

## Research Letter

# PacketTwins: A Novel Method for Capacity Estimation of a Heavy-Loaded Path

Jie Xu

School of Information Science and Engineering, Graduate University of Chinese Academy of Sciences, Beijing 100049, China

Correspondence should be addressed to Jie Xu, xu.jie@mails.gucas.ac.cn

Received 24 March 2009; Accepted 20 June 2009

Recommended by Jun Bi

We propose a method called PacketTwins for estimating the capacity of heavy-loaded paths. Unlike popular packet pair methods, which probe a path with a series of two equal-sized packets, PacketTwins uses a series of twin probe packets that are slightly different in size. By sending twin probe packets alternately, we can obtain new information about selecting valid samples for capacity estimation when a network path is heavy loaded.

Copyright © 2009 Jie Xu. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

## 1. Introduction

Several packet pair methods have been proposed for estimating the bottleneck capacity of an end-to-end path with help of the dispersion of probe packet pair. Moreover, to deal with disruptions caused by cross traffic, various filter algorithms designed to find uncontaminated probe samples have been proposed. For example, in [1], CapProbe was proposed to find valid probe samples with the help of end-to-end delay information. While CapProbe is very accurate and converges quickly when the path is light-loaded, its performance degrades when the path is heavy loaded.

In this letter, we propose a method called PacketTwins for estimating the capacity of heavy loaded paths. Instead of using packet pairs, PacketTwins uses twin packets whose sizes are slightly different to probe a path's capacity. Existing approaches, such as [2–4], use different sized packets to measure the capacity of each intermediate link and other metrics along the path. In contrast, PacketTwins focuses on the estimation of the path capacity (i.e., the bottleneck link capacity) by exploiting the different sizes of the twin probe packets, in addition to the traditional parameters like dispersion and delay. Moreover, PacketTwins preserves the simplicity of CapProbe, and it is less intrusive than packet train-based tools (e.g., Pathrate [5]).

## 2. PacketTwins

**2.1. Feasibility.** Consider the scenario where two packets sent back-to-back cross a path with capacities  $(C_1, C_2, \dots, C_n)$ . The path consists of  $n$  links, and the bottleneck link falls on the  $k$ th link. Mathematically, the path capacity  $C = C_k = \min\{C_1, C_2, \dots, C_n\}$ . Suppose that the two packets leave the  $i$ th link at  $t_i^1$  and  $t_i^2$ , respectively. Packet dispersion is defined as the time interval between the departure times of the two packets. While crossing the bottleneck link, the two probe packets still remain back-to-back; however, the dispersion of the packets changes at the bottleneck link. The new dispersion can be expressed as

$$\Delta_k = \frac{L_2}{C_k}, \quad (1)$$

where  $L_2$  is the size of the second probe packet.

Specifically, the time that each packet leaves the  $i$ th link can be expressed as follows:

$$t_i^j = t_{i-1}^j + \frac{L_j}{C_i}, \quad j = 1, 2. \quad (2)$$

Therefore, the dispersion after the  $i$ th link can be calculated by

$$\Delta_i = t_i^2 - t_i^1 = \Delta_{i-1} + (L_2 - L_1) \frac{1}{C_i}. \quad (3)$$

Recursively, the dispersion measured at the end of the path can be expressed as

$$\Delta = \Delta_k + (L_2 - L_1) \sum_{i=k+1}^n \frac{1}{C_i} = \frac{L_2}{C_k} + (L_2 - L_1) \sum_{i=k+1}^n \frac{1}{C_i}. \quad (4)$$

Equation (4) shows that if the two probe packets are of equal size ( $L_1 = L_2$ ), the packet dispersion after the bottleneck link remains the same as  $L_2/C_k$  until the end of the path. However, if the two packets are different in size, the dispersion will be disrupted by links after the bottleneck link. This may be the reason that most packet pair methods use two equal-sized packets.

In this work, we find that when we transmit two different sized probe packets alternately, the disturbance in the second part of (4) can be eliminated. Specifically, when the sending order of the two packets is changed (i.e., the packet of size  $L_2$  is sent first), the dispersion can be expressed as follows:

$$\Delta' = \Delta'_k + (L_1 - L_2) \sum_{i=k+1}^n \frac{1}{C_i} = \frac{L_1}{C_k} + (L_1 - L_2) \sum_{i=k+1}^n \frac{1}{C_i}. \quad (5)$$

Combining (4) and (5), the path capacity can be calculated by

$$C = C_k = \frac{L_1 + L_2}{\Delta + \Delta'}. \quad (6)$$

**2.2. Size Difference of Twin Packets.** The assumption made in the above deduction is that the packet dispersion calculated in (3) is always larger than the transmission time of the second probe packet. Therefore, to use (3), the following constraint must be satisfied:

$$\Delta_{i-1} + (L_2 - L_1) \frac{1}{C_i} \geq \frac{L_2}{C_i} \implies \Delta_{i-1} \geq \frac{L_1}{C_i}. \quad (7)$$

Similarly, when the sending order of the twin packets is changed, the following constraint must be met:

$$\Delta'_{i-1} + (L_1 - L_2) \frac{1}{C_i} \geq \frac{L_1}{C_i} \implies \Delta'_{i-1} \geq \frac{L_2}{C_i}. \quad (8)$$

Specifically, for  $i = k + 1$ , the inequalities become

$$\frac{L_2}{L_1} \geq \frac{C_k}{C_{k+1}}, \quad \frac{L_1}{L_2} \geq \frac{C_k}{C_{k+1}}. \quad (9)$$

Without loss of generality, we assume that  $L_2$  is larger than  $L_1$ . Since  $C_k < C_{k+1}$  and their differences are normally in the scale of Mbs, the above inequalities can be easily resolved by setting  $L_1$  and  $L_2$  very close to each other. Recursively, there will always be a  $\Delta_k$  term in the dispersion; therefore, for the subsequent downstream links, when the sizes of twin packets are slightly different (e.g., by two bits), the constraints in (7) and (8) will always be satisfied. In addition, bottleneck link capacity estimation based on (6) will yield valid results.

**2.3. Dispersion Variance of Twin Local Modes.** When a path is heavily loaded, probe packets may experience queuing delays after the bottleneck link. Suppose that the delay experienced by two probe packets at the  $i$ th link is  $d_i^1$  and  $d_i^2$ , respectively. When the sending order is changed, the queuing delay of the twin packets becomes  $d_i^{1'}$  and  $d_i^{2'}$ . Thus, the departure time of probe packets from the  $i$ th link in (2) should be modified as follows:

$$t_i^j = t_{i-1}^j + \frac{L_j}{C_i} + d_i^j, \quad j = 1, 2. \quad (10)$$

Consequently, the dispersion in (3) should be revised as

$$\Delta_i = \Delta_{i-1} + (L_2 - L_1) \frac{1}{C_i} + (d_i^2 - d_i^1). \quad (11)$$

Similarly, when the sending order of twin packets is changed, the dispersion becomes

$$\Delta'_i = \Delta'_{i-1} + (L_1 - L_2) \frac{1}{C_i} + (d_i^{2'} - d_i^{1'}). \quad (12)$$

If a number of sample packets experience the same delay after the bottleneck link, they form a local mode [5] in the dispersion distribution. However, as the sending order of twin probes alternates, there are actually two similar local modes even though the samples experience the same delay as shown in (11) and (12). In other words, the different sizes of twin packets and the alternating sending order split each *local mode* into two similar local modes, which we call *twin local modes*.

We observe that the variance of *twin local modes* actually provides an additional clue for filtering valid probe samples. Specifically, the dispersion variance of *twin local modes* after the  $i$ th link is

$$\begin{aligned} \Delta_i^\Delta &= \Delta_i - \Delta'_i \\ &= \Delta_{i-1}^\Delta + \frac{2(L_2 - L_1)}{C_i} - (d_i^1 - d_i^{1'}) - (d_i^{2'} - d_i^2). \end{aligned} \quad (13)$$

Consider the sample probes of *twin local modes* crossing the  $i$ th link. For ease of analysis, we assume that all the samples of the same local mode experience exactly the same processing delays after the bottleneck link. The delays experienced by the first probe packets are all the same ( $d_i^1 = d_i^{1'}$ ), since neither the probe size nor the sending order has any influence. However, the queuing delays of the second probe packets fall into two categories depending on whether they are influenced by the first packets or not.

First, if the dispersion of twin packets is “saturated” by cross traffic, the delay of the second probe packet will be influenced by the cross packets that saturate the dispersion as well as by the first probe packet. In this situation, the variance of packet dispersion of the *twin local modes* after the  $i$ th link can be easily calculated by

$$\Delta_i^\Delta = \frac{y + L_2}{C_i} - \frac{y + L_1}{C_i} = \frac{L_2 - L_1}{C_i}, \quad (14)$$

where  $y$  denotes to the size of packets that saturate the dispersion.

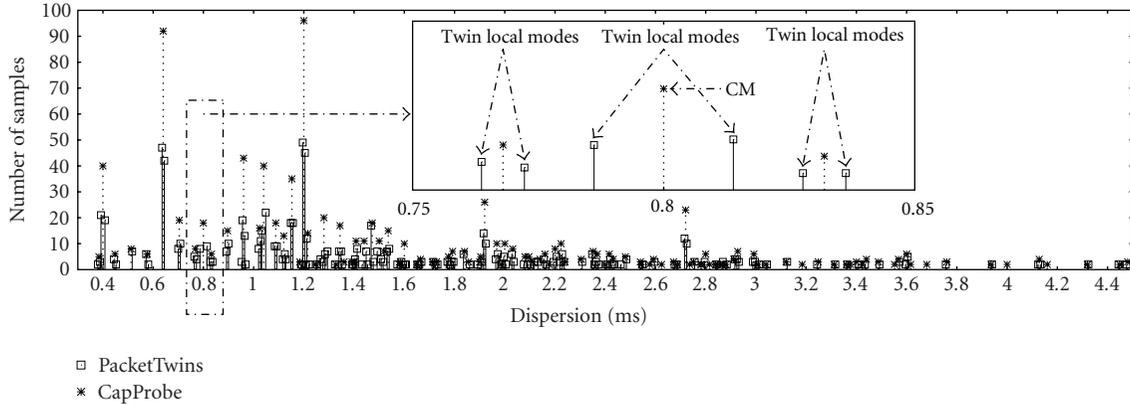


FIGURE 1: Dispersion distributions of CapProbe and PacketTwins. CapProbe uses 200-byte probes, and PacketTwins uses 199- and 201-byte twin probes. All the links on the path are 80% utilized.

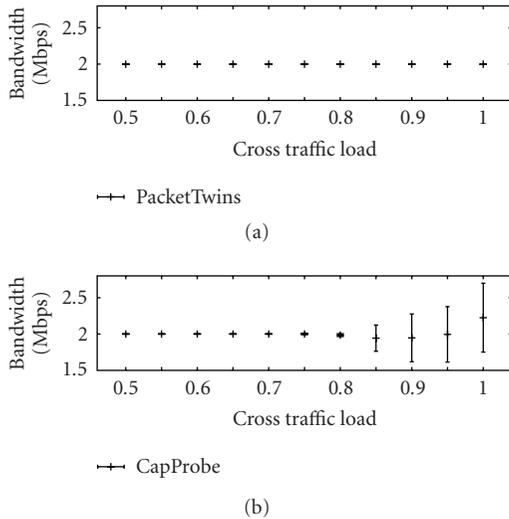


FIGURE 2: Estimation results with PacketTwins and CapProbe.

Second, if the dispersion is not “saturated”, the queueing delay should be same for all the second probe packets ( $d_i^2 = d_i^1$ ) if they belong to *twin local modes*. According to (13), the dispersion variance after the  $i$ th link can be expressed as

$$\Delta_i^\Delta = \Delta_{i-1}^\Delta + \frac{2(L_2 - L_1)}{C_i}. \quad (15)$$

In summary, the dispersion variance after the  $i$ th link can be classified into two categories. Clearly, the dispersion variance in the nonsaturated category is much larger than that in the saturated case. Therefore, the maximal dispersion variance occurs when the dispersion of twin packets is never saturated after the bottleneck link. Specifically, the maximal dispersion variance at the end of the path can be expressed as follows:

$$\max \Delta^\Delta = (L_2 - L_1) \left( \frac{1}{C_k} + 2 \sum_{i=k+1}^n \frac{1}{C_i} \right). \quad (16)$$

We observe that nonsaturated cases hardly ever occur when  $d_i^1 > d_i^2$ . This is because the dispersion is rather tightly compressed, and so it can be easily saturated by heavy cross traffic. In other words, the maximal dispersion variance only occurs for  $d_i^1 \leq d_i^2$ . When  $d_i^1 = d_i^2$ , the dispersion variance reaches the maximum with the smallest dispersion. Consequently, we can conclude that *twin local modes* that achieve the maximal dispersion variance with the smallest dispersion correspond to valid samples when the path is heavily loaded.

**2.4. Algorithm Description.** Based on above analysis, we outline the steps of the PacketTwins algorithm. First we send out twin probe packets alternately. To minimize the impact of variations in cross traffic, we alternate the sending order of the twin packