Characterizing Tight-Link Bandwidth of Multi-Hop Paths Using Probing Response Curves

Seong Kang

Joint work with Dmitri Loguinov

LTE Systems Lab
Samsung Electronics, Seoul, Korea

June 18, 2010

Agenda

- Introduction
  - Motivation and goals
  - Definition of bandwidth
- PRC-MT
  - Basic idea and issues
  - Proposed approach
- Evaluation
  - With and without interrupt delay
- Conclusion

Motivation

- 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 existing approaches focuses on end-to-end measurements

Motivation 2

- Existing approaches can be classified into measurement tools and theoretical models

– Measurement tools

  - Usually based on extensive simulation
  - But, no convergence analysis with general cross-traffic

– Theoretical models

  - Usually have provable convergence
  - But, no practical implementation

- In addition, OS and hardware-related timing irregularities make delay measurements not perfect

– Cause unknown performance issues in real networks

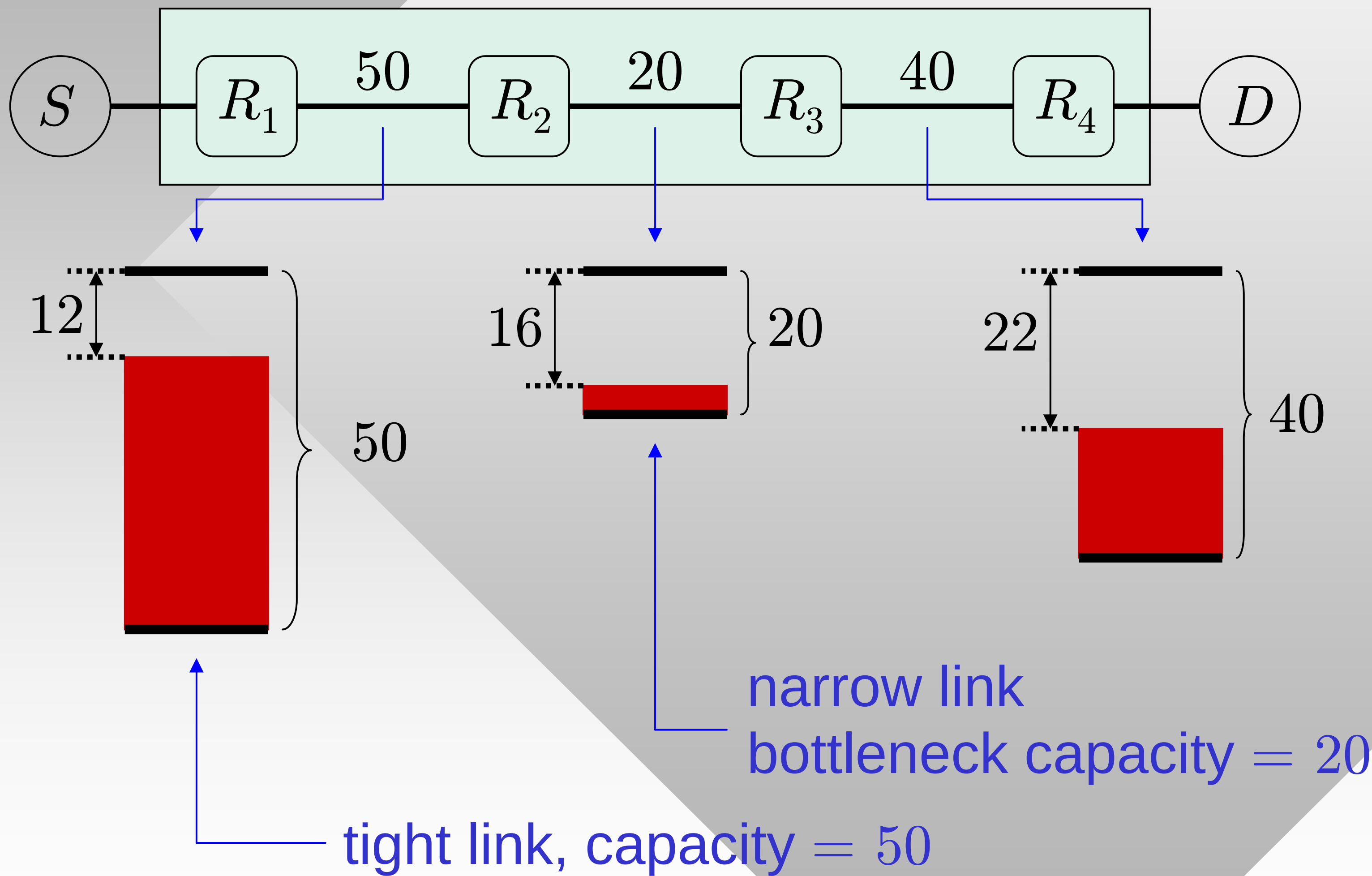
Goals

- Develop a practical measurement tool based on a recent theoretical model
- Achieves asymptotic accuracy in multi-hop path networks with arbitrary cross-traffic
- Simultaneously measures both the capacity and available bandwidth of the tight link
