Accurate Delay Measurement for Parallel Monitoring of Probe Flows

Kohei Watabe,
Shintaro Hirakawa,
and Kenji Nakagawa

Graduate School of Engineering, Nagaoka University of Technology

November 30, 2017

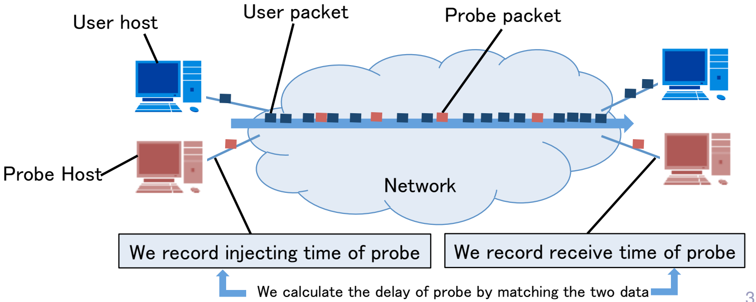
Outline

K. Watabe
S. Hirakawa
K. Nakagawa

- Background
- Assumptions and models
- The key idea of the proposed method
- How to estimate delay metrics
- Experiments
- Conclusions and future works

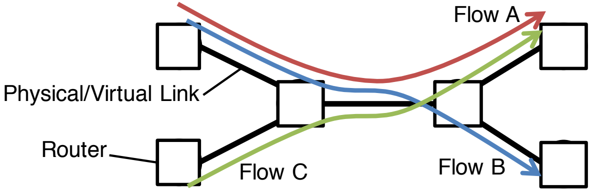
End-to-end Delay Measurements

- End-to-end delay is fundamental for a performance evaluation.
- An active measurement is a common method for end-to-end delay measurements.
  - Probe packets are injected into a network for measurement.
  - It is important to achieve accurate measurement without increasing the number of probe packets.
  - Since a large delay is a rare event in the modern Internet, high quantile of delay distribution is still hard to measure.



Parallel Monitoring of Probe Flows

- Multiple probe flows are monitored to measure delays on multiple paths in parallel for most measurement applications.
  - e.g., SLA monitoring by Internet Service Providers.
- In previous works, only one probe flow of the multiple probe flows is utilized to measure the end-to-end delay on a path.
- However, the information concerning a flow can be utilized supplementary for improving a measurement of another flow.



Objectives

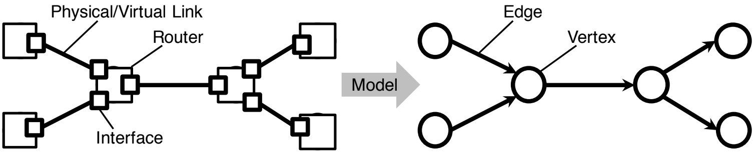
- We propose a novel parallel flow monitoring method in which information concerning all flow can be utilized.
  - The observation results of multiple flows are partially converted into the results of a flow that we want to measure.
  - It does not require any internal information of a measured network (i.e. a topology, current queue size).
- We evaluate the proposed method though simulations.
  - We confirm that the observation results of 72 parallel flows of active measurement are appropriately converted between each other.

Network Model

- A network considered within the scope of this work is represented by a directed graph.

  edge : a physical/virtual link and interfaces at both ends of the link.

  vertex : a part of a network device other than its interfaces.



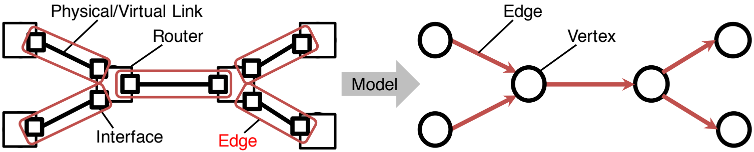
- An end-to-end delay is consisted of propagation delay and queueing delay.
  - Propagation delay can be regarded as a constant.
  - Edges with large queueing delay are sparse.
- To measure delay on paths, probe packets are periodically injected for all or a part of paths.

Network Model (2)

- A network considered within the scope of this work is represented by a directed graph.

edge : a physical/virtual link and interfaces at both ends of the link.

