IMR-Pathload: Robust Available Bandwidth Estimation under End-Host Interrupt Delay

Seong Kang

Joint work with Dmitri Loguinov

Internet Research Lab
Department of Computer Science
Texas A&M University, College Station, TX 77843

April 30, 2008

Agenda

- Introduction
- Interrupt moderation
- Analysis of Pathload
  - Impact of interrupt delays
  - Trend detection problem
- IMR-Pathload
- Performance evaluation
- Wrap-up

Introduction

- Bandwidth estimation is an important area of Internet research
  - Plays an important role in characterizing network paths
  - Potentially can help various Internet applications
- The vast majority of tools focuses on end-to-end measurements
  - The ultimate goal is to measure diverse Internet paths under various traffic and network conditions
  - Fast estimation and high accuracy are desired

Introduction (2)

- All existing methods heavily rely on high-precision delay measurement at end-hosts
  – However, delay measurements are not perfect in practice
  – Interrupt delays at NIC cause timing irregularity
- State of the art tools attempt to reduce the effect of interrupt delays
  – Pathchirp and Pathload aim to “weed out” packets affected by interrupt delays

Introduction (3)

- Pathchirp
  - Sends substantially more packets by setting an option manually
  - Not desirable since it prolongs measurement duration
- Pathload
  - Filters out affected packets without increasing the number of probing packets
  - Has limited effect when interrupt delays are non-trivial
- Goal
  - To develop a tool that is robust to timing irregularity caused by NIC’s interrupt moderation
  - Mainly focus on improving Pathload

Agenda

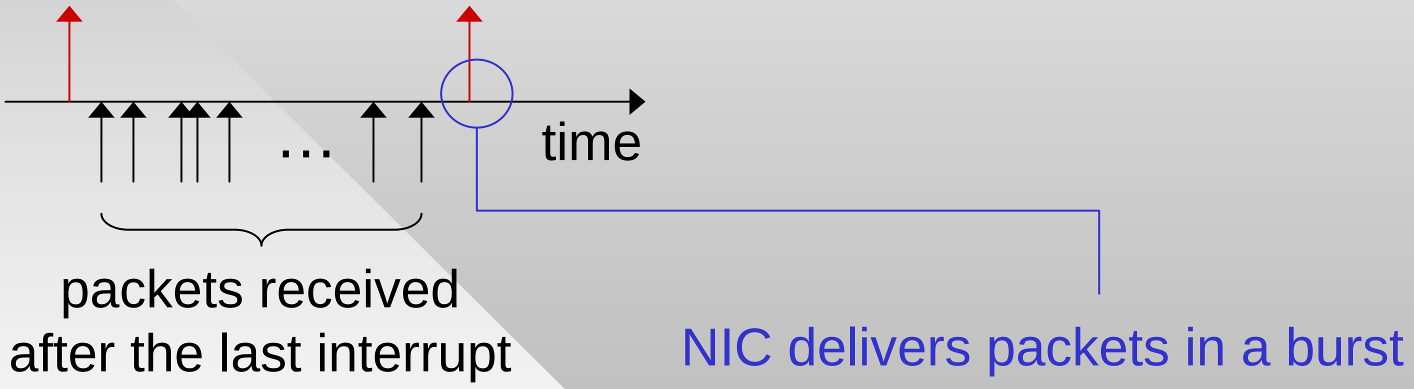
- Introduction
- Interrupt moderation
- Analysis of Pathload
  - Impact of interrupt delays
  - Trend detection problem
- IMR-Pathload
- Performance evaluation
- Wrap-up

Interrupt Moderation

- Packet arrival/departure events at a network interface card (NIC) is handled by the CPU through interrupts
- Generating interrupts for every packet event creates significant per-packet overhead
  – For a Gigabit Ethernet NIC, an interrupt could be generated every $12\ \mu s$ with packets of size 1500 bytes
  – Substantial overhead for interrupt handling
- Solution to this is using interrupt moderation
  – Delays generation of a new interrupt
  – Stores packets at NIC until the next interrupt

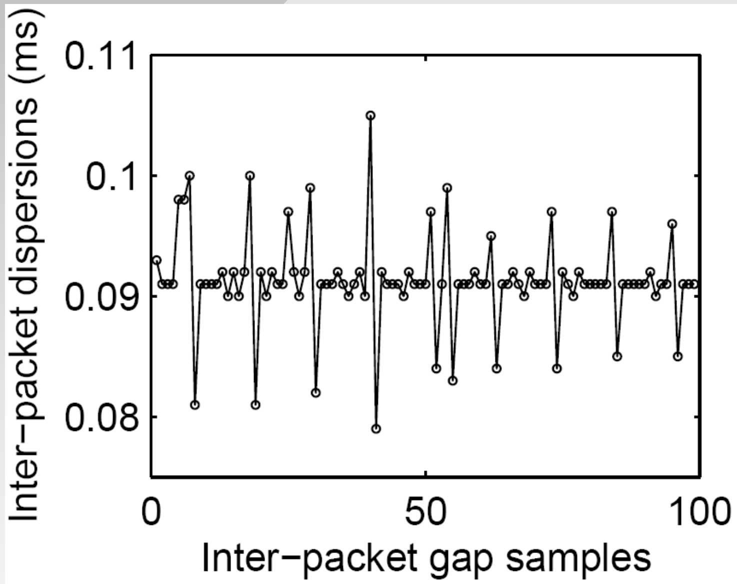
Interrupt Moderation (2)

