Capacity estimation using packet dispersion techniques : recent progress and open issues

Ravi S. Prasad

and

Constantinos Dovrolis

Computer and Information Sciences

University of Delaware

Capacity estimation tools

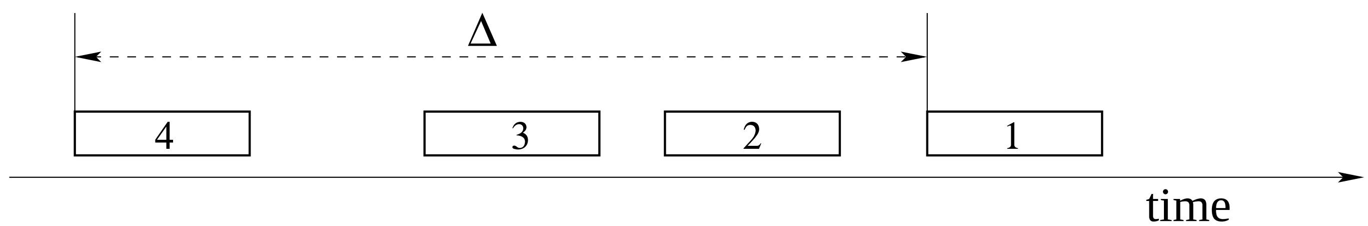
Tool	Author	Measurement objective	Methodology
pathchar	Jacobson	Per-Hop Capacity	Variable Packet Size
clink	Downey	Per-Hop Capacity	Variable Packet Size
pchar	Mah	Per-Hop Capacity	Variable Packet Size
bprobe	Carter	End-to-End Capacity	Packet Pairs
nettimer	Lai	End-to-End Capacity	Packet Pairs
sprobe	Saroiu	End-to-End Capacity	Packet Pairs
pathrate	Dovrolis	End-to-End Capacity	Packet Pairs & Trains
pipechar	Guojun	End-to-End Bottleneck	Packet Trains
cprobe	Carter	End-to-End Avail-BW	Packet Trains
pathload	Jain	End-to-End Avail-BW	Self-Loading Periodic Streams
TReno	Mathis	Bulk-Transfer-Capacity	Emulated TCP throughput
cat	Allman	Bulk-Transfer-Capacity	Standardized TCP throughput
IPerf	NLANR-DAST	Maximum TCP throughput	Parallel TCP streams

Overview

- Packet dispersion techniques : a review
  - Packet pair/train dispersion
  - Dispersion and capacity
  - Role of receiving host
- Important issues for high capacity paths
  - Instruction cache misses
  - Batched interrupt
  - NIC to userspace latency
    - Minimum measurable dispersion

Packet dispersion

- Inter-arrival time of first and last packet of a train



Δ : Packet Dispersion for a train of length 4.

- If packets arrive back-to-back, dispersion "set" by the capacity of the path
- May get affected by cross-traffic
  - All tools use some statistical method to discard wrong measurement
  - Cross-traffic effects are not considered here
    - ∗ see Dovrolis *et. al.* INFOCOM 2001