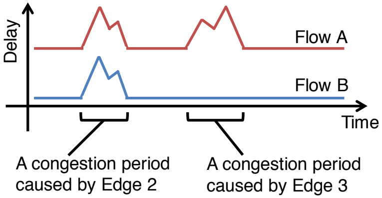
vertex : a part of a network device other than its interfaces.

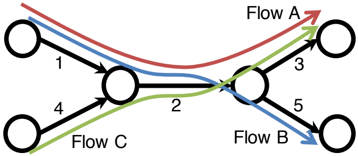


- An end-to-end delay is consisted of propagation delay and queueing delay.
  - Propagation delay can be regarded as a constant.
  - Edges with large queueing delay are sparse.
- To measure delay on paths, probe packets are periodically injected for all or a part of paths.

Overlap of Queueing Delay Processes

$\hat{\chi}_{\mathrm{A}}(t)$: A virtual delay which is the queueing delay experienced by a virtual packet injected into the path of Flow A at time t.

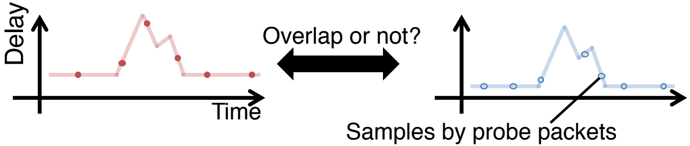




- If $\hat{\chi}_{\mathrm{A}}(t)$ and $\hat{\chi}_{\mathrm{B}}(t)$ in a congestion period tightly overlap, information of the period can be utilized each other.
- Sufficient conditions for overlap is as follows:
  1. The two paths of Flow A and B have the same source;
  2. Routes from the source to the last congested edge on the paths are common;
  3. A queueing delay that packets experience on edges after the last congested edge can be negligible.

Sample Conversion Technique

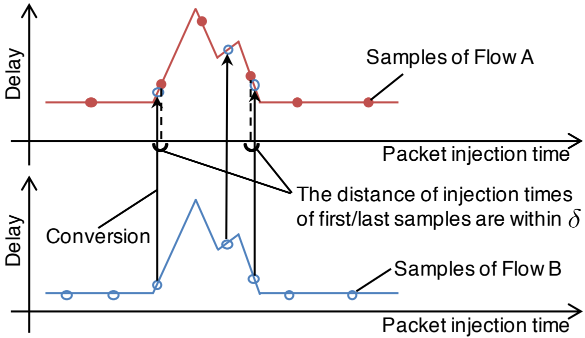
- It is, however, difficult to discriminate the above conditions 2 and 3 without using topology and current queue size.
- We design a method that uses samples of a virtual queueing delay process by probe packets.



- We discriminate whether processes overlap by the following two steps:
  1. Conversion Process
  2. Clustering Process

Conversion Process

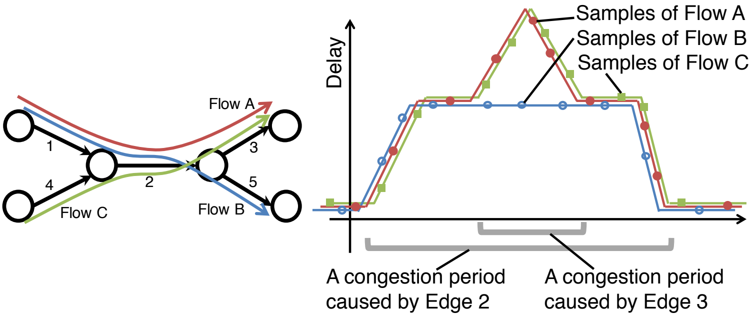
- We consider that virtual delay processes overlap if the two flows satisfy the following conditions:
  1. The two flows have the same source;
  2. The interval between the packet injection times of the first samples in a congestion period is smaller than δ;
  3. The interval between the packet injection times of the last samples in a congestion period is smaller than δ.



- Similar conversion can be considered for destination version.

Clustering Process

- If multiple edges are congested at the same time, the conversion process may convert inappropriate samples.



