Packet Doppler: Network Monitoring using Packet Shift Detection

Tongqing Qiu, Nan Hua, Jun (Jim) Xu
Georgia Tech

Jian Ni, Hao Wang, Richard (Yang) Yang
Yale University

Presenter: Richard Ma

December 10th, 2008

Motivation

- Users demand strict guarantees from their network providers on key metrics:
  - Stringent demand on performance monitoring
- Dominant type of monitoring approach is based on end-to-end probing.

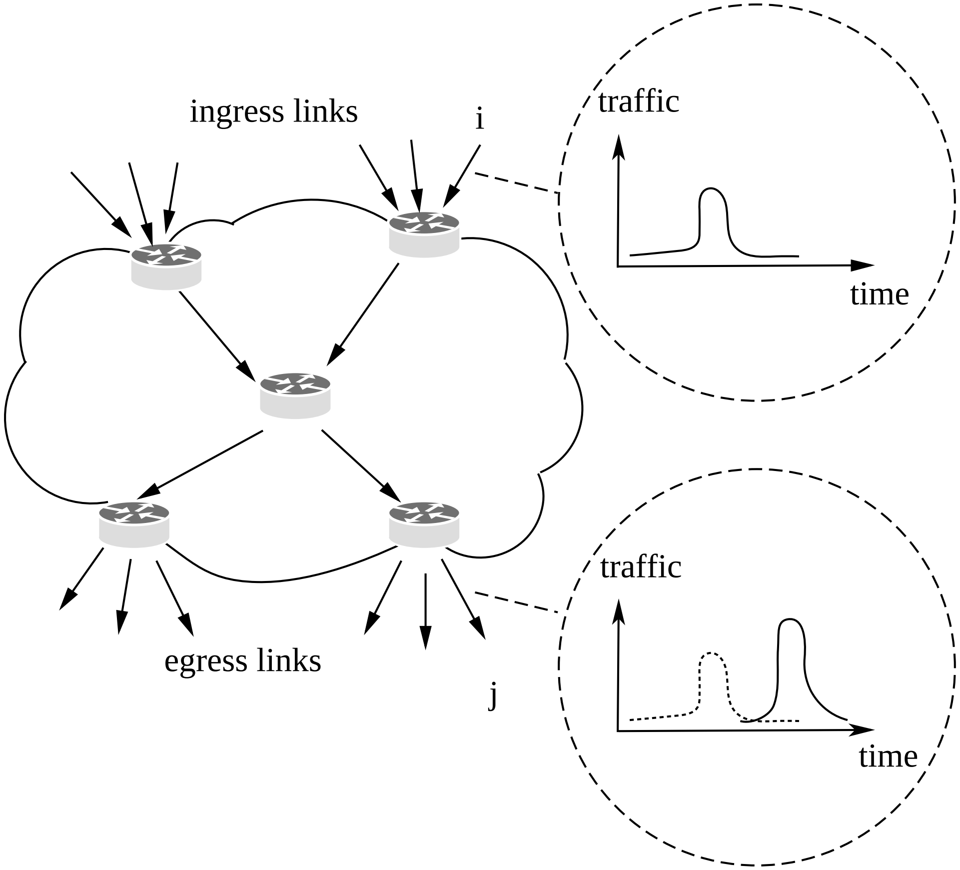
The Drawbacks of End-to-end Probing

- Probing is overhead: reduces usable network bandwidth.
  - Scalability issue: n edge nodes, probing n^2 pairs.
- Probing may cause perturbation to real traffic.
  - The effect of perturbation can be more serious when the network load is high, the moment when it is more important to get more accurate measurements.
- Active probing estimates the performance of the probing traffic instead of the real traffic.
  - Performance anomalies (e.g., bursts) between probing intervals may be missed.

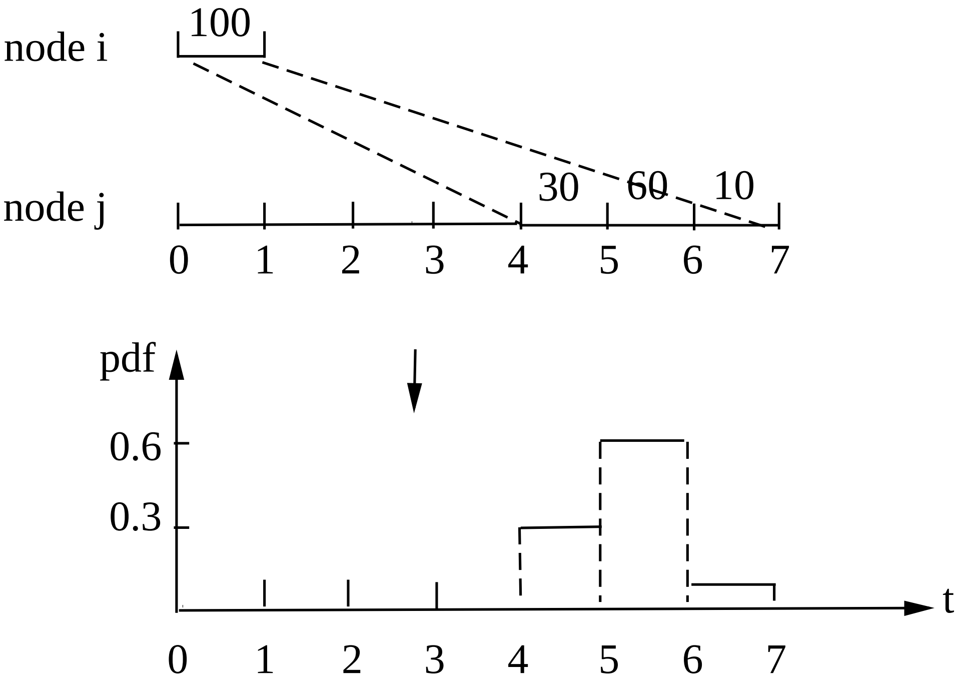
Our contributions

- Packet Doppler, a passive monitoring methodology.
- Packet Doppler can estimate delay with high precision and low overhead.
- Packet Doppler can detect burst of delay changes.
- Two novel decoding methods to effectively improve decoding accuracy.

The phenomenon of packet shift



Estimation of delay distribution



Estimation of delay distribution from the packet shift process.

General Steps

- Encoding Procedure
  - Process each packet (maximize information)
  - Hash to one position in the bitmap b (minimize cost)

- Decoding Procedure: Estimate the number of packets [Wang *et al.* 1990]

$$
\begin{aligned}
\hat{n}_i(s) &= b \ln \frac{b}{u_{A_i(s)}}; \\
\hat{m}_j(t) &= b \ln \frac{b}{u_{D_j(t)}}; \\
\hat{x}_{i,j}(s,t) &= \hat{n}_i(s) + \hat{m}_j(t) - b \ln \frac{b}{u_{A_i(s) \cup D_j(t)}}. \qquad (1)
\end{aligned}
$$

Limitation of Naive Decoding

- When the amount of traffic between i and j is small relative to the cross traffic at i and/or j , the naive decoding scheme may report imprecise delay values since the intersection estimation can be drown out by noise.
- Better decoding schemes are needed.
  - Sequential Hypothesis Testing (SHT)
  - Iterative Proportional Fitting (IPF)

A Hypothesis Testing Perspective

- Every possible shift can be viewed a hypothesis H. Based on the bitmap observation, we test all the hypotheses and select the most likely hypothesis as the detection result.

- We consider K hypotheses H_1, H_2, $\ldots$, H_K. We want to find the most likely hypotheses H_t.

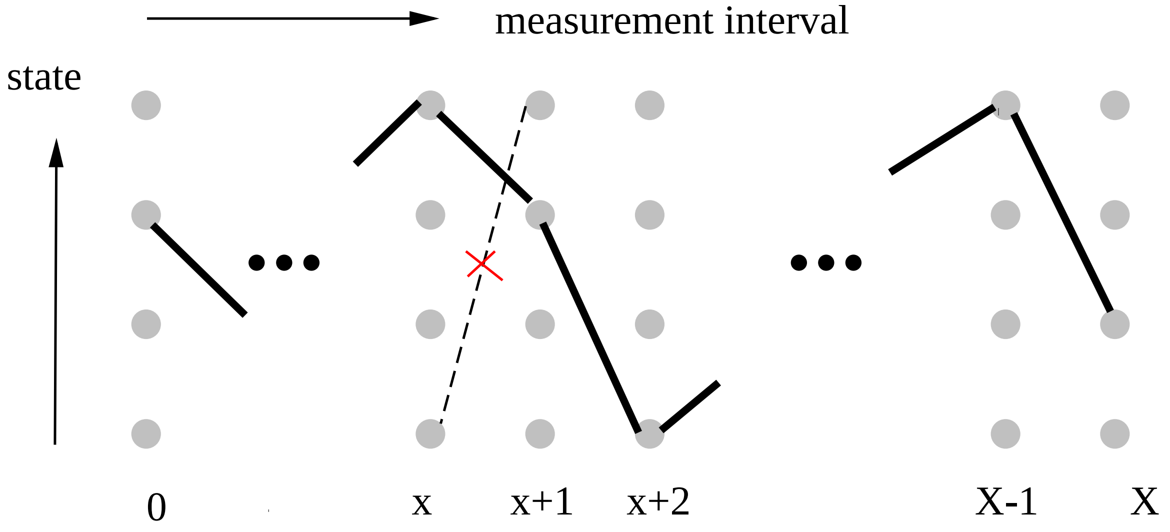
- Applying likelihood ratio test, we compute the likelihood ratio $\lambda_t(\hat{x})$ for each hypothesis, and find t to maximize $\lambda_t(\hat{x})$

$$\lambda_t(\hat{x}) = \frac{\sup_{\Theta_t} L(\theta_t|\hat{x})}{\sup_{\Theta} L(\theta_t|\hat{x})}. \tag{2}$$

Sequential Hypothesis Testing

- Instead of performing hypothesis testing for each and every source interval individually, we bundle 2 or more consecutive source intervals together and perform a joint hypothesis testing on them.
- The hypothesis space grows exponentially with the number of intervals bundled together.
- near-FCFS assumption: We assume that, for any time s, the vast majority (say more than 95%) of packets in $A_i(s)$ will be delivered no later than those of $A_i(s+1)$ by more than one unit of time.

Sequential Hypothesis Testing (cont.)



An example of Viterbi decoding. There are four states. The solid line represents the active path, and dash line represents an example of an invalid path.

- Viterbi decoding is a dynamic programming algorithm in essence.

Iterative Proportional Fitting

- Improve the estimation (minimize the noise) based on conservation law.
- The estimation noises are asymptotically Gaussian noises [Wang *et al.* 1990].
- Conservation law: all packets coming into the network have to exit somewhere at some point in time.

$$\hat{n}_i(s) + \epsilon_i(s) = \sum_j \sum_{t \in T_{ij}(s)} (\hat{x}_{ij}(s,t) + \epsilon_{ij}(s,t)), \ \forall i, s;$$

Iterative Proportional Fitting(cont.)

- The procedure: **Iteratively** compute the noise $\epsilon_i(s)$, $\epsilon_j(t)$ and $\epsilon_{ij}(s,t)$, **proportionally fitting** based on variance. For example:

$$\epsilon_i^*(s) \quad = \quad \frac{v_i(s)}{v_i(s) + \sum_{j,t} v_{ij}(s,t)} \Big(\sum_{j,t} \hat{x}_{ij}(s,t) - \hat{n}_i(s) \Big),$$

  Then update the estimation value $\hat{n}_i(s)$ $\hat{m}_j(t)$ $\hat{x}_{ij}(s,t)$ according.

- We can prove that IPF will converge to the joint MLEs.

Evaluations: Experiment Setup

- An event-driven platform: one event occurs when there is a packet arrives at an ingress link or departs from an egress link.
- **Delay:** Delay traces collected from one source node to five PlanetLab nodes at five geographically distributed locations.
- **Traffic demand:** traffic demand traces collected from Abilene.
- **Performance metrics:** mean, variance and distribution (PDF) of delays.

Overhead

- Packet Doppler:
  - Per link digest size: 2 Kb.
  - Per link digests generated and stored per second: 2 Mb.
  - If the SLA monitoring process is performed online, the transmission bandwidth cost is also 2 Mb/s.
  - Network-wide storage/digest size: n * 2 Mb.
- Packet probing:
  - SLAm: probing bandwidth for each pair: 80Kb/s.
  - Network-wide bandwidth: n^2 * 80 Kb/s.
- Observation: The larger the network, the more benefit by Packet Doppler.

Experiment Results: mean delay and variance

Destination	SLAm	Naive	SHT	SHT+IPF
Node0	0.24	0. 21	0.11	0.09
Node1	0.39	0.31	0.10	0.03
Node2	0.53	0.43	0.03	0.01
Node3	1.39	1.04	0.07	0.05
Node4	2.56	1.35	0.15	0.13

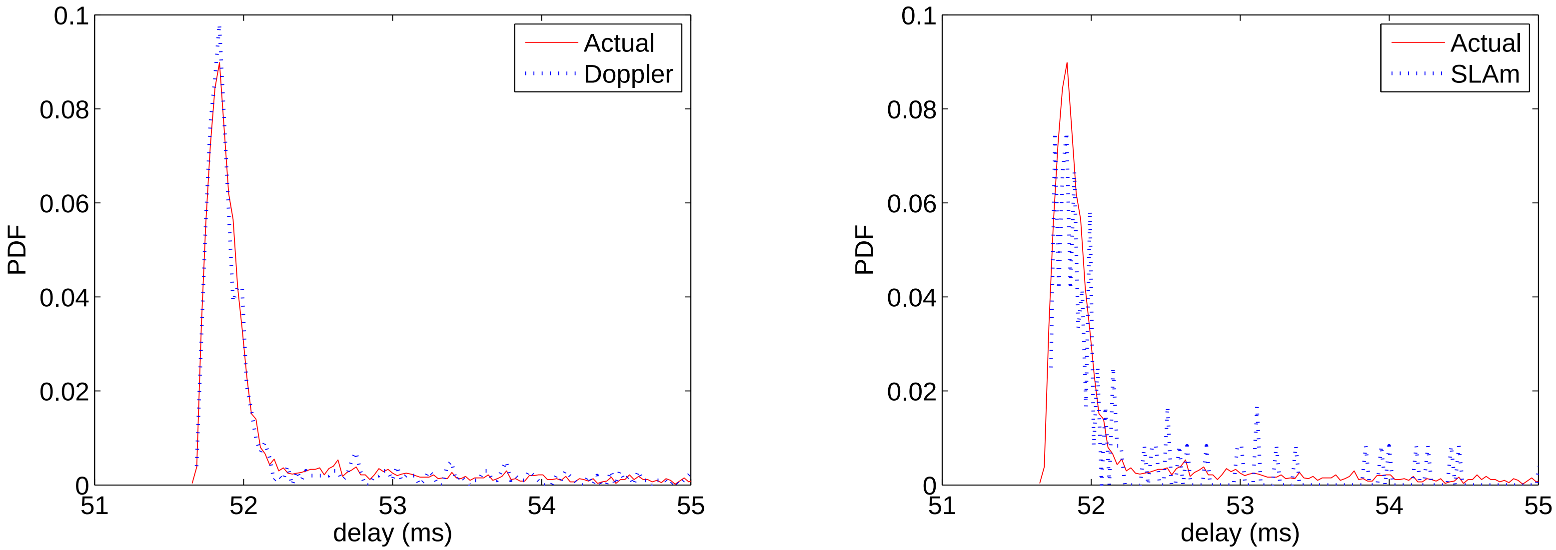
Table 1: Absolute error of mean delay estimation (milliseconds)