Packet dispersion and capacity estimation

- The dispersion after the bottleneck link

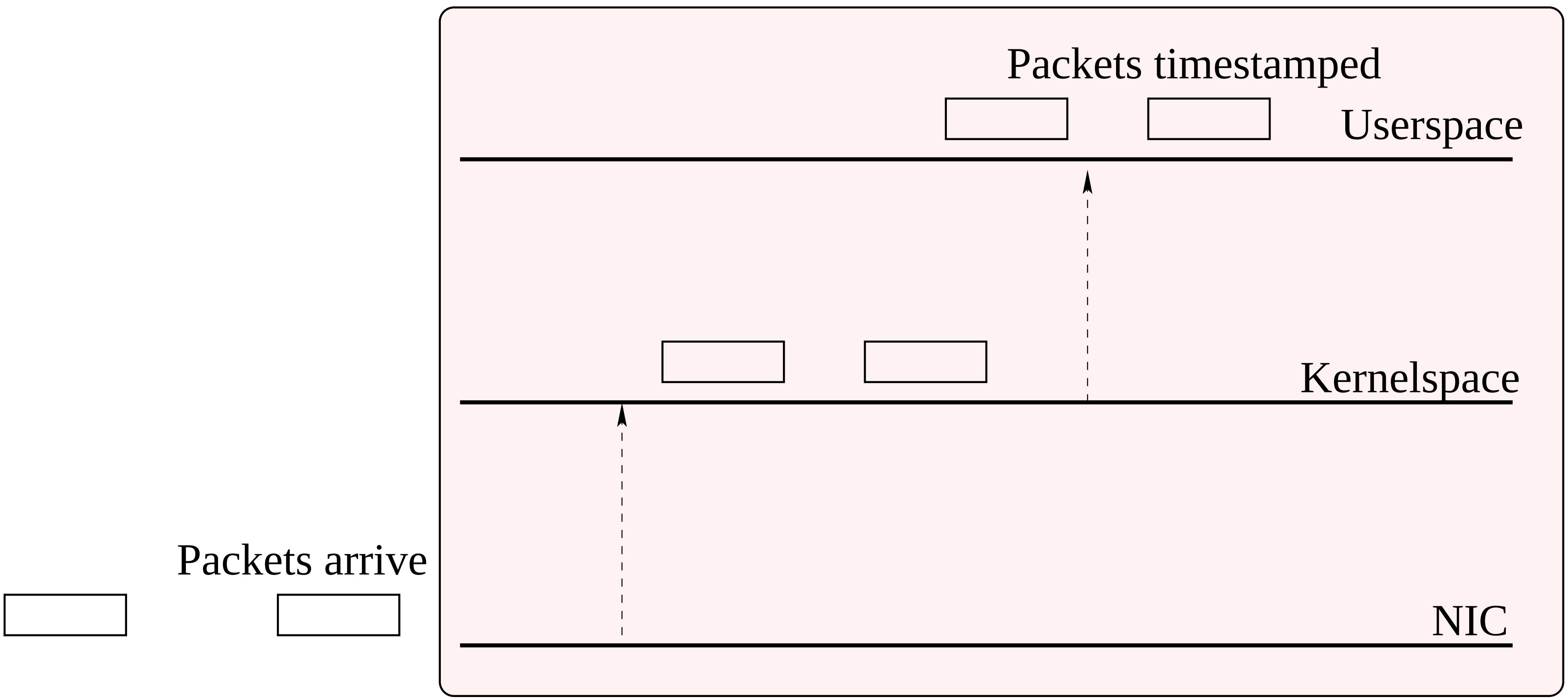
$$\Delta = \frac{(n-1)L}{C} \quad (1)$$

  - C : Capacity of the bottleneck link and path
  - n : Train length
  - L : Size of each packet

