

Median Filtering Simulation of Bursty Traffic

Auc Fai Chan, John Leis
 Faculty of Engineering and Surveying
 University of Southern Queensland
 Toowoomba Queensland 4350

Abstract—The estimation of Retransmission Timeout (RTO) in Transmission Control Protocol (TCP) affects the throughput of the transmission link. If the RTO is just a little larger than the Round Trip Time (RTT), retransmissions will occur too often, and this increases congestion in the transmission link [1]. If the RTO is much larger than the RTT, the response to retransmit when a packet is lost will be too slow, and this will decrease the throughput in the transmission link. Currently, Jacobson’s Algorithm [2] for estimation of RTO is implemented in TCP. He uses an Exponential Weighted Moving Average (EWMA) filter to estimate RTT and then determines RTO. The EWMA filter is good if RTT follows a Gaussian distribution. In reality, traffic on the Internet is bursty and follows a heavy-tailed distribution. Median filter has been recognized as a useful filter due to its edge preserving characteristic for image processing applications, and it performs well for heavy-tailed distributions. Thus the median filter is efficient for removing impulsive noise [3], [4]. In this paper, we demonstrate the applicability of the median filter in our simulations for timeout estimation in the presence of bursty traffic flows.

Index Terms—median filter TCP congestion Pareto exponential

I. INTRODUCTION

THE TCP specification that specifies the first original algorithm is RFC 793, which was published in 1981 [5]. The first original algorithm, after several years of implementation on the Internet, was found to have ambiguity in the measurement of Round Trip Time (RTT) that was used when the packets were retransmitted. In 1987, Phil Karn and Craig Partridge found that TCP was suffering from a problem they called retransmission ambiguity. They presented a novel and effective method to clarify this retransmission ambiguity problem, in their paper, “Improving Round-Trip Time Estimates in Reliable Transport Protocols”, [6]. Their method could mitigate congestion in Internet, but, it was Jacobson and Karels, who later addressed congestion on Internet more effectively. In October, 1986, Van Jacobson reported the first Internet congestion collapse [2]. The data throughput from Lawrence Berkeley Laboratory to University of California Berkeley, which are 400 yards apart, dropped from 32 Kbps to 40 bps. In 1988, Van Jacobson published his paper, “Congestion Avoidance and Control”, [2]. Jacobson developed the congestion avoidance mechanisms that are now incorporated by almost all TCP implementations. All the TCP implementations are based on the four intertwined congestion control algorithm, slow start, congestion avoidance, fast retransmit and fast recovery. Currently, Jacobson Algorithm for estimation of Retransmission Timeout (RTO) is implemented in TCP. Jacobson Algorithm uses Exponential Weighted Mov-

ing Average (EWMA) filters to estimate RTT and then determine RTO. EWMA filter is good for signals that exhibit Gaussian distribution. In reality, RTT’s are often impulsive, and follow the heavy tailed distribution. Using EWMA Filter to estimate heavy tailed distribution is inadequate. In digital image processing, Median filter is known to eliminate positive and negative impulses [3], [4]. Therefore, median filter can do better when dealing with heavy-tailed distribution. Thus we employ Median Filter in our simulations for timeout estimation in the presence of bursty traffic flows.

The focus of this research is to apply median filter to network simulations for retransmission timeout estimation, in the presence of bursty traffic flows. The contribution of this research is that our simulation results show that, the median filter can perform better than the EWMA filter, in terms of throughput and packet drop percentage, i.e. the median filter delivers more throughput and loses less packets.

II. PROBLEMS WITH RTT ESTIMATION

Jacobson Algorithm includes firstly finding the difference between the sample RTT and the old Estimated RTT.

$$Difference = M_{i-1} - A_{i-1} \quad (1)$$

A_{i-1} Old Estimated RTT
 M_{i-1} Sample RTT [7], [8]

Secondly, find the new Estimated RTT

$$A_i = A_{i+1} + (G_1)(Difference) \quad (2)$$

$$= A_{i+1} + G_1(M_{i-1} - A_{i-1}) \quad (3)$$

A_i New Estimated RTT

G_1 is a constant, which is typically equal to 1/8 [7]–[9]. The last expression 3 above states that we make a new prediction (A_i), based on the old prediction (A_{i-1}), plus a fraction ($G_1 = 1/8$) of the prediction error ($M_{i-1} - A_{i-1}$).