- It has become a common practice with Gigabit NICs
- At a single interrupt, NIC delivers multiple packets to the kernel

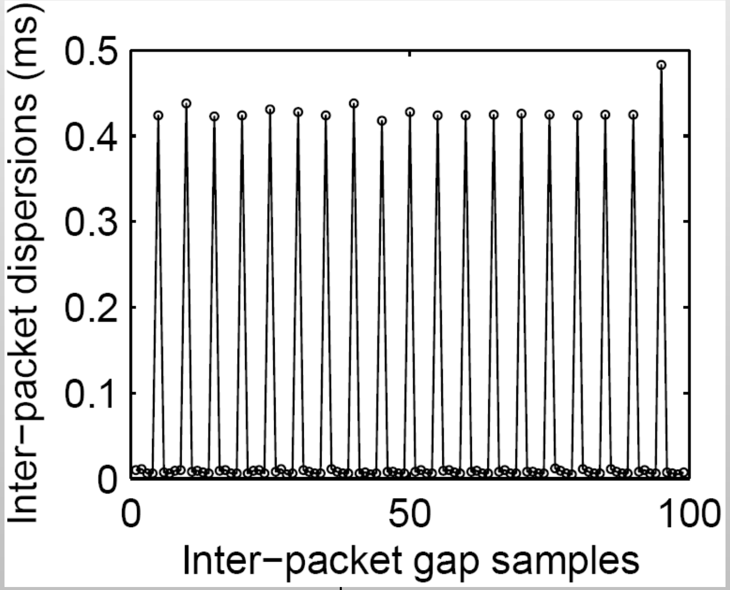


Interrupt Moderation (3)

- Impact on inter-packet dispersions



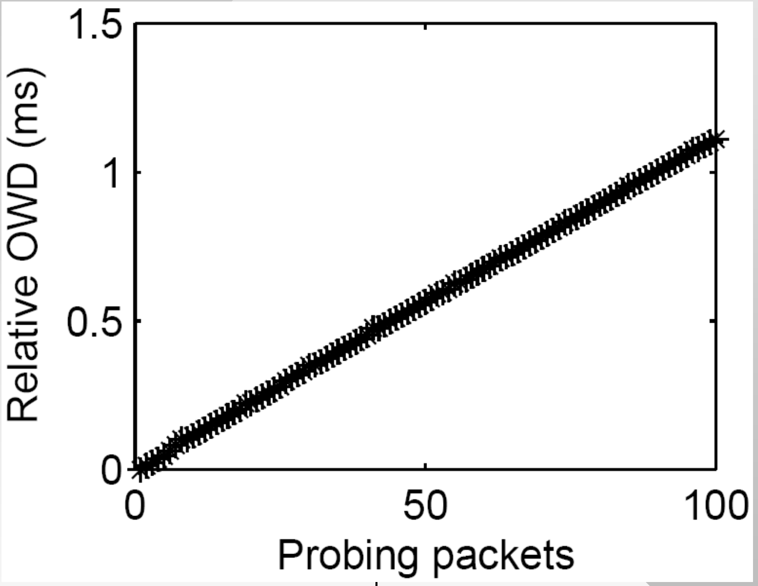
no interrupt moderation

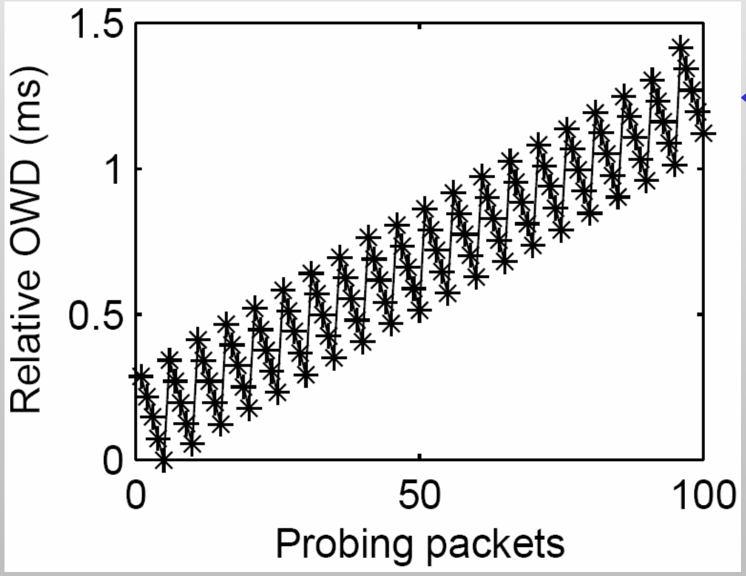


interrupt moderation

Interrupt Moderation (4)