- For packet pair, n=2

$$\Delta = \frac{L}{C} \quad (2)$$

Role of receiving host



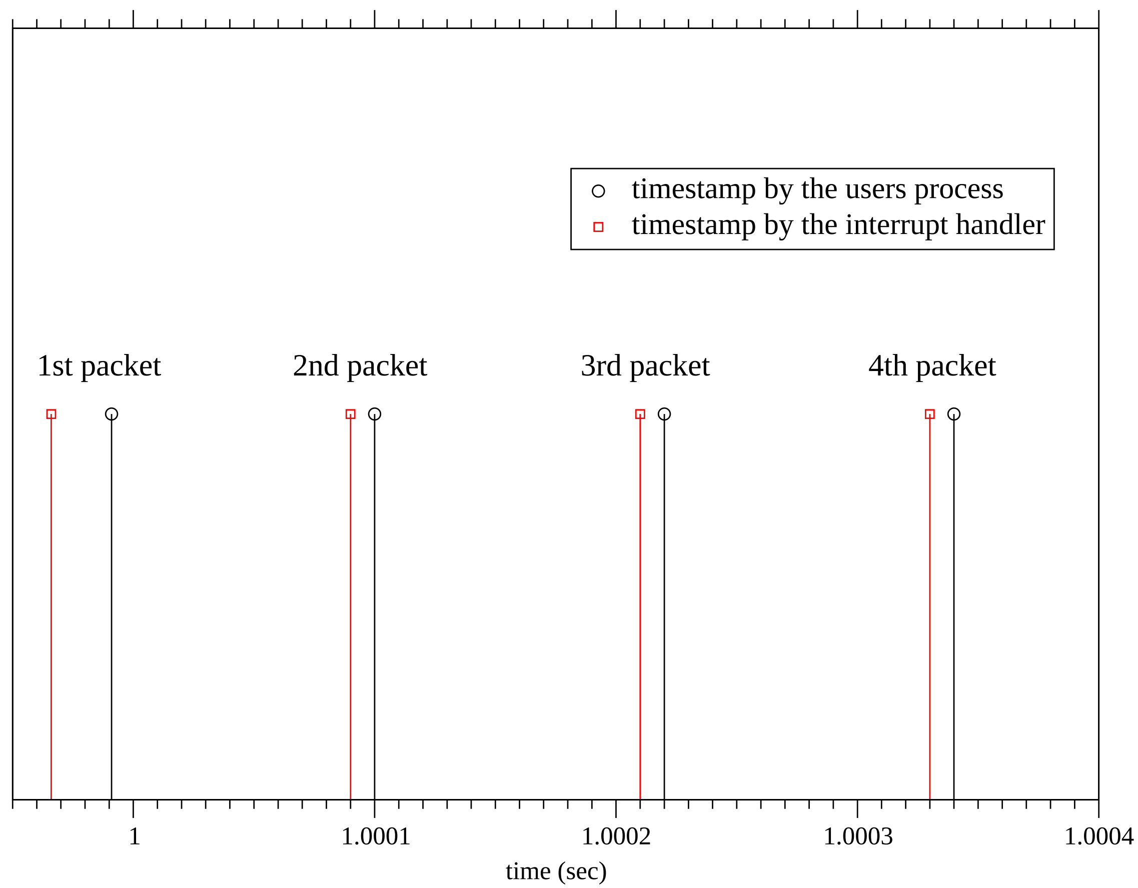
Questions

- Is dispersion preserved till timestamping?
  - If one packet takes more processing time than other?
- What is the minimum dispersion can be measured?
- What if processing takes more time than dispersion?

Important issues for high capacity paths

- Instruction cache misses
- Batched interrupt
- NIC to userspace latency

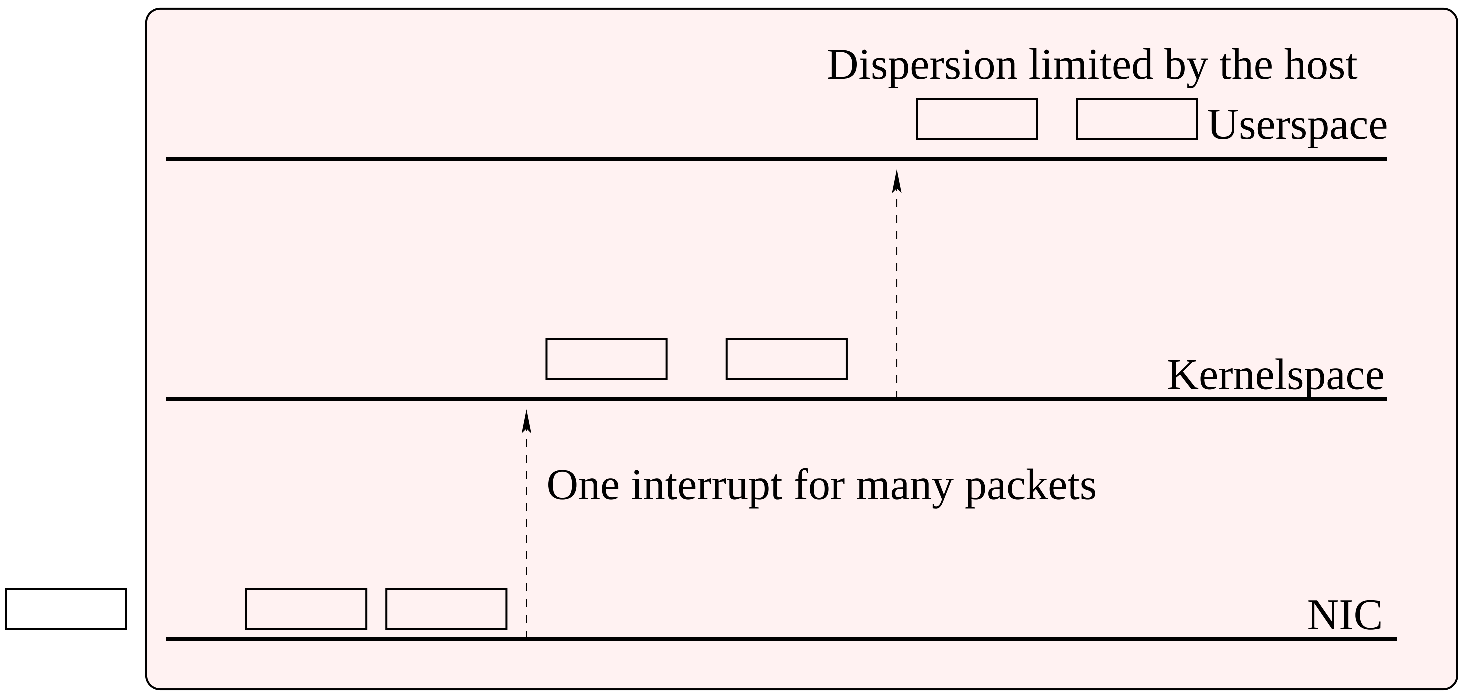
Instruction cache misses



- The first packet of each train needs different processing time
- “Incorrect” dispersion measured for the 1st packet-pair in each train
- pathrate sends an extra packet to initialize the cache.

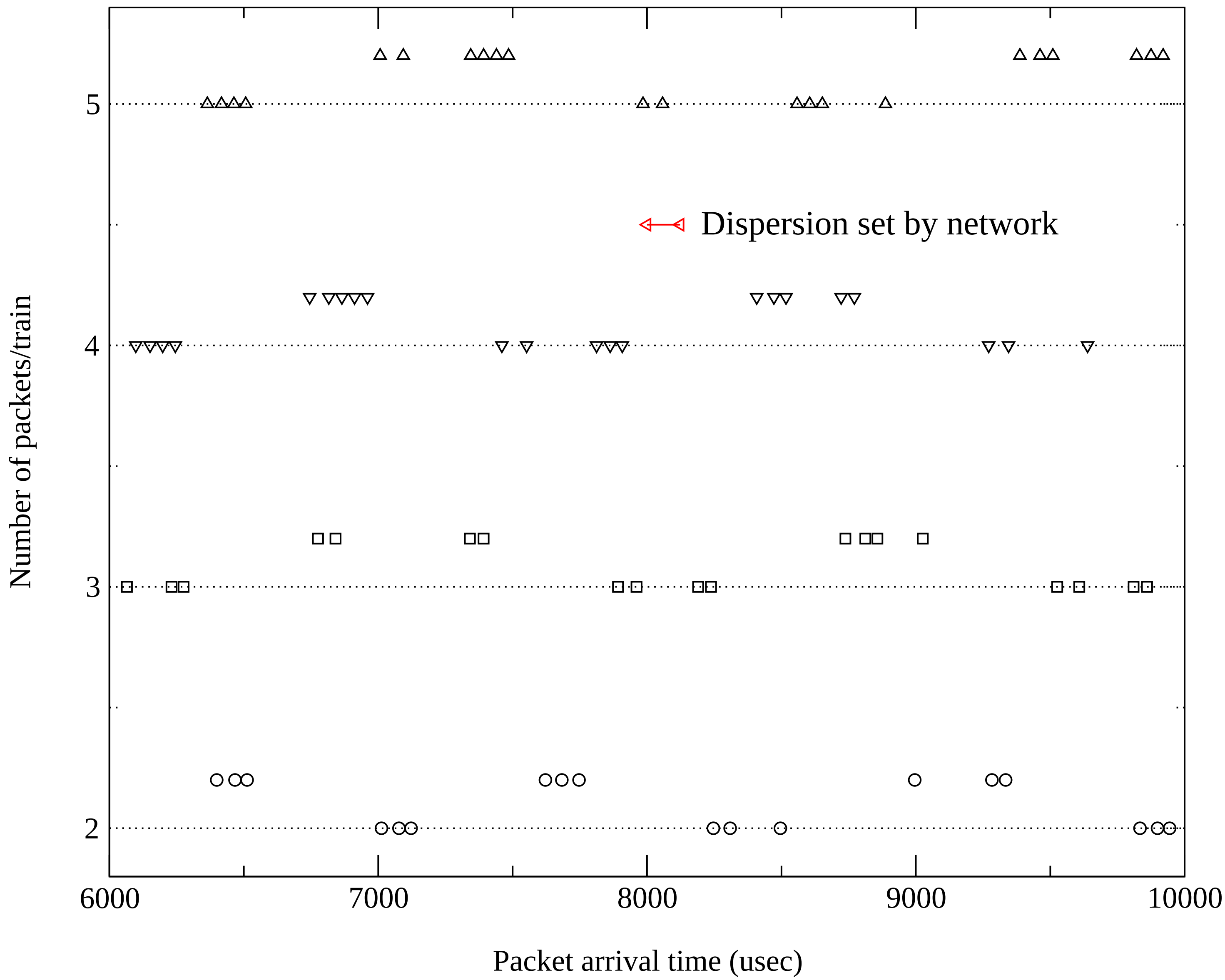
Batched interrupt

- If a packet arrives while interrupt handler still running?