The prediction error is the sum of two components:

- 1) Error due to noise in the measurement, which is random and unpredictable, such as fluctuations in competing traffic. We denote this part by E_r .
- 2) Error due to a poor choice of A_{i-1} . We denote this part by E_e .

$$\text{Then } A_i = A_{i-1} + G_1 E_r + G_1 E_e$$

The $G_1 E_e$ term moves A_i in the correct direction while $G_1 E_r$ moves A_i in a random direction [2]. If A_i follows a Gaussian distribution (also called normal distribution), $G_1 E_r$

will cancel one another after a number of samples, and A_i will converge to the correct value.

However, in 1995, two researchers Paxson and Floyd found that FTP data connections had bursty arrival rate. In addition, the distribution of the number of bytes in each burst has a heavy right tail [10].

In statistic, heavy-tailed distributions are probability distributions whose tails are not exponentially bounded [11]. That is, they have heavier tails than the exponential distribution. One of the simplest heavy-tailed distributions is the Pareto distribution [12]. In the next section, we introduce the probability density functions (pdf) of exponential distribution and Pareto distribution and the similarity between them.

III. PROBABILITY DENSITY FUNCTIONS OF EXPONENTIAL AND PARETO DISTRIBUTION

Exponential distribution [13] is defined as:

$$N(x) = \begin{cases} \frac{N_0}{e^{\alpha x}} & \text{for } x \geq 0 \\ 0 & \text{elsewhere} \end{cases} \quad (4)$$

In order to see the similarity between exponential distribution and Pareto distribution, we simplify equation 4 as follows:

- $e = 2.7$ (to one place of decimal)
- Let $\alpha = 1$ (Probability can be 1)

Then, $N(x) = N_0/2.7^x$ where N_0 , the value of $N(x)$ at time $x = 0$, is a constant. Later, we show that one example of Pareto distribution is $f(x) = Constant/x^{2.7}$.

Now we introduce Pareto distribution. The definition of Pareto distribution [14] is:

$$f(x) = \begin{cases} \frac{\alpha k^\alpha}{x^{\alpha+1}} & \text{for } x > k \\ 0 & \text{elsewhere} \end{cases} \quad (5)$$

Let us consider the simple case where $k = 1$ and $\alpha = 1.7$. We have

$$f(x) = \begin{cases} \frac{1.7}{x^{2.7}} & \text{for } x > 1 \\ 0 & \text{elsewhere} \end{cases} \quad (6)$$

We see that there is similarity between exponential distribution and Pareto distribution. The parameter k specifies the minimum value of x . The parameter α determines the shape of $f(x)$. In the simulations of Pareto traffic, we set the shape of Pareto, so we are actually setting the value of α . We plot exponential distribution and Pareto distribution in Fig.1 for comparison.

In Fig.1, we see that Pareto distribution decreases much more slowly than exponential distribution, or we say that, Pareto distribution has a heavier tail than exponential distribution.

In 1997, Willinger *et al.* [16] generated an Ethernet traffic by superposition of many Pareto traffic sources, setting them ON and OFF. During ON period, the source transmits a burst of packets. During OFF period, the source is in idle and no

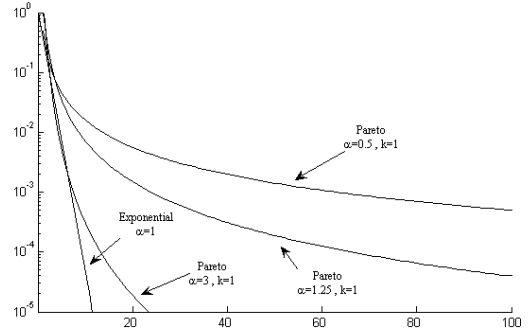


Fig. 1: Exponential and Pareto probability density functions [15].

packets are transmitted. We also set ON, OFF and shape of the Pareto traffic in our simulations.

Weighted-average is good if RTT follows a Gaussian distribution. In reality, traffic on Internet is bursty and follows a heavy-tailed distribution. Median filter is known in image processing to eliminate salt and pepper noise which is positive and negative impulses. Therefore, median filter can do better when dealing with heavy-tailed distribution.

IV. SIMULATION ENVIRONMENT

The Network Simulator, version 2 (NS2) is used in the following simulations. In both Case 1 and Case 2, dumbbell topology is used with 6 pairs of nodes. All the links have the same bandwidth. The bottleneck link is 1Mb and the queue type is DropTail. For simplicity, we consider only TCP traffic which is a one-way flow between source nodes and destination nodes.

