^a<http://spin.rice.edu/Software/STAB>

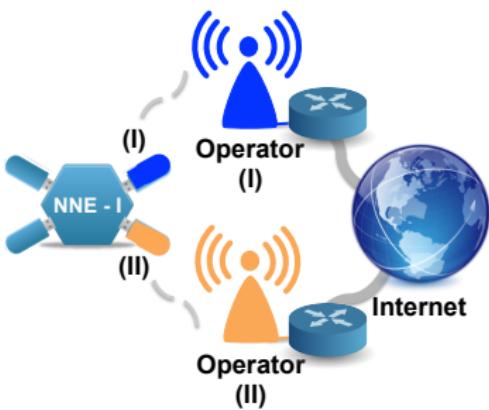
What are we testing?

SBD robustness in unpredictable “real” environments

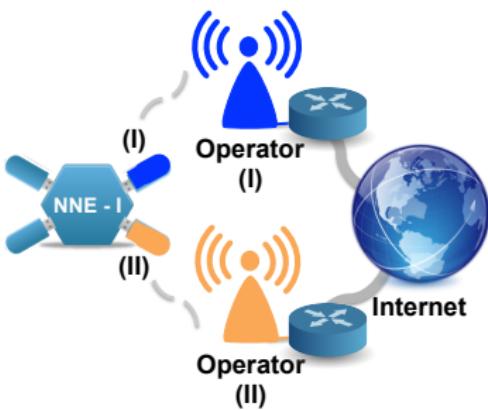
Experiment Setup

NorNet Edge (NNE-I):

- ▶ Operator I: Telenor
- ▶ Operator II: NetCom
- ▶ Separate network



Experiment Setup



NorNet Edge
(NNE-I):

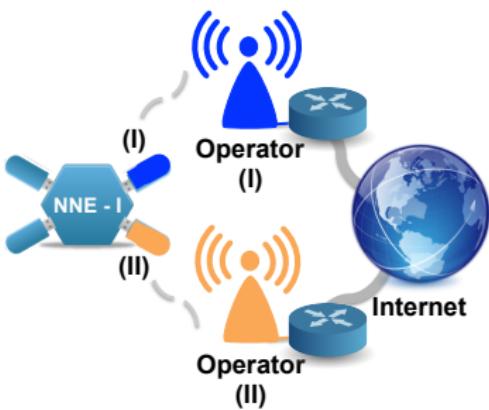
- ▶ Operator I: Telenor
- ▶ Operator II: NetCom
- ▶ Separate network

How we did it:

- ▶ D-ITG (Distributed Internet Traffic Generator)^a

^a<http://traffic.comics.unina.it>

Experiment Setup



NorNet Edge
(NNE-I):

- ▶ Operator I: Telenor
- ▶ Operator II: NetCom
- ▶ Separate network

How we did it:

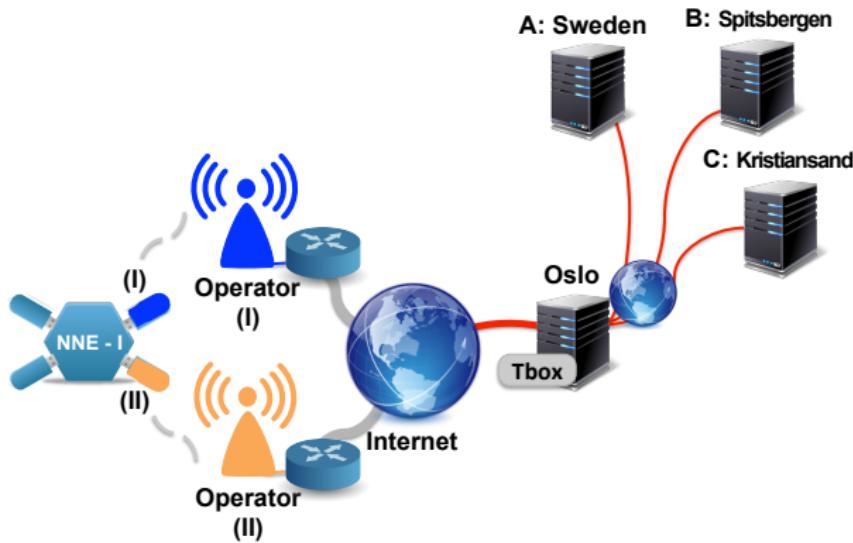
- ▶ D-ITG (Distributed Internet Traffic Generator)^a