- Impact on one-way delays (OWD) of probing packets
  - Difference between the sending time and arrival time





no interrupt moderation

interrupt moderation

Negative trend in a single burst appears when receive latency is larger than inter-packet spacing at the sender

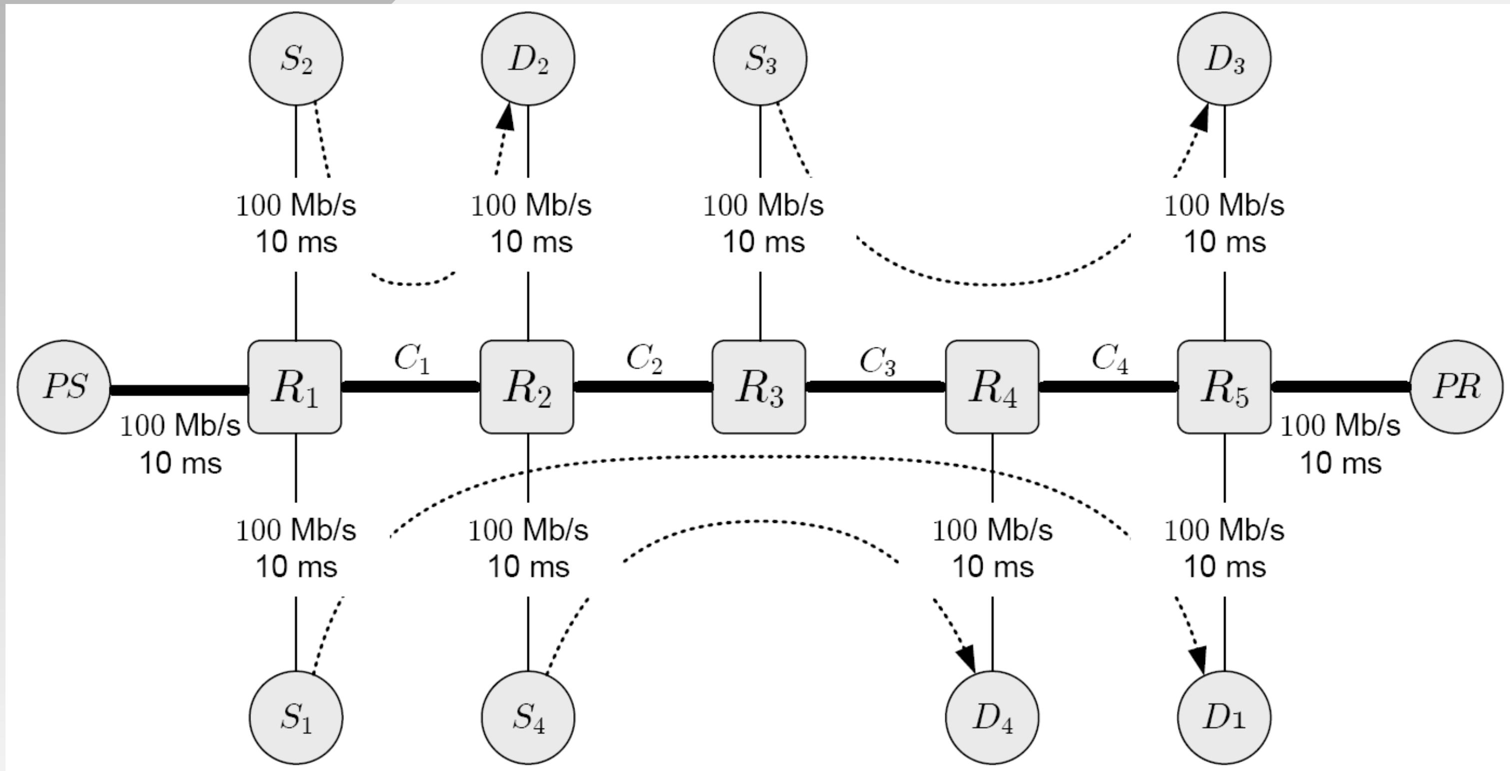
Agenda

- Introduction
- Interrupt moderation
- Analysis of Pathload
  - Impact of interrupt delays
  - Trend detection problem
- IMR-Pathload
- Performance evaluation
- Wrap-up

Observation

- Many paths in PlanetLab cannot be measured by Pathload
  - We suspect that timing irregularity due to interrupt moderation is the major reason
- Thus, we investigate how interrupt delays affect Pathload's estimation
  - Conduct experiments in Emulab for different interrupt delays at the receiver
- We start by describing a topology for Emulab experiments

Experimentation Topology



– The speed of all access links is 100 Mb/s (delay 10 ms)

– The remaining links between two routers have capacities C_i and propagation delay 40 ms