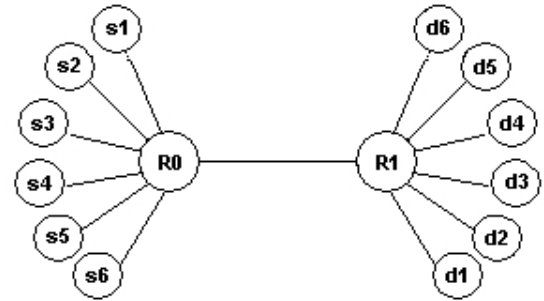


Fig. 2: Topology for Simulation Environment Case 1 and 2.

Case 1: The traffic source is exponential. There are 6 sources and 6 destinations. s_1 sends packets to d_1 ; s_2 sends packets to d_2 , and so on. s_1 through s_6 are TCP agents. d_1 through d_6 are TCP-sink agents. All links from sources to R0 are duplex links, with bandwidth of 10 Megabits, delay of 10 ms, and DropTail queues. R0R1 is a duplex link, with bandwidth of 1 Megabits, delay of 100 ms and Drop Tail queues. Queue size (the buffer size of routers R0 and R1) is set to 40. The aim to set R0R1 with bandwidth 1 megabits is to make R0R1 a bottleneck which implies a congested link. Later, it will be

shown that median filter outperforms the weighted-average filter under this congested situation.

Case 2: The traffic source is Pareto. The bandwidth and queue type are the same as Case 1. The parameter α , which determines the shape of Pareto distribution, is set to 2.5. The ON period is set to 1 second, while the OFF period is set to 2 seconds. (Please refer to the end of Section III)

V. SIMULATION RESULTS

Case 1 : Exponential traffic After the simulation, the accumulative goodputs are plotted in Fig.3.

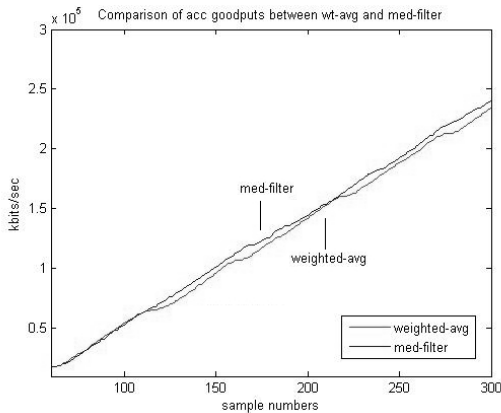


Fig. 3: Comparison of accumulative goodputs in exponential traffic.

Table I: Exponential Traffic Analysis.

	<i>throughput</i>	<i>goodput</i>	<i>loss%</i>
Original TCP	753440	740960	1.66%
Median filter of size 5	760240	752160	1.06%

From Table I, two improvements made by median filter of size 5 are listed below:

- 1) The total goodputs from median filter of size 5 is 752160 while that from the original TCP is 740960. Therefore, median filter of size 5 can deliver more goodputs.
- 2) The loss percentage of median filter of size 5 is 1.06%, while that of the original TCP is 1.66%. Therefore, median filter of size 5 has lower loss percentage.

Next, RTT and RTO are plotted in the same figure for comparison. Fig.4 shows RTT and RTO from the original TCP, while Fig.5 shows RTT and RTO from median filter of size 5.

In Fig.4, the RTO obtained from weighted average algorithm is much larger than RTT so the response to retransmit when a segment is lost will be too slow; and this will decrease the throughput of the transmission path [1].

If RTO is just a little larger than RTT, retransmission will be too often, and that increases congestion in the transmission path [1]. A desirable RTO is one which is close to, and always larger than the RTT. In Fig.5, the RTO estimated by median filter is possible to increase throughput and at the same time decrease congestion in the transmission path.

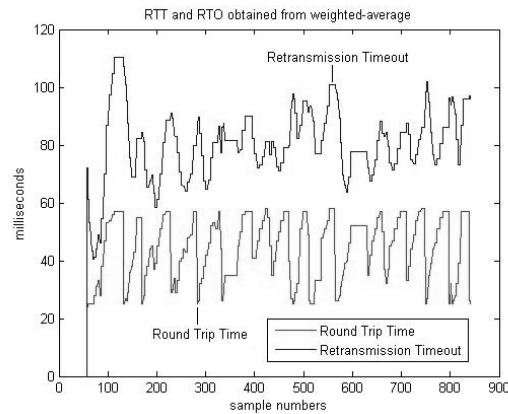


Fig. 4: RTT and RTO obtained from weighted-average in exponential traffic.