Destination	SLAm	Naive	SHT	SHT+IPF
Node0	20.3%	18.2%	7.5%	4.9%
Node1	54.3%	43.4%	14.6%	9.2%
Node2	50.4%	50.3%	8.3%	5.1%
Node3	36.5%	35.1%	3.8%	1.9%
Node4	70.1%	52.2%	5.6%	2.6%

Table 2: Relative error of delay variance estimation

- Packet Doppler (SHT+IPF) is the best to estimate the mean delay and delay variance.

Experiment Results: delay distribution

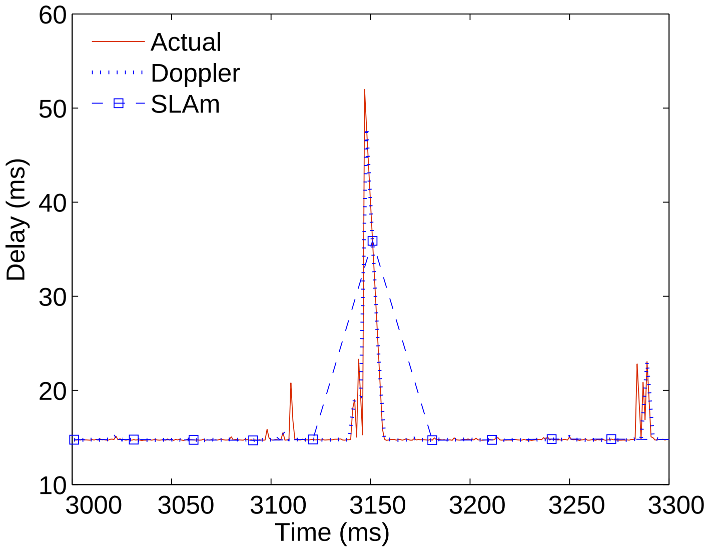


(a) Doppler

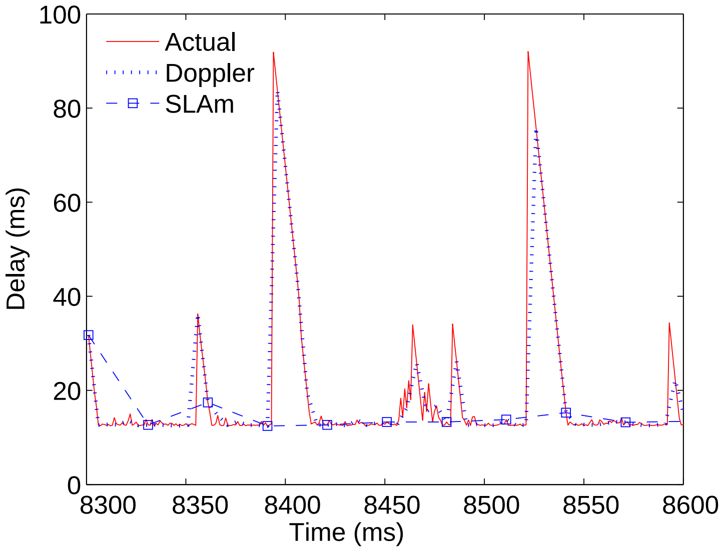
(b) SLAm

- Packet Doppler matches the real delay distribution well.

Experiment Results: burst detection



(c)



(d)

- Take two short periods for example, Packet Doppler can detect more bursts.

Summary

- In this paper, we propose Packet Doppler as a scheme to monitor delay distribution, a key service level agreement metric.
- Packet Doppler is passive and thus do not cause perturbation to real traffic.
- It is also scalable with low storage and bandwidth overhead.
- Evaluations using realistic delay and traffic demands show that Packet Doppler achieves high accuracy and can detect burst events that may be undetected by probing based schemes.
- Future work: Extend to packet loss.

Thank you!

Questions?