^a<http://traffic.comics.unina.it>

Findings:

- ▶ Operator I and II: 3G links with different rates
- ▶ Queue in Operator I appears to have AQM (ongoing study)
 - ▶ Operator II rarely drops packets

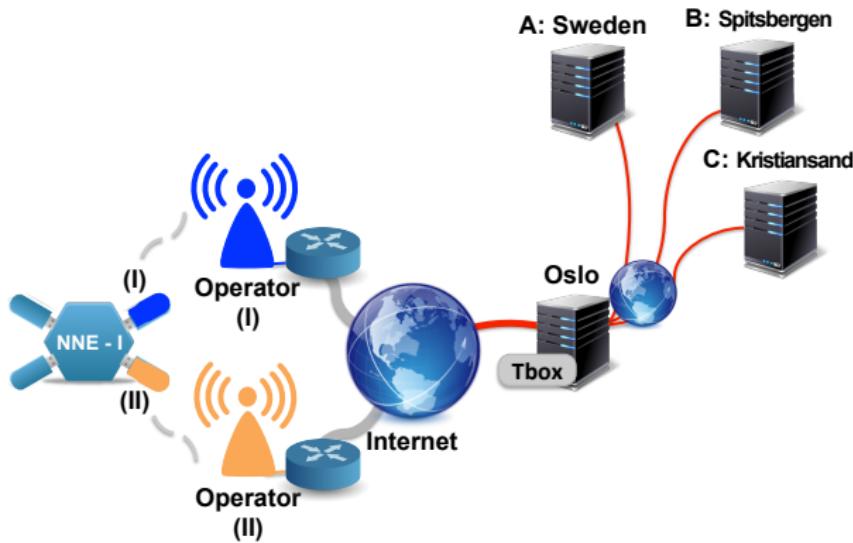
Experiment Setup



NorNet Core: Background Traffic

- ▶ A: Exponential
- ▶ B & C: LRD
(Hurst=0.8) traffic
- ▶ Bottlenecks mostly on 3G link

Experiment Setup



NorNet Core: Background Traffic

- ▶ A: Exponential
- ▶ B & C: LRD (Hurst=0.8) traffic
- ▶ Bottlenecks mostly on 3G link

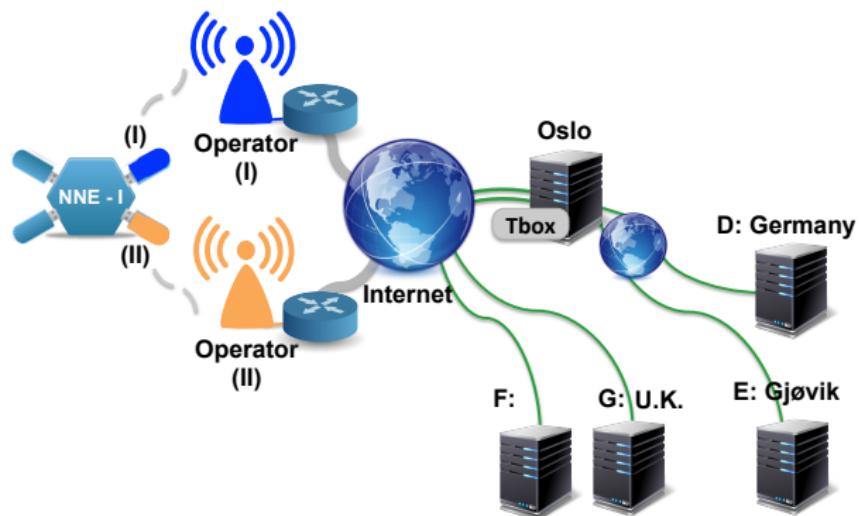
How we did it:

- ▶ A: 1x exponential distributed flow: Mean rate 92 pps (I) and 460 pps (II)
- ▶ B & C: Each 8x exponential distributed flows with Pareto distributed *on* and exponential distributed *off* times: 72 pps (I) and 90 pps (II)
- ▶ Packet sizes are variable with 1000 Byte on average

Experiment Setup

Application Traffic

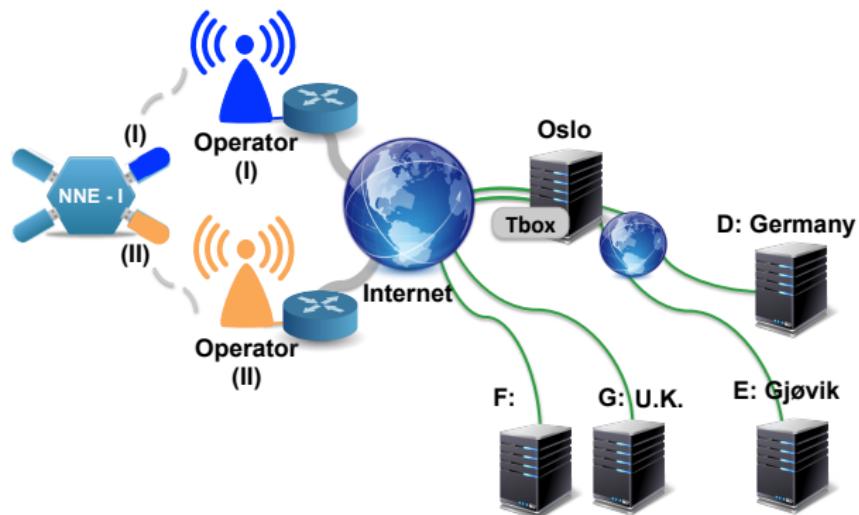
- ▶ D,E, & F:
Exponential
 - ▶ G: CBR



Experiment Setup

Application Traffic

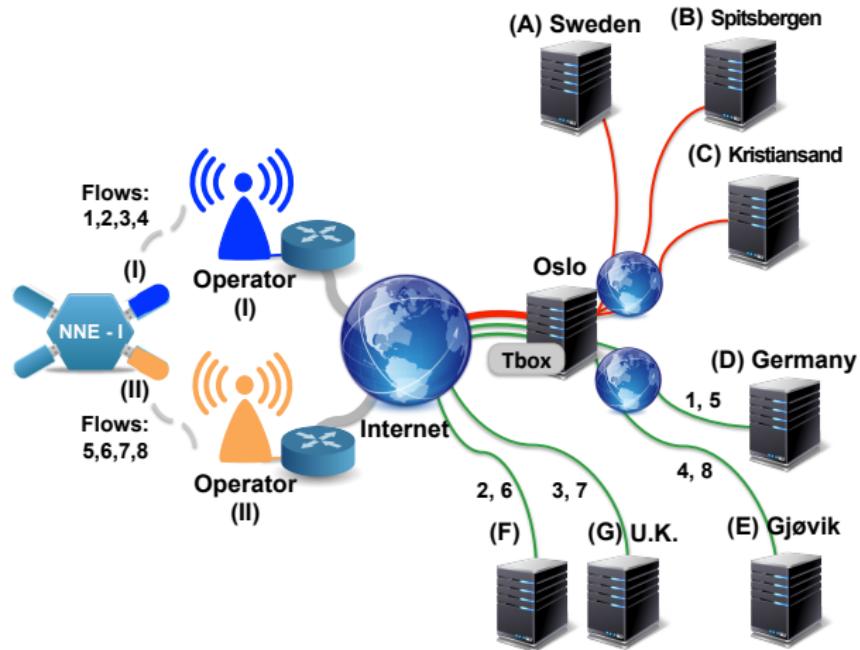
- ▶ D,E, & F: Exponential
- ▶ G: CBR



How we did it?:

- ▶ D, E & F: Exponential distributed flow: 100 pps with 50 B packets
- ▶ G: CBR (Constant Bitrate) of 100 pps with 50 B packets
- ▶ D, E, F & G: Flows without any characteristic that eases grouping