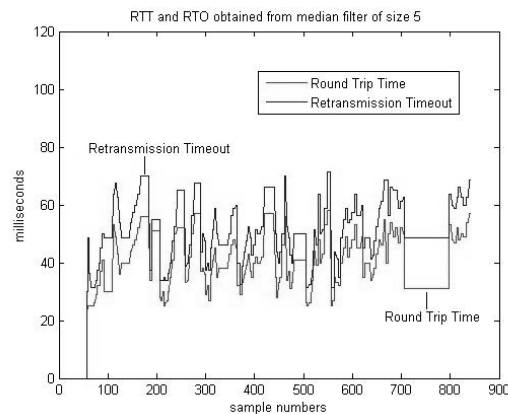


Fig. 5: RTT and RTO obtained from median filter of size 5 in exponential traffic.

In order to have a good understanding of median filter working in exponential traffic, the RTO values obtained from median filters of size 7 and size 9 are shown in Fig.6 and Fig.7.

The loss percentage in the transmission is calculated by the following formula: $\text{loss percentage} = (\text{throughput} - \text{goodput}) / \text{throughput} \times 100\%$

The loss percentages of median filters of size 5, size 7, size 9 and the original TCP are shown in Table II.

Table II: Comparison of loss percentages in exponential traffic.

	<i>throughput</i>	<i>goodput</i>	<i>loss%</i>
median filter size 5	760240	752160	1.06%
median filter size 7	653280	647360	0.91%
median filter size 9	728720	722480	0.86%
original TCP	753440	740960	1.66%

The conclusion is that, all loss percentages of median filters are lower than that of the original TCP.

Case 2 : Pareto traffic Similar to Case 1, the accumulative goodputs are plotted in Fig.8. In Fig.9 and Fig.10, RTT and RTO are plotted in the same figure for comparison.

From Table 3, two features in the comparison of goodputs are:

- 1) The goodput from median filter of size five is 748560,

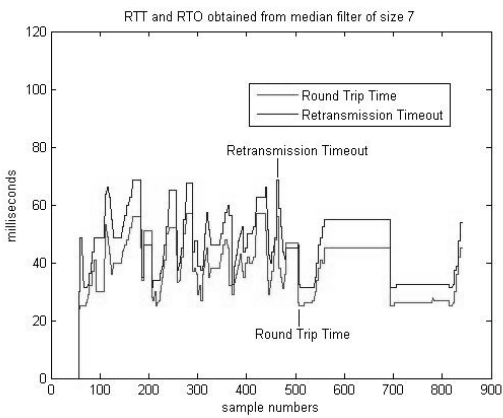


Fig. 6: RTT and RTO obtained from median filter of size 7 in exponential traffic.

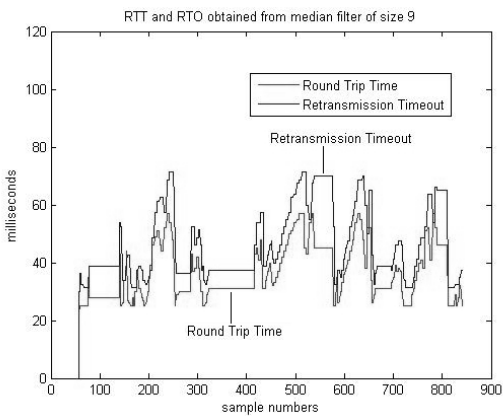


Fig. 7: RTT and RTO obtained from median filter of size 9 in exponential traffic.

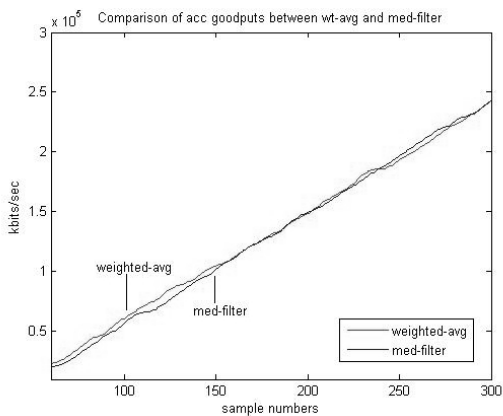


Fig. 8: Comparison of accumulative goodputs in Pareto traffic.

while that from the original TCP is 711760. Therefore, median filter can deliver more goodputs.

- 2) The loss percentage of median filter of size five is 1.58%, while that of the original TCP is 1.93%. Therefore median filter has lower loss percentage.

Fig.10 shows that the RTO from median filter of size 5, is better than the RTO in Fig.9 using weighted average estimation.

Table III: Pareto Traffic Analysis.

	<i>throughput</i>	<i>goodput</i>	<i>loss%</i>
Original TCP	725760	711760	1.93%
Median filter of size 5	760560	748560	1.58%

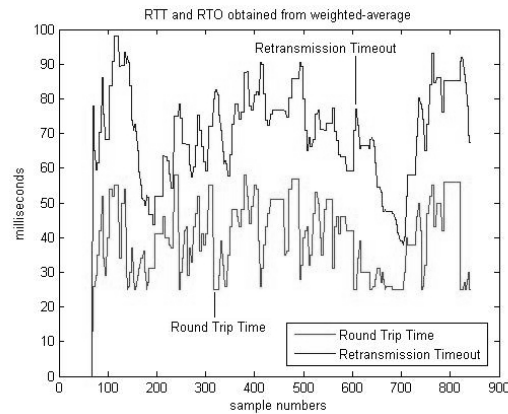


Fig. 9: RTT and RTO obtained from weighted-average in Pareto traffic.

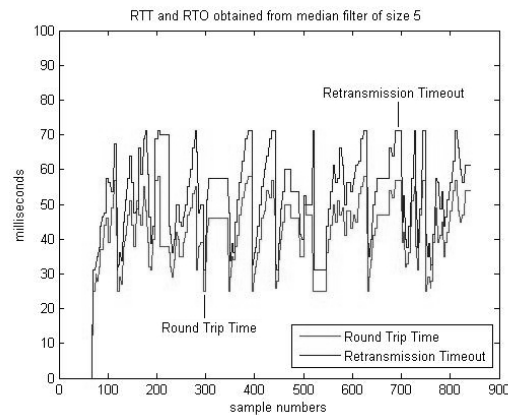


Fig. 10: RTT and RTO obtained from median filter of size 5 in Pareto traffic.

The RTOs obtained from median filters of size 7 and size 9 are shown in Fig.11 and Fig.12.

The loss percentages of median filters of size 5, size 7 and size 9 are shown in Table IV.

Table IV: Comparison of loss percentages in Pareto traffic.

	<i>throughput</i>	<i>goodput</i>	<i>loss%</i>
median filter size 5	760560	748560	1.58%
median filter size 7	676640	667520	1.35%
median filter size 9	756080	744400	1.54%

The conclusion for loss percentages is that, all loss percentages from median filters of sizes 5, 7, 9 are lower than the loss percentage from the weighted average estimation, which is 1.93%.

When comparing goodputs among median filters of sizes 5, 7, and 9, the goodput from median filter of size 5 is the highest.

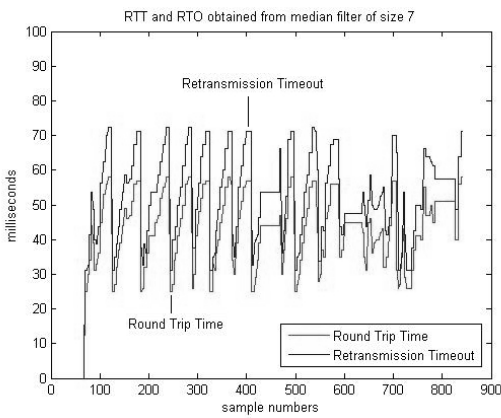


Fig. 11: RTT and RTO obtained from median filter of size 7 in Pareto traffic.

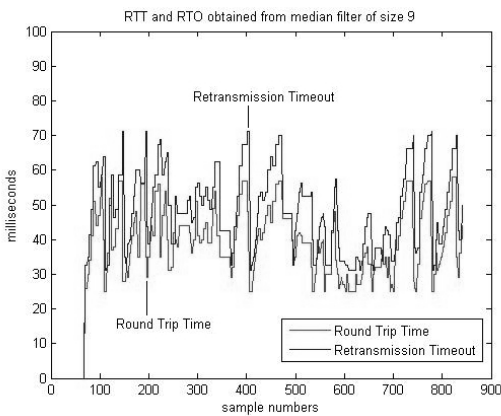


Fig. 12: RTT and RTO obtained from median filter of size 9 in Pareto traffic.