- Smaller packets, slower hosts get affected

Batched interrupts : An example

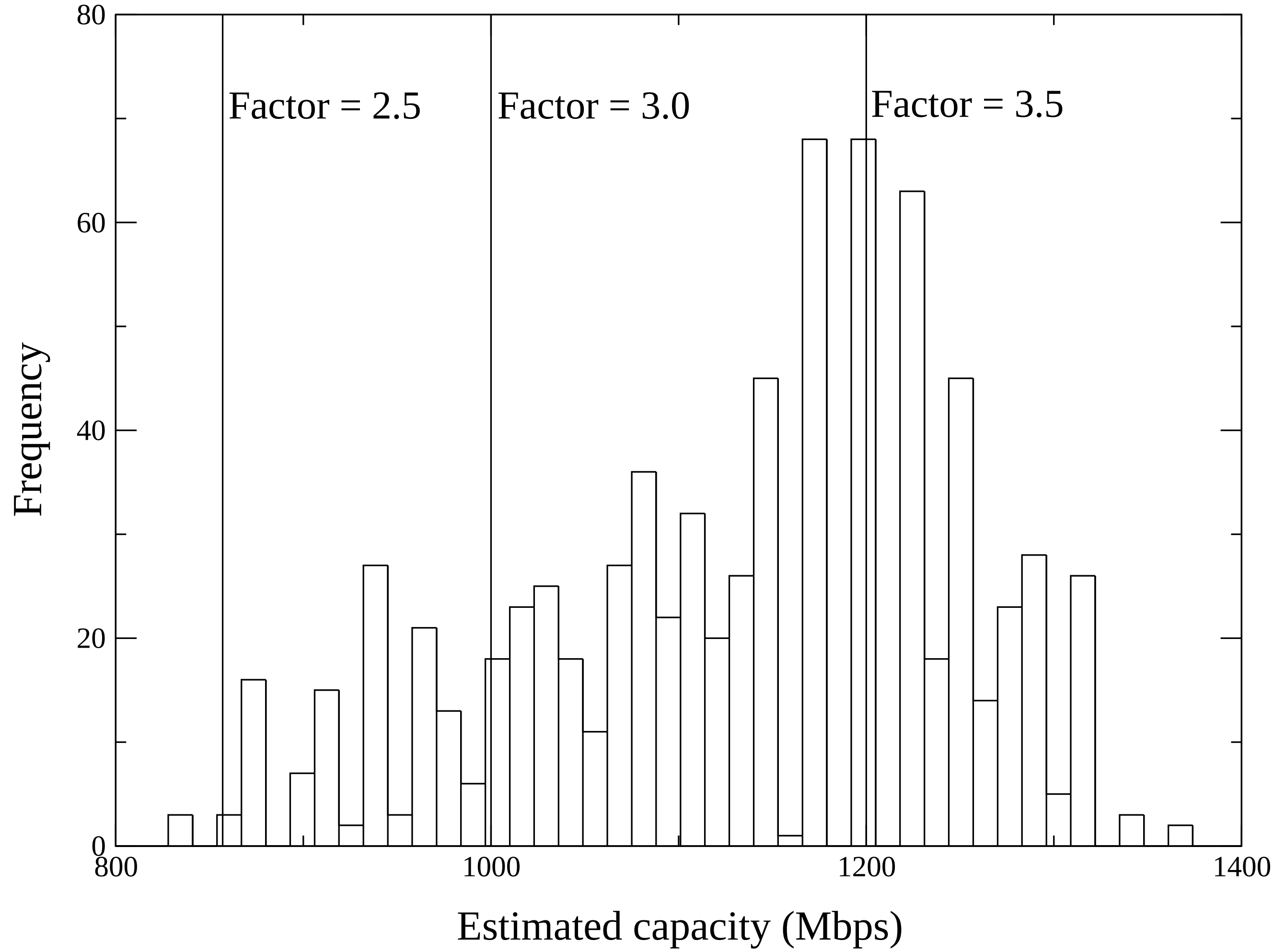


NIC to userspace latency

- The minimum dispersion that a host can measure
- To calculate it?
  - Calculate user to kernel latency
  - Multiple by a factor to get NIC to userspace latency
- To find this factor?

  For linux kernel 2.4.2 (with hacked interrupt handler),
  - Obtained interrupt handler to user process latency
  - Obtained kernel to user process latency
- This factor is assumed to be constant over different OSes.

Effect of minimum acceptable dispersion



Summary

- OS issues become important for high capacity paths
- Need a better way to measure minimum acceptable dispersion
- Data transfer rate may be limited by the hosts and not the network

Thank you!