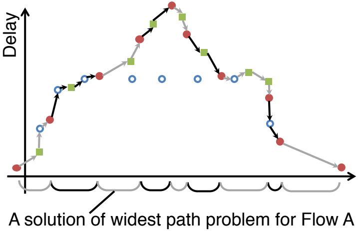
- To remove inappropriate samples, we utilize a clustering technique in machine learning.
- Based on samples that are converted, we construct clusters of flows using a clustering technique.

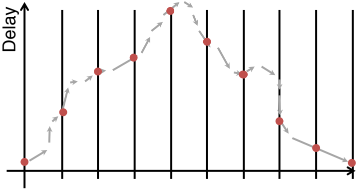
Clustering Process (2)

- To use general clustering techniques, we transform samples of each flow into n-dimensional vectors.
  1. Samples are connected by a solution of the widest path problem.
     - The cost of the edge between two samples is set to the reciprocal of distance
$$\frac{1}{\sqrt{\frac{\beta^2}{\delta^2}(t^i_\mathrm{A} - t^j_\mathrm{B})^2 + (x^i_\mathrm{A} - x^j_\mathrm{B})^2}}.$$

t^i_A : the injection time of ith probe packet of Flow A.
x^i_A : the delay of ith probe packet of Flow A.
β : a control parameter to tune scale of distance of injection time and delay.

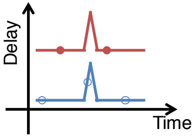
  2. For each flow, we transform the path into an n-dimensional vector by making the vertices evenly spaced.



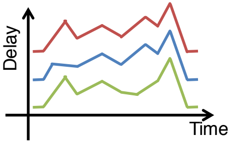


Limitations

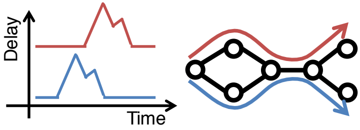
- The proposed method has the following limitations.
  - Momentary congestion: it cannot improve accuracy by converting samples in very short congestion periods.
  - Non-sparse congestion: it cannot detect start and end times of a congestion period if congested edges are not sparse.
  - Complex routes: it cannot convert samples between flows that are once forked and rejoined.



Momentary congestion



Non-sparse congestion



Complex routes

Estimators for Parallel Flow Monitoring

- The samples by our method is not uniformly distributed.
- It is needed to give weight w_i for ith sample with delay d_i.

$$w_i = \begin{cases} N_j/\mathcal{N}_j & \text{for samples in } j\text{th congestion period,} \\ 1 & \text{otherwise.} \end{cases}$$

  - Average end-to-end delay can be estimated by weighted average

$$\frac{1}{N}\sum_{s} w_i d_i,$$

  - q-quantile of end-to-end delay can be estimated by kth delay when delays are arranged in ascending order, where

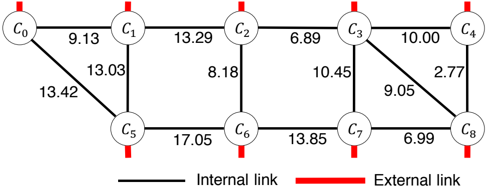
$$k = \arg\max_{j} \left\{ \sum_{i=1}^{j} w_i \leq qN \right\}.$$

N : The number of original sample.
N_j : The number of original sample in jth congestion period.
$\mathcal{N}_j$: The number of all sample in jth congestion period.

Simulation Settings

- We perform NS-3 simulations to confirm that samples of parallel flows are appropriately converted between each other.
- 3 types of traffic stream between all pairs of 9 nodes in a network (i.e., 72 flows stream for each type).

Stationary	Packet size	600 [Byte]
	Traffic pattern	Poisson arrivals
	Traffic intensity	388.8 [Kbps] (4% of a link capacity)
Burst	Packet size	500 [Byte]
	Traffic pattern	On/off process with periodic arrivals
	Traffic intensity	8,000 [Kbps] in burst periods
	Burst period	Exponential distribution with mean 0.1 [s]
	Idle period	Exponential distribution with mean 4.0 [s]
Probe	Packet size	74 [Byte]
	Traffic pattern	Periodic arrivals

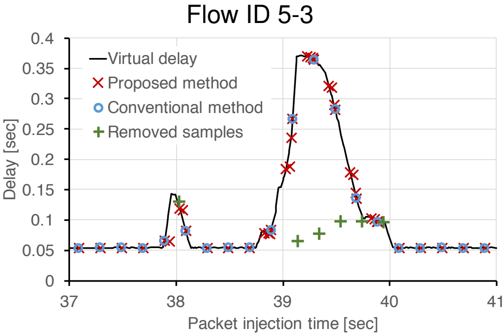


Simulation Settings (2)

- The parameters in the proposed method are as follows:
  - $\beta = 0.16$: The tuning parameter in clustering process.
  - $\delta = 0.2$: Probe packet intervals.
  - $x_{\mathrm{th}} = 0.01$: The threshold to define start/end time of congestion periods.
- In the clustering process, we use Minimum Entropy Clustering (MEC) with 2 parameters:
  - $\alpha = 0.001$: The radius parameter.
  - $e = 10$: The expected number of clusters.
- The simulation time is 42 [s] and we only use the data from 20 [s] to 42 [s].
- The simulation is repeated 10 times by changing the phase of the probe packet injection time.

Examples of Added/removed Samples

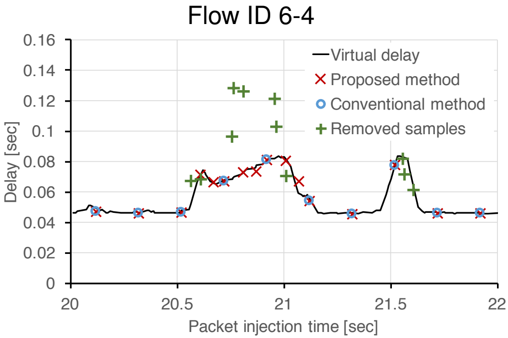
- We depict examples of samples by the conventional and the proposed method.



- The number of samples of the proposed method is larger than those of the conventional method.
- The samples tightly approximate the virtual delay.
- Most of removed samples are not on the virtual delay.

Examples of Added/removed Samples (2)

- We depict examples of samples by the conventional and the proposed method.



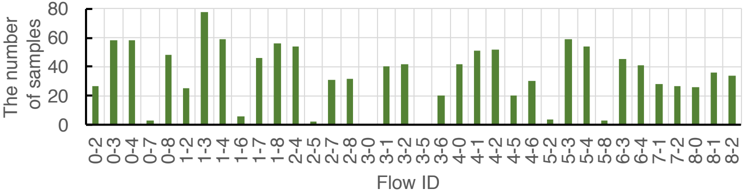
- The number of samples of the proposed method is larger than those of the conventional method.
- The samples tightly approximate the virtual delay.
- Most of removed samples are not on the virtual delay.

Delays, Number of samples

CNSM 2017

K. Watabe
S. Hirakawa
K. Nakagawa

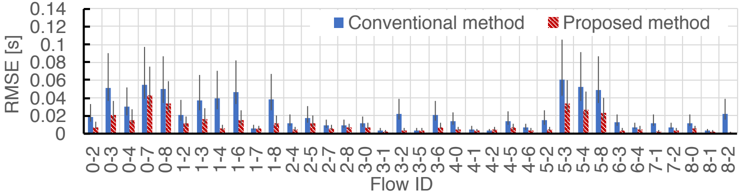
- The number of added samples by the proposed method is shown.
- We display only flows that experience large delay.



- The number of original samples that are obtained from probe packets is 110 samples.
- Up to 78 samples are converted into samples of a flow.

Delays, Number of samples (2)

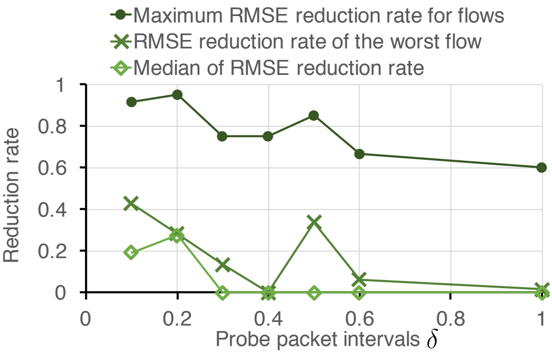
- We also evaluate Root Mean Squared Errors (RMSE) when the 99th-percentile of end-to-end delay is measured.
- The error bars represent 95% confidence intervals.



- The proposed method provides up to 95% reduction of RMSE (Flow ID 8-2).
- The RMSE reduction rate of the worst flow is reduced by 28% (Flow ID 0-7 vs 5-3).

Dependency of RMSE on Probe Intervals δ

- We compare RMSE of 99th-percentile of end-to-end delay by changing probe packet interval δ from 0.1 to 1.0 [s].



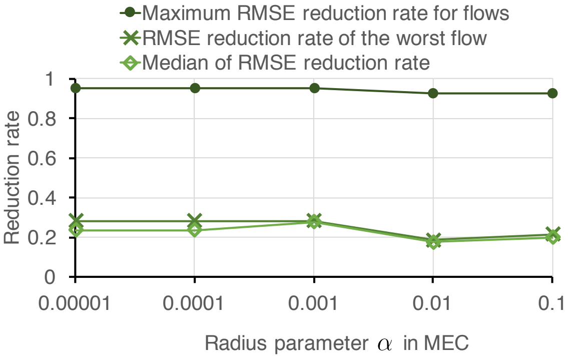
- The values of maximum RMSE reduction rate for flows are high.
- RMSE reduction rate of the worst flow and median of RMSE reduction rate occasionally decrease to nearly 0.

Independency of RMSE in MEC

CNSM 2017

K. Watabe
S. Hirakawa
K. Nakagawa

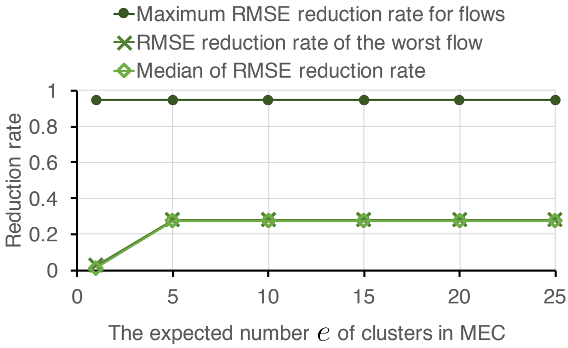
- We confirm the dependency of RMSE on parameters of MEC.



- It is confirmed that the RMSE is independent of the parameter from the figure.

Independency of RMSE in MEC (2)

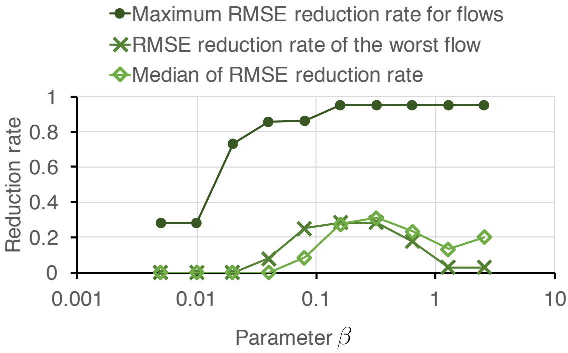
- We confirm the dependency of RMSE on parameters of MEC.



- It is confirmed that the RMSE is independent of the parameter from the figure.

Dependency of RMSE on β

- We verify the dependency of the performance of the proposed method on parameter β.



- The maximum RMSE reduction rate for flows increases as parameter β increases.
- The other reduction rate decrease due to the inappropriate conversions of samples when β is between 0.64 and 2.56.

Conlusion

In this paper, we proposed a parallel flow monitoring method that achieves accurate measurement by partially converting the observation results each other.

- The proposed method adds to samples of a flow from the samples of the other flows.
- It also removes inappropriate samples using a clustering technique in machine learning.
- It provides up to 95% reduction of errors, and the error of the worst flow among all flows is reduced by 28%.

Future Works

- We have the following 4 directions for future works.
  - Automatic tuning of the parameters.
  - Development of a method that utilizes a network topology for the conversion process.
  - Extension toword measurements of other metrics.
  - Evaluation of the proposed method using real network traffic.

Thank you for your kind attention.