When Figure 9, which is RTT and RTO obtained from weighted-average, is compared with Figure 10, which is RTT and RTO obtained from median filter of size 5, it is found that in the case of weighted-average, the curve of RTO is further away from the curve of RTT. In the case of median filter, the curve of RTO is closer to the curve of RTT. A desirable RTO is one which is close to, and always larger than the RTT. The reason is that, if the RTO is too large, the response to retransmit when a packet is lost will be too slow, and this will decrease the throughput of the transmission path.

VI. CONCLUSION

The estimation of RTO based on median filter algorithm is proposed and analyzed in this paper. Median filter is a useful filter due to its preserving characteristic for image processing applications. From the experimental results, we find that it performs well for heavy-tailed distributions to eliminate impulses in bursty traffic flows. As a result, median filter can perform better than the weighted-average filter because consistent RTT and small RTO are obtained, which are desirable factors for high connection throughput to alleviate traffic congestion.

REFERENCES

- [1] D. E. Comer, *Internetworking with TCP/IP: Principles, Protocols, and Architecture Volume 1*, 5th ed. Pearson Education Inc, New Jersey, 2006, pp. 204 – 210.
- [2] V. Jacobson, "Congestion avoidance and control," *ACM SIGCOMM Computer Communication Review*, vol. 18, no. 4, pp. 314 – 329, Aug 1988.
- [3] J. Astola and P. Kuosmanen, *Fundamental of Nonlinear Filtering*. CRC Press, United States, 1997.
- [4] I. Pitas and A. N. Venetsanopoulos, *Nonlinear Digital Filters: Principles and Applications*. Kluwer Academic Publishers, Boca Raton, 1990.
- [5] J. Postel. RFC793: Transmission Control Protocol. Viewed on 07 May 2008. [Online]. Available: <http://www.rfc-editor.org/rfc/rfc793.txt>
- [6] P. Karn and C. Partridge, "Improving round-trip time estimates in reliable transport protocols," *ACM SIGCOMM Computer Communication Review*, vol. 31, no. 1, pp. 67 – 74, Jan 1995.
- [7] J. F. Kurose and K. W. Ross, *Computer Networking: A Top-Down Approach Featuring the Internet*, 3rd ed. Pearson Education Inc, New Jersey, 2005, pp. 236 – 239.
- [8] L. L. Peterson and B. S. Davie, *Computer Networks: A System Approach*, 4th ed. Morgan Kaufmann Publishers, San Francisco, 2000, pp. 204 – 206.
- [9] L. P. Ma, K. E. Barner, and G. R. Arce, "TCP retransmission timeout algorithm using weighted medians," *IEEE/ACM Transactions on Networking*, vol. 11, no. 6, pp. 569 – 572, Jun 2004.
- [10] V. Paxson and S. Floyd, "Wide area traffic: The failure of poisson modeling," *IEEE/ACM Transactions on Networking*, vol. 11, no. 6, pp. 569 – 572, Jun 2004.
- [11] S. Asmussen, *Applied Probability and Queues*, 2nd ed. Springer, Berlin, 2003.
- [12] Pareto distribution. Answers.com. Viewed on 26 December 2009. [Online]. Available: <http://www.answers.com/topic/pareto-distribution>
- [13] D. H and V. V. Nuclear Power. Viewed on 12 January 2010. [Online]. Available: http://www.coloradocollege.edu/dept/ev/courses/EV212/Block5_2002/fission.htm
- [14] J. E. Freund and R. E. Walpole, *Mathematical Statistics*, 4th ed. Prentice Hall, New Jersey, 1987, p. 218.
- [15] X. J. Hei. Heavy-Tailed Distributions. Viewed on 12 January 2010. [Online]. Available: <http://www.ee.ust.hk/~heixj/publication/comp660f/node4.html>
- [16] W. Winginger, M. S. Taqqu, R. Sherman, and D. Wilson, "Self-similarity through high variability: Statistical analysis of ethernet lan traffic at the source level," *IEEE/ACM Transactions on Networking*, vol. 5, no. 1, pp. 71 – 86, Feb 1997.