– TCP cross-traffic is generated by Iperf traffic generator

- Run 100 threads in each cross-traffic source S_i

Experimentation Setup

Experimentation scenarios	Different link bandwidths (Mb/s)							
	C_1	A_1	C_2	A_2	C_3	A_3	C_4	A_4
Case-I	75	31.84	90	51.69	90	42.05	[60]	40.77
Case-II	75	41.32	90	70.76	90	46.77	[60]	26.39
Case-III	[60]	35.88	90	70.76	[90]	23.39	75	18.10
Case-IV	[60]	21.60	90	65.99	90	42.07	75	36.72
Case-V	[60]	50.25	90	61.17	90	41.99	75	50.86
Case-VI	75	28.97	90	37.8	90	13.86	[60]	31.22

– Shaded values in each row represent the capacity and available bandwidth of the tight-link for each case
  - Tight link is the link with the smallest available bandwidth
– Values in square brackets represent the capacity of the narrow link for each case
  - Narrow link represents the link with the lowest speed

Computer Science, Texas A&M University

Experimentation Setup (2)

- Define e_A to be relative estimation error:

$$e_A = |A - \tilde{A}|/A$$

true available bandwidth of a path

estimate of A

Estimation Reliability

- Next examine estimation behavior of Pathload with various interrupt delays

With small interrupt delay, estimation accuracy is over 80%

Interrupt delay δ	Evaluation scenario Case-I	Case-II	Case-III	Case-IV	Case-V	Case-VI
$0\ \mu s$	9.45%	8.00%	7.57%	6.48%	16.58%	15.01%
$100\ \mu s$	1.44%	8.52%	14.9%	5.74%	3.6%	20.74%
$125\ \mu s$	$--$	$--$	15.01%	$--$	$--$	34.65%
$> 125\ \mu s$	$--$	$--$	$--$	$--$	$--$	$--$

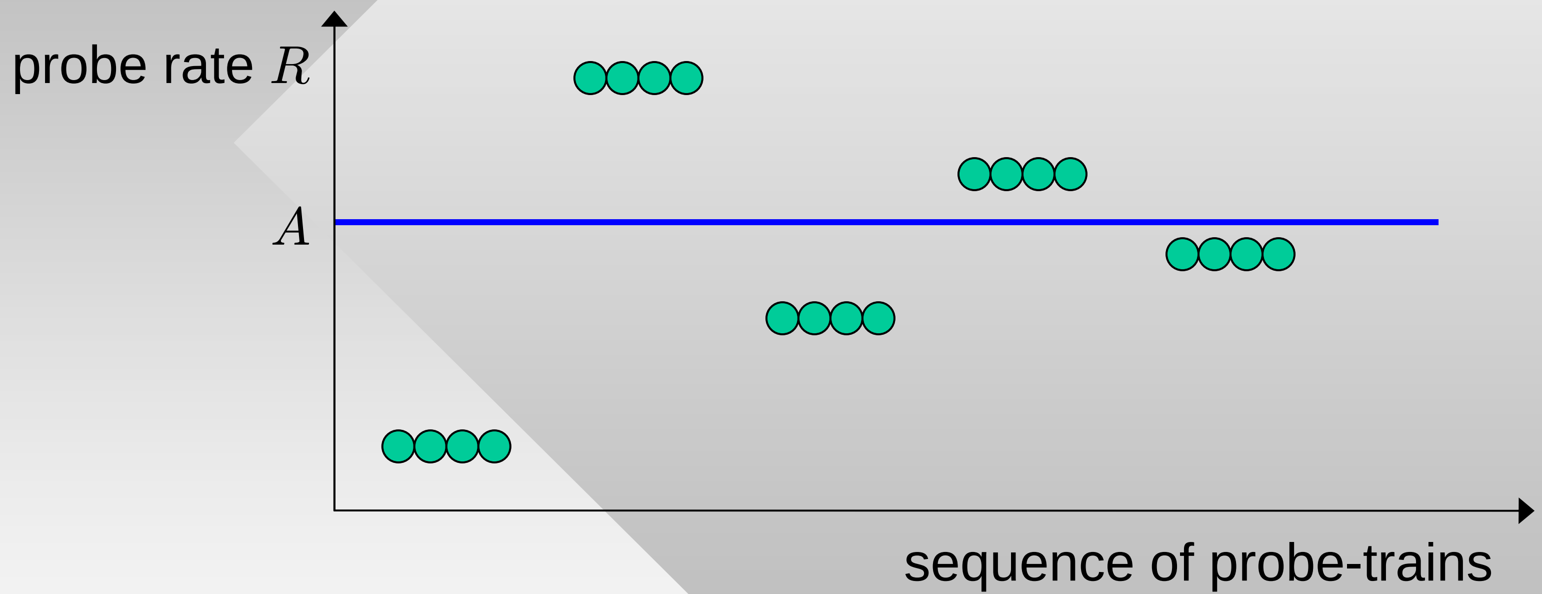
When the delay becomes larger, Pathload is unable to produce reliable estimates

Estimation Algorithm

- Recall that Pathload sends a sequence of packet-trains with a rate R
  - Each train includes N back-to-back packets
- Receiver examines OWDs in each train and returns their trend information to the sender
- Sender adjusts its probe rate R in a binary search fashion based on the trend information
  - Increase the probe rate R if no trend is detected
  - Decrease R if an increasing trend is detected