- Robust to various timing irregularities

Definition of Bandwidth 1

- Bottleneck bandwidth
  - Capacity of the slowest link of a path
  - The slowest link is often called *narrow* link
- Available bandwidth
  - The smallest unused bandwidth of links in the path
  - The link with the smallest unused bandwidth is called *tight* link

Definition of Bandwidth 2



- Available bandwidth of the path: $A = 12$

PRC-MT

- PRC-MT is a practical implementation that exploits certain characteristics of input probing rate r_I and corresponding arrival rate r_O
  - r_I represents the average sending rate of packets in a probe-packet train of size N
  - r_O represents the average rate of probe packets arriving at the receiver
- PRC-MT utilizes the concept of Probing Response Curve (PRC), which is a functional relationship between r_I and r_I / r_O

Basic Idea

- Define F to be the ratio of r_I and r_O under fluid cross-traffic

$$F = \frac{r_I}{r_O} = \begin{cases} 1 & r_I \le A_t \\ \frac{\lambda_t + r_I}{C_t} & A_t < r_I \le B \end{cases}$$

tight-link available bandwidth

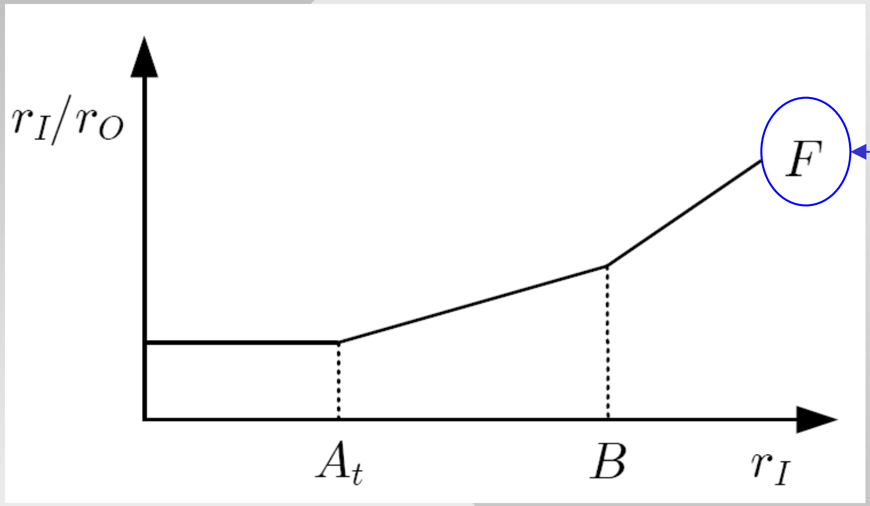
amount of cross-traffic that traverses the tight link

tight-link capacity

a certain input rate that is no less than the second smallest available bandwidth of the path