Experiment Setup

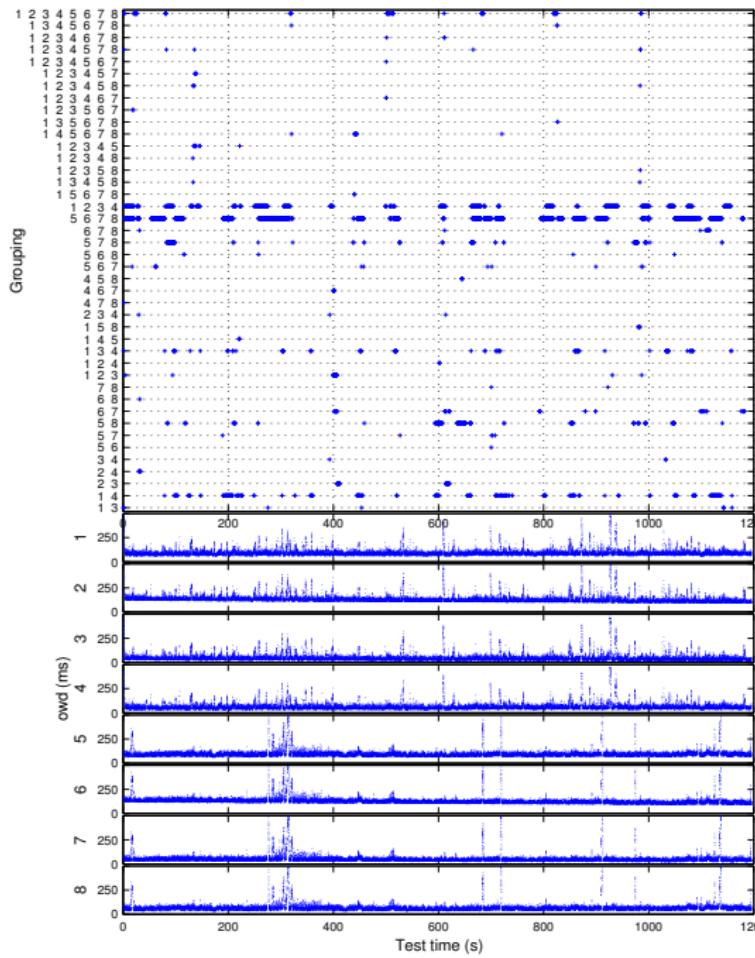


Idea:

- ▶ Create bottleneck with background traffic and detect it with application traffic

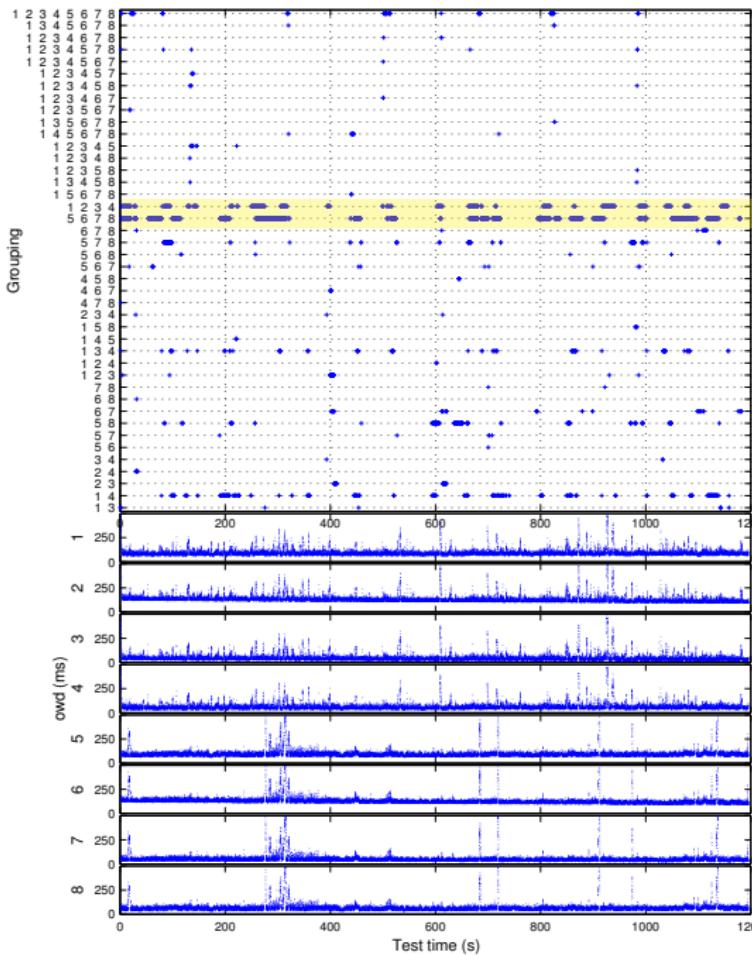
All Together:

- ▶ Background: 90% of traffic
- ▶ Application: minority of traffic, each two flows: one through each 3G link



NorNet results

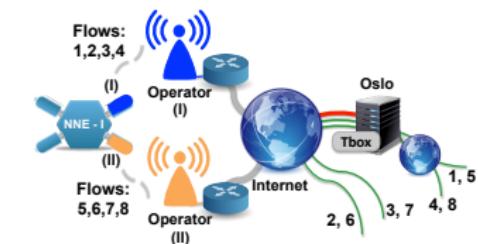
- ▶ Experiment run: 20 min
 - ▶ SBD decision: 350 ms
 - ▶ SBD simulation settings

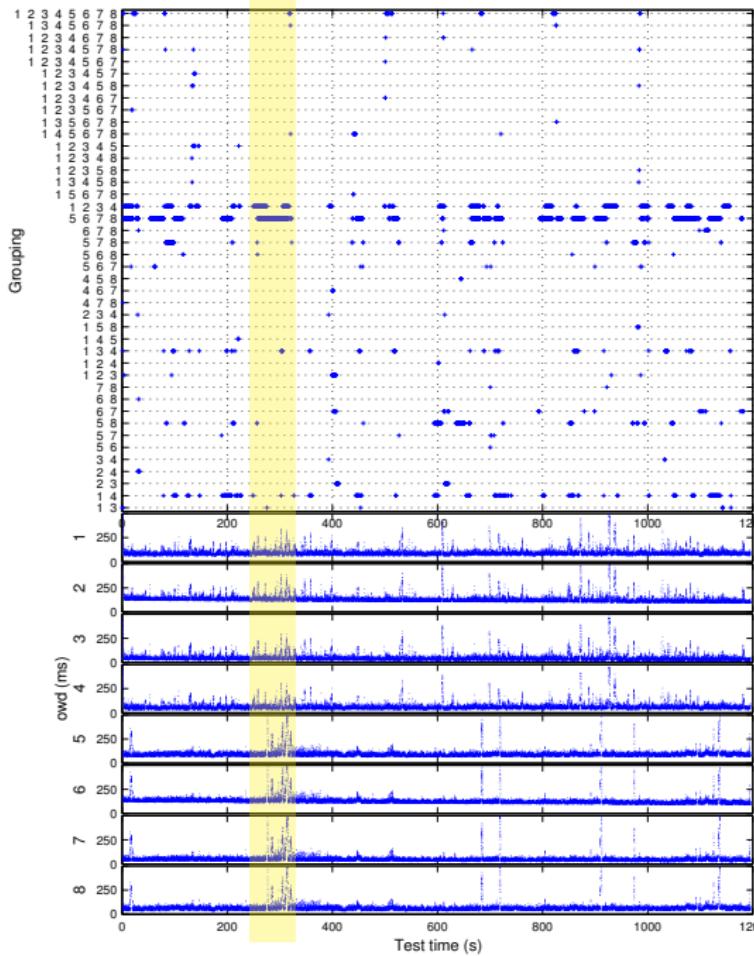


NorNet results

- ▶ Experiment run: 20 min
- ▶ SBD decision: 350 ms
- ▶ SBD simulation settings

- ▶ Transitory bottlenecks
- ▶ During congestion events grouping is usually correct

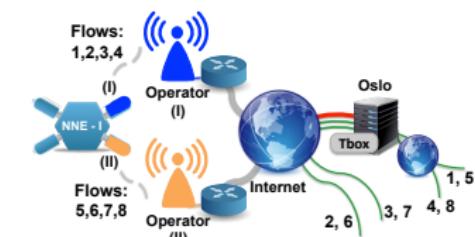


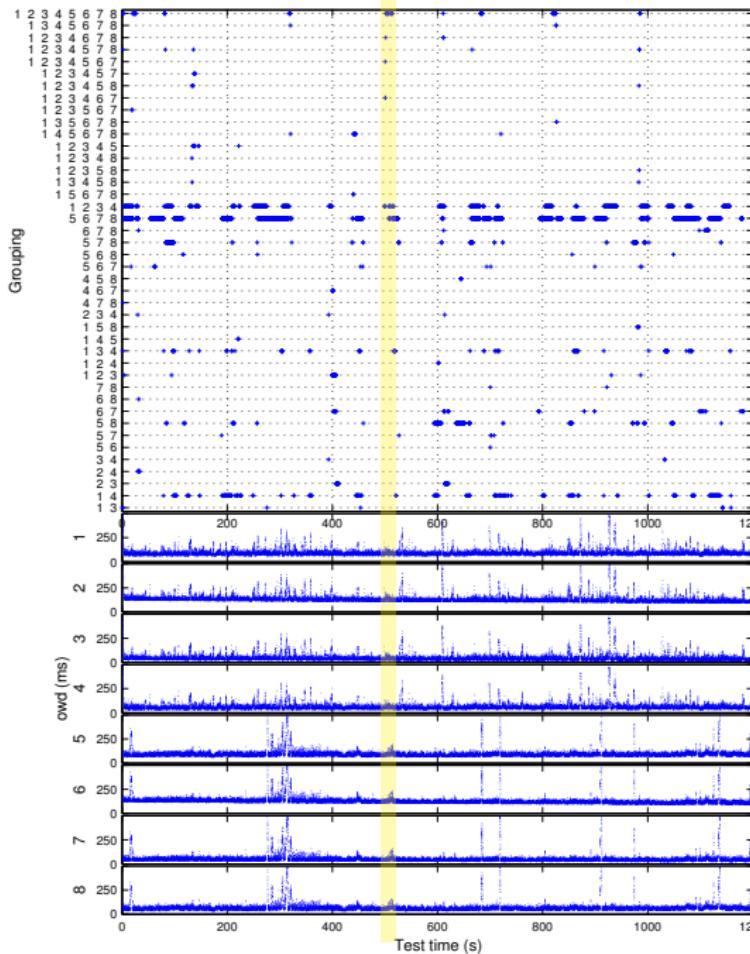


NorNet results

- ▶ Experiment run: 20 min
 - ▶ SBD decision: 350 ms
 - ▶ SBD simulation settings

- ▶ Significant congestion on both 3G links

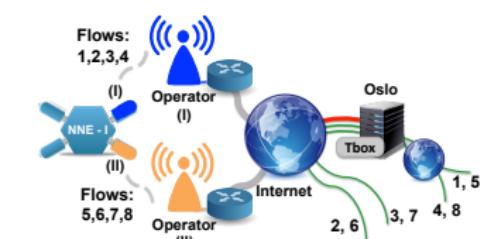


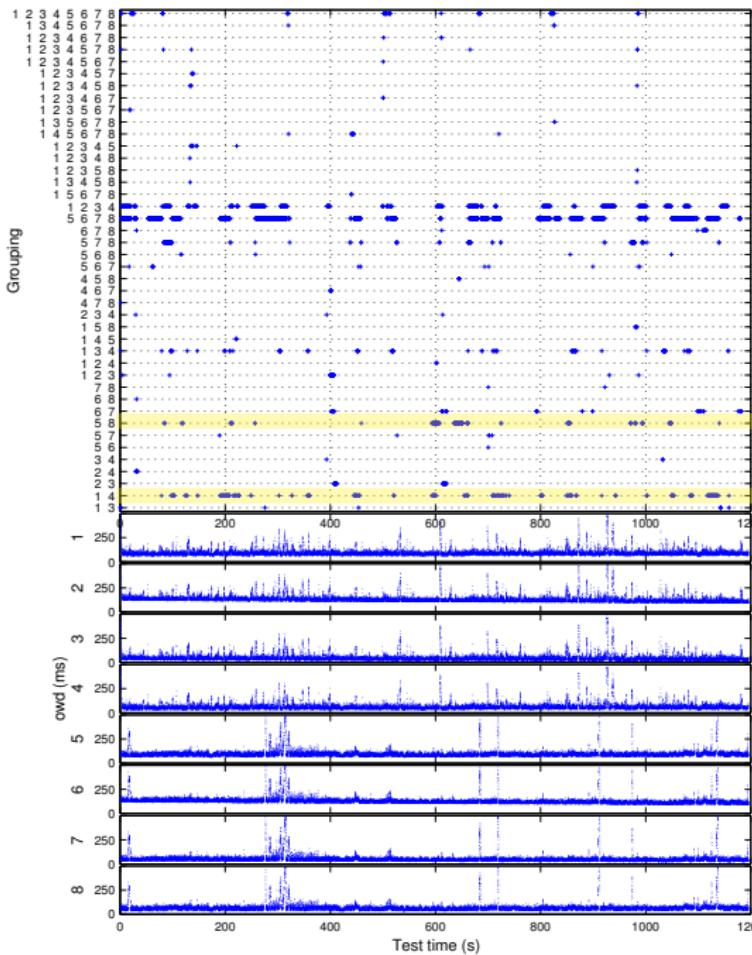


NorNet results

- ▶ Experiment run: 20 min
 - ▶ SBD decision: 350 ms
 - ▶ SBD simulation settings

- Both 3G links sharing a common disturbance

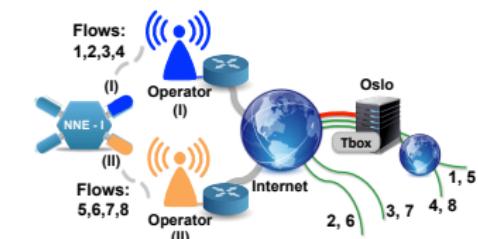




NorNet results

- ▶ Experiment run: 20 min
- ▶ SBD decision: 350 ms
- ▶ SBD simulation settings

- ▶ 1,4,5,& 8 share Tbox
 - ▶ bottleneck at times
- ▶ 1 & 4 share 3G link I
- ▶ 5 & 8 share 3G link II



Lessons Learned

Experiment Setup with NorNet

Traffic Generation

Conclusions and further work

Our Shared Bottleneck Detection mechanism

- ▶ OWD summary statistic estimates:
 - ▶ variance, skew, frequency
- ▶ Groups dynamically changing bottlenecks
 - ▶ But multiple bottlenecks make correct grouping difficult
 - ▶ How realistic are multiple bottlenecks?
- ▶ Shown to be robust in simulation and real network tests

Future work

- ▶ Quantitative performance statistics
- ▶ Comparative performance
- ▶ Integration with coupled congestion control
 - ▶ Time scales of detection
 - ▶ Oscillating bottlenecks
 - ▶ MPTCP (in progress)
 - ▶ Flow State Exchange proposal for IETF RMCAT

Extra slides

Notes on results

- ▶ Decisions made every 350 ms, but based on 6 – 15 s statistics.
- ▶ Decision “points” are large for legibility, but it can tend to magnify errors.
- ▶ Results illustrates what can and can’t be done.

Working with Coupled Congestion Control

- ▶ Summary statistics are gathered at the receivers
- ▶ Shared bottlenecks to a receiver
 - ▶ Receiver does grouping and sends information to senders
- ▶ Shared bottlenecks from a sender:
 - ▶ Receivers send summary statistics for grouping at sender.
- ▶ Can provide the necessary information for a future multi-sender multi-receiver coupled congestion control.