Estimation Algorithm (2)

- Search for an appropriate probe rate R



Agenda

- Introduction
- Interrupt moderation
- Analysis of Pathload
  - Impact of interrupt delays
  - Trend detection problem
- IMR-Pathload
- Performance evaluation
- Wrap-up

Trend Detection

- PCT and PDT metrics are used for trend detection
- PCT (Pairwise Comparison Test)

$$PCT = \frac{1}{n} \sum_{j=2}^{n} I(X_j > X_{j-1})$$

OWD of a packet j–1 in a set of size n

$I(Y)$ is one if Y holds, zero otherwise

  – Represents the fraction of consecutive OWD pairs that are increasing

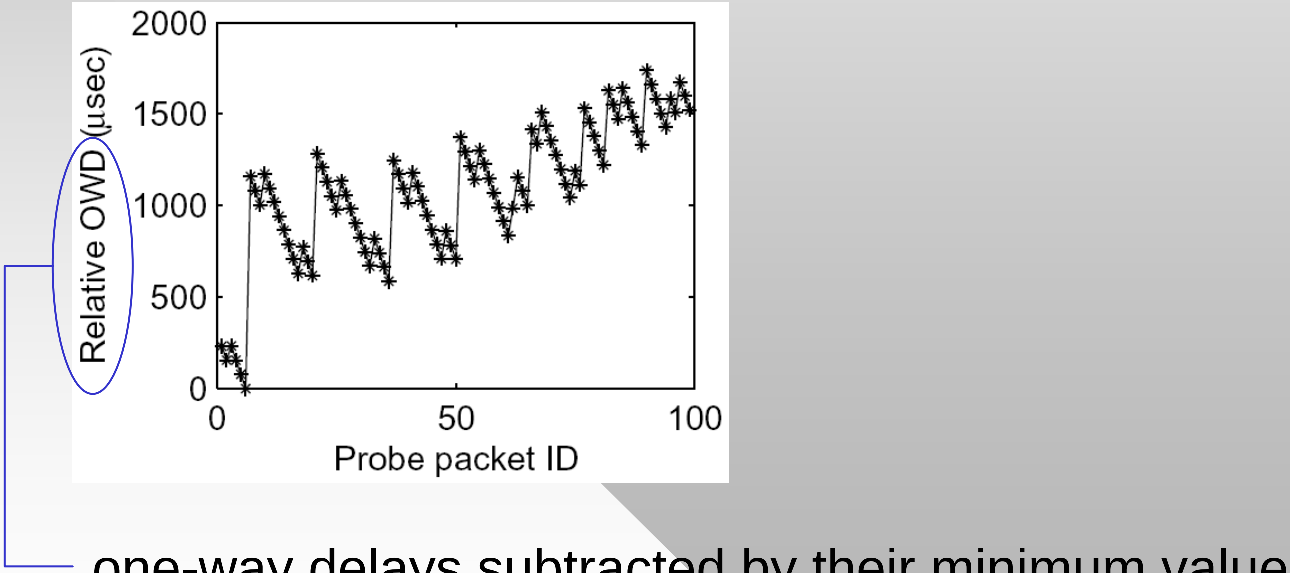
- PDT (Pariwise Difference Test)

$$PDT = (X_n - X_1) / \sum_{j=2}^{n} |X_j - X_{j-1}|$$

  – Quantifies how strong the difference between the first and last OWDs in the data set is

Trend Detection (2)

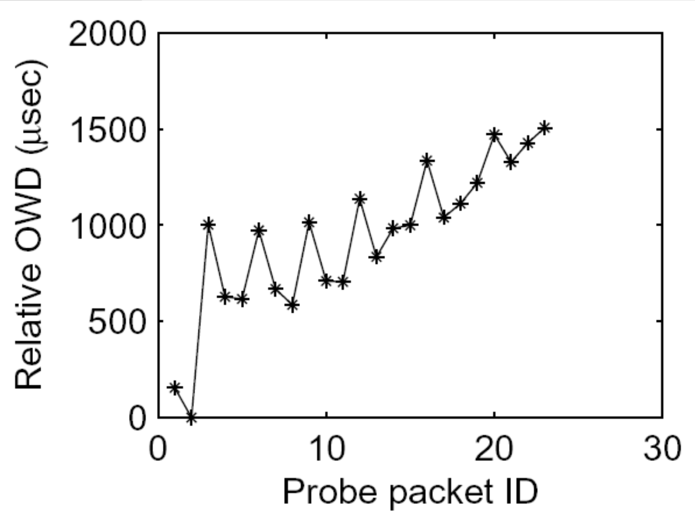
- To assess Pathload's trend detection mechanism, we conduct experiments for Case I ($A = 31$ Mb/s)
  - Collect OWD data by running Pathload with a fixed rate R=38 Mb/s and interrupt delay $\delta = 250\ \mu s$



  - OWDs exhibit increasing trend overall

Trend Detection (3)

- Before applying PCT and PDT tests, Pathload eliminates coalesced (back-to-back) packets



- However, it is unable to detect an increasing trend in the OWDs as it obtains $PCT = 0.5$, $PDT = 0.11$
  - "increasing" if $PCT > 0.66$, "non-increasing" if $PCT < 0.54$
  - "increasing" if $PDT > 0.55$, "non-increasing" if $PDT < 0.45$

Agenda

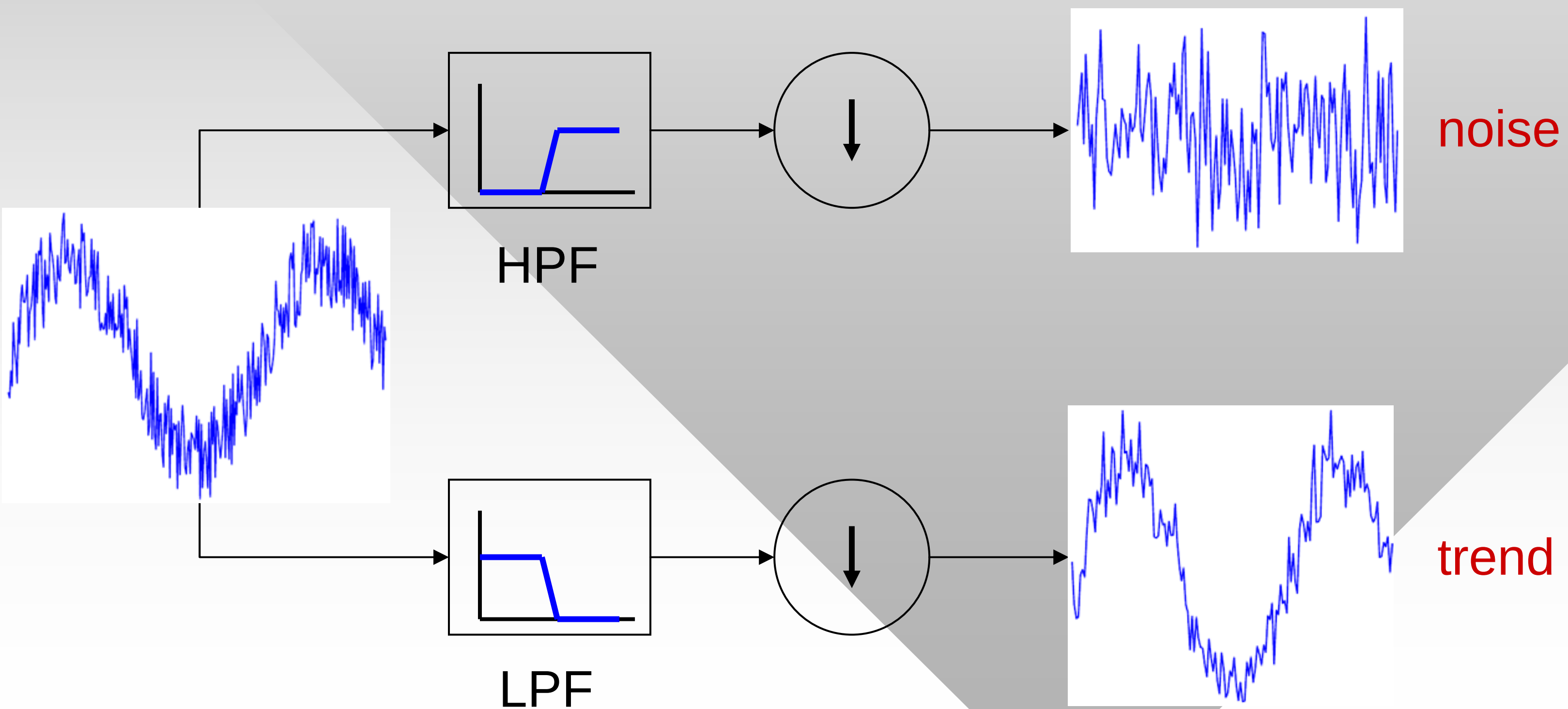
- Introduction
- Interrupt moderation
- Analysis of Pathload
  - Impact of interrupt delays
  - Trend detection problem
- IMR-Pathload
- Performance evaluation
- Wrap-up

IMR-Pathload

- Characterizing delay trend in measured noisy OWD data is a difficult problem
  - Pathload's trend detection algorithm is not much effective in dealing with this
- To overcome this, we introduce two noise-filtering techniques in bandwidth measurement
  - Wavelet-based signal processing
  - Window-based averaging
- IMR-Pathload
  - Interrupt Moderation Resilient Pathload

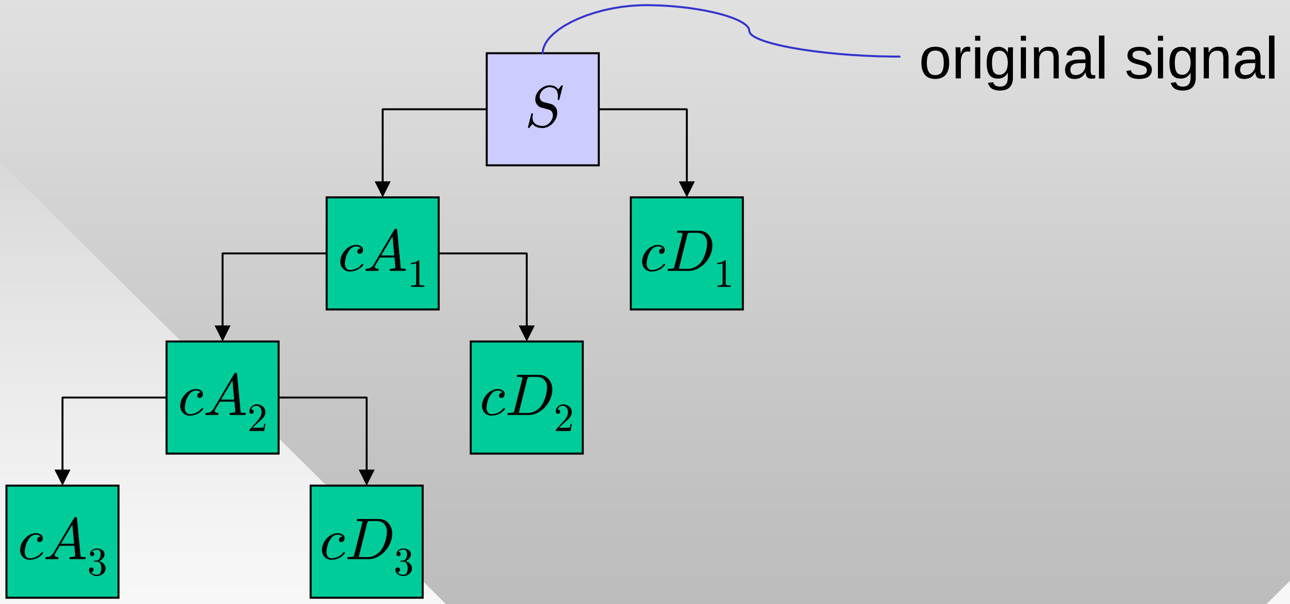
IMR-Pathload (2)

- OWD process can be decomposed into two components using wavelet decomposition
  - Scale coefficients represent deterministic "trend"
  - Wavelet coefficients represent stochastic "noise"



IMR-Pathload (3)

- Decomposition can be iterated
  - Successive scale coefficients are decomposed in turn



- cA_j: scale coefficients in level j
- cD_j: wavelet coefficients in level j

IMR-Pathload (4)

- OWD data are processed using wavelet decomposition or k-packet window-based averaging
  - For experiments, we use Daubechies length-4 wavelets
- Scale coefficients are given by:

$$h_0 = \frac{1+\sqrt{3}}{4\sqrt{2}},\ h_1 = \frac{3+\sqrt{3}}{4\sqrt{2}},\ h_2 = \frac{3-\sqrt{3}}{4\sqrt{2}},\ h_3 = \frac{1-\sqrt{3}}{4\sqrt{2}}$$

- Wavelet coefficients are:

$$g_0 = h_3,\ g_1 = -h_2,\ g_2 = h_1,\ g_3 = -h_0$$

IMR-Pathload (5)

- Assume that a sequence s_0, s_1, …, s_{n-1} is an input to the j-th stage filters
- Then, $cA_{j,k}$ and $cD_{j,k}$ are given by:

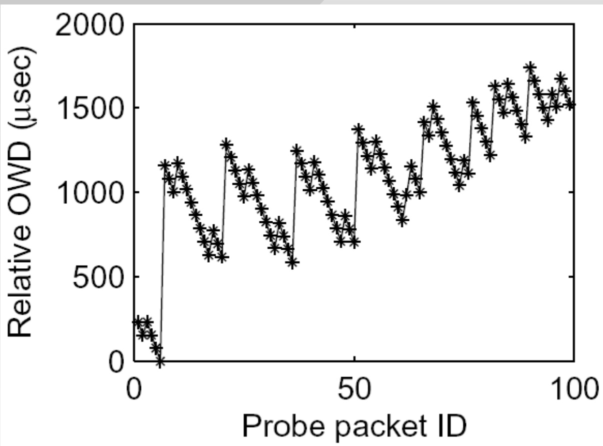
$$cA_{j,k} = h_0 s_{2k} + h_1 s_{2k+1} + h_2 s_{2k+2} + h_3 s_{2k+3}$$

$$cD_{j,k} = g_0 s_{2k} + g_1 s_{2k+1} + g_2 s_{2k+2} + g_3 s_{2k+3}$$

IMR-Pathload (6)

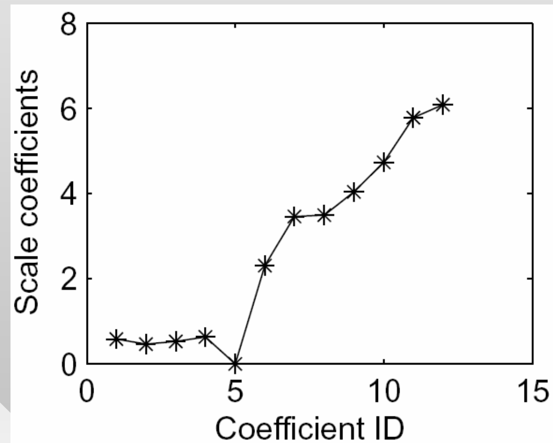
- Effect of de-noising on trend detection

original OWDs



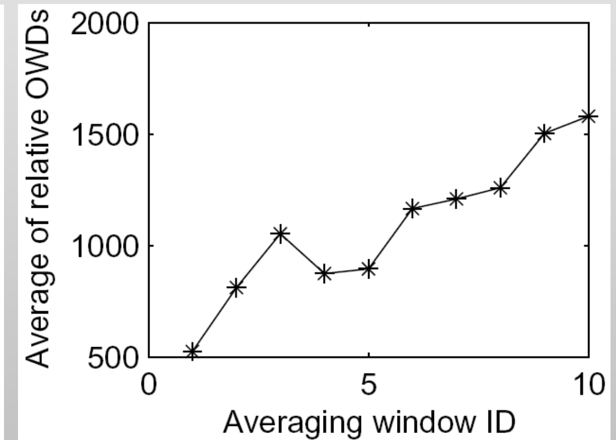
PCT=0.5, PDT=0.11

wavelet decomposition



PCT=0.75, PDT=0.78

window averaging



PCT=0.8, PDT=0.74

Pathload: unable to detect increasing trend

IMR-Pathload: able to detect increasing trend accurately

Agenda

- Introduction
- Interrupt moderation
- Analysis of Pathload
  - Impact of interrupt delays
  - Trend detection problem
- IMR-Pathload
- Performance evaluation
- Wrap-up

Performance Evaluation

- Emulab experiments
  - Investigate estimation accuracy of IMR-Pathload under a wide range of interrupt delays
  - Main metric is the relative estimation error e_A
- Internet experiments
  - Measure Internet paths between several sites in US
  - Show how reliably IMR-Pathload measures Internet paths compared to the original Pathload

Emulab Experiment

Estimation method	Interrupt delay δ	Evaluation scenario					
		Case-I	Case-II	Case-III	Case-IV	Case-V	Case-VI
IMR-Pathload (wavelet)	0 μs	2.46%	1.23%	3.47%	2.69%	3.71%	6.52%
	100 μs	6.47%	4.5%	3.02%	4.42%	5.98%	12.17%
	125 μs	7.21%	2.64%	3.88%	1.32%	6.1%	10.77%
	500 μs	5.12%	2.17%	6.78%	3.24%	7.23%	5.56%
IMR-Pathload (average)	0 μs	2.07%	2.24%	2.1%	2.18%	9.67%	5.05%
	100 μs	0.19%	0.71%	11.69%	1.32%	4.19%	6.82%
	125 μs	1.44%	1.82%	12.58%	1.59%	2.64%	7.89%
	500 μs	4.43%	4.59%	9.27%	2.55%	8.95%	6.48%

- IMR-Pathload produces available bandwidth estimates for all cases with 88-99% accuracy
- Even with a large interrupt delay $\delta = 500\ \mu s$, it measures the paths within e_A=10% error
  - Recall that the original Pathload can measure none of the paths when $\delta > 125\ \mu s$

Internet Experiment

- Measure each path during 5 different periods of time in a day
  - Run both tools 3 times for each time period over a particular path
- If a tool can measure a path in all 3 times for a period, we report their average as its bandwidth estimate
- If a tool fails to measure a path at lest once in 3 trials, we consider that the tool cannot reliably measure that particular path during that period

Internet Experiment (2)

Internet paths	Method	Available bandwidth estimates (Mb/s)				
		9 – 10 am	12 – 1 pm	3 – 4 pm	7 – 8 pm	11 – 12 pm
HP → Wustl	IMR-Pathload	12.2	11.9	13	12.8	13.1
	Pathload	−−	−−	−−	−−	−−
UMD → HP	IMR-Pathload	93	92.8	92.3	93.2	94.7
	Pathload	95.1	91.7	91.2	93.2	92.6
UMD → TAMU	IMR-Pathload	100	98.1	98.3	99.4	98.4
	Pathload	−−	−−	−−	−−	−−
HP → UMD	IMR-Pathload	12.9	11.8	13.3	12.3	12.6
	Pathload	20	−−	16.9	−−	−−

Pathload cannot reliably measure these paths

Wrap-up

- Pathload exhibits estimation instability under non-negligible interrupt delays
  – Instability stems from the fact that its delay-trend detection mechanism is unreliable
- IMR-Pathload provides robust trend detection under a wide range of interrupt delays
  – Signal de-noising facilitates accurate trend-detection
- IMR-Pathload significantly improves measurement stability of the original Pathload under various network settings