Basic Idea 2

- Hypothetical fluid response curve F

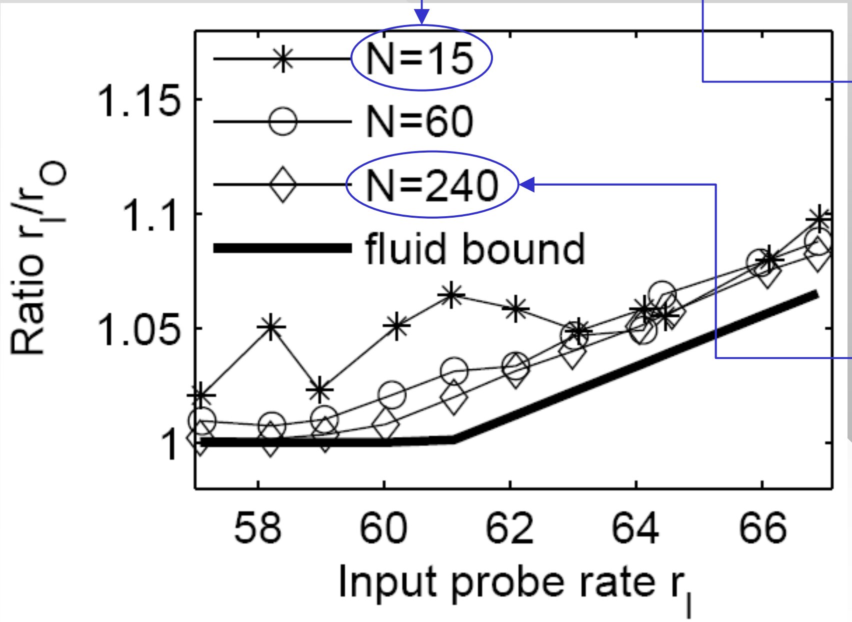


consists of piece-wise linear segments (at least two in a multi-link path)

- Note that A_t is identified by the first break point
- C_t can be extracted by taking inverse of the slope (α) of the second line segment
  - $C_t = 1/\alpha$

Practical Issues

- Real response curve Z is different from the fluid counter part F (with finite probe-train size N and finite probe packet size q
  - $Z - F > 0$ in real networks



When probe-train length N is small, Z fluctuates substantially and exhibits large deviation from F

When N increases, Z shows prominent two linear lines and its deviation from F becomes smaller

Practical Issues 2

- For accurate discovery of the first break point, variation in Z should be small for different r_I
- More importantly, the second line segment should be parallel to that in F
  - If the second segments in Z and F are parallel, then we can use any two points on the second segment in Z to compute its slope
- However, fluctuation of line segments in Z depends on probe parameters and unknown path characteristics

Practical Issues 3

- Hence, selection of path-specific N for a given probe-packet size is very important
- N needs to be large enough to obtain reliable line segments
  - However, it is desirable to have as small N as possible to reduce measurement overhead and avoid too much packet loss within a probe train
- In addition, building PRC requires substantial probing overhead since A_t could be anywhere between 0 and C_t

Proposed Approach

- Probe for path-specific probing parameters
  - Probe for an initial input rate r_I with a single-probe train
  - Employ iterative probing for packet-train length N in binary search fashion
- Do not build entire PRC to reduce probing overhead
  - Instead employ binary search-like iterative probing for available bandwidth A_t
  - Extract tight-link capacity C_t without additional probing

Parameter Selection

- First, probe for an initial rate r_I by sending a single probe-train
  - Compute $r_I = q/\mathrm{E}[y]$

average inter-packet dispersion of packets in the probe-train

- Next, note that r_I/r_O saturates at a certain value as N becomes large

$$r_I/r_O \approx \frac{\lambda_t + r_I}{C_t}$$

- Thus, iteratively probe for the smallest N based on a binary search between N_{min} and N_{max}, while keeping the variation of r_I/r_O within a certain threshold

Bandwidth Probing

- Available bandwidth (A_t) – iterative probing
  (1) In k-th iteration, sends probe-trains with rate $R(k)$
  (if $k = 0$, $R(k)$ is set to the initial r_I)
  (2) If asserted to be $R(k) > r_O$, then update the upper bandwidth bound $W_H = R(k)$ and records $(R(k)$, $r_O)$ pair
  (3) If asserted to be $R(k) < r_O$, then update the lower bandwidth bound $W_L = R(k)$
  (4) Compute $R(k{+}1) = (W_H + W_L)$ / 2 for next iteration
  (5) If W_H – $W_L \leq$ threshold, return $A_t = (W_H + W_L)/2$
  (6) Otherwise, repeat steps (1) – (4)

Bandwidth Probing 2

- Capacity (C_t)

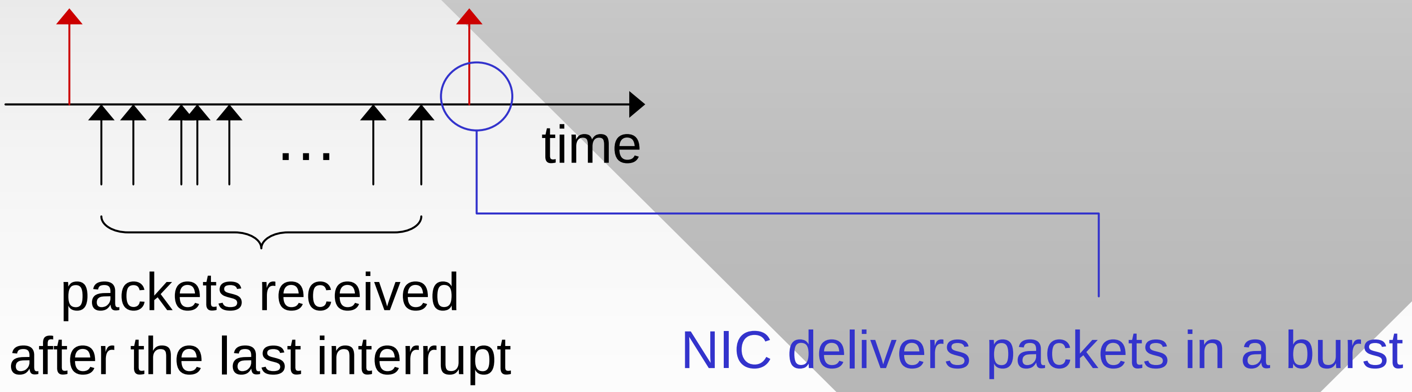
(1) Select two $(R(k), r_O)$ pairs recorded in the A_t probing step (2), which is the farthest two points that satisfy $R(k) \geq W_H$

(2) Compute the slope (α) of the line segment connecting the two points

(3) Then, return $1/\alpha$ as an estimate of C_t

Interrupt Delays

- Interrupt moderation is widely used with modern Gigabit network cards
  - Reduce CPU utilization and increase network throughput
- At a single interrupt, NIC delivers multiple packets to the kernel



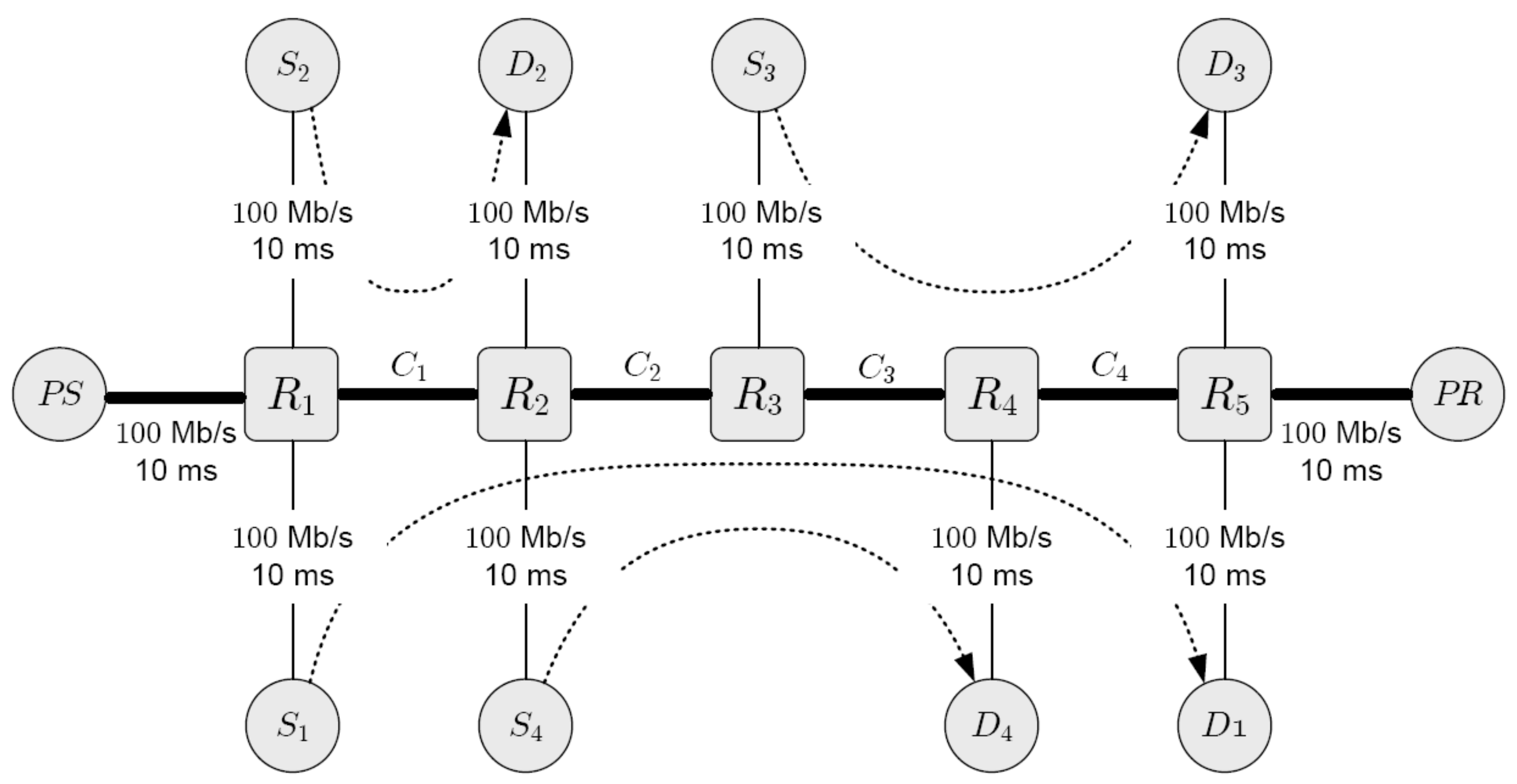
Interrupt Delays 2

- A wide range of interrupt delays with Intel Gigabit NICs
  - 83 – 250 usec for Windows
  - 125 – 1000 usec for Linux
- Some study suggests at least 470 usec delay to achieve good throughput and substantially reduce CPU utilization

Impact of Interrupt Delays

- IGI/PTR, Spruce, CapProbe
  - Not accurate regardless of an interrupt delay (δ)
- Pathchirp
  - Accurate, but requires substantially more probe data with non-trivial δ
  - Prolonged measurement duration
- Pathload
  - Accurate with small delays (less than 125 usec)
  - Becomes unreliable when $\delta > 125$ usec
- IMRP significantly improves Pathload's estimation reliability
  - Use wavelet-based signal de-noising

Evaluation Topology



Experimental Setup

	Different link bandwidths (Mb/s)							
	C_1	A_1	C_2	A_2	C_3	A_3	C_4	A_4
Case-I	75	31.8	90	51.6	90	42.1	[60]	40.7
Case-II	75	41.3	90	70.7	90	46.7	[60]	26.4
Case-III	[60]	35.8	90	70.7	[90]	23.4	75	18.1
Case-IV	[60]	21.6	90	65.9	90	42.1	75	36.7
Case-V	[60]	50.2	90	61.1	90	41.9	75	50.8
Case-VI	75	28.9	90	37.8	90	13.8	[60]	31.2

- Shaded values represent tight-link capacity C_t and available bandwidth A_t of the path for each case
- Values in square brackets are the capacities C_n of the narrow link

Performance Comparison

- Under no interrupt delay

Evaluation scenario	Relative estimation error e_A							
	PRC-MT		Pathload		Pathchirp		IGI / PTR	
	e_A	time	e_A	time	e_A	time	e_A	time
Case-I	3.49%	89 sec	9.45%	69 sec	10.84%	200 sec	10.58/16.02%	3 sec
Case-II	2.35%	115 sec	8%	69 sec	8.53%	200 sec	4.21/9.93%	4 sec
Case-III	0.88%	96 sec	7.57%	70 sec	0.39%	200 sec	72.76/30.28%	5 sec
Case-IV	5.05%	138 sec	6.48%	69 sec	1.62%	200 sec	19.72/24.63%	6 sec
Case-V	5.51%	102 sec	16.58%	108 sec	19.81%	200 sec	13.38/5.31%	3 sec
Case-VI	9.74%	125 sec	15.01%	99 sec	18.04%	200 sec	98.56/59.24%	5 sec

Available bandwidth estimation error

Methods	Relative estimation error e_C			
	Case-II		Case-IV	
	e_C	time	e_C	time
PRC-MT	2.52%	115 (sec)	3.51%	138 (sec)
Pathrate	28.33%	2191 (sec)	21.67%	2191 (sec)
CapProbe	47.32%	500 (sec)	63.38%	500 (sec)

Capacity Estimation Error

Performance Comparison 2

- Interrupt delay: 500 usec

Evaluation scenario	Relative estimation error e_A									
	PRC-MT		IMR-Pathload		Pathload		Pathchirp		IGI / PTR	
	e_A	time	e_A	time	e_A	time	e_A	time	e_A	time
Case-I	3.71%	90 sec	5.12%	88 sec	−−	−−	7.22%	200 sec	62.34/22.8%	4 sec
Case-II	3.83%	89 sec	2.17%	89 sec	−−	−−	13.57%	200 sec	62.37/13.83%	4 sec
Case-III	1.55%	133 sec	6.78%	95 sec	−−	−−	5.14%	200 sec	44.14/44.81%	4 sec
Case-IV	0.19%	89 sec	3.24%	99 sec	−−	−−	13.01%	200 sec	59.03/21.25%	5 sec
Case-V	5.81%	92 sec	7.23%	79 sec	−−	−−	11.26%	200 sec	69.21/1.64%	3 sec
Case-VI	5.56%	96 sec	5.56%	80 sec	−−	−−	3.54%	200 sec	29.15/67.68%	5 sec

Available bandwidth estimation error

Methods	Relative estimation error e_C			
	Case-II		Case-IV	
	e_C	time	e_C	time
PRC-MT	1.72%	89 (sec)	7.65%	86 (sec)
Pathrate	17.5%	2191 (sec)	18.33%	2191 (sec)
CapProbe	57.65%	500 (sec)	81.77%	500 (sec)

Capacity Estimation Error

Conclusion

- Proposed a new bandwidth measurement tool called PRC-MT
  - Exploits characteristics of probing response curves
  - Extracts both bandwidth metrics of the tight link over multi-hop paths
- Evaluated PRC-MT with other existing tools under non-trivial interrupt delay
  - PRC-MT produced available bandwidth and capacity estimates with high accuracy
  - Timing irregularity caused by interrupt moderation significantly affects performance of tools such as Pathload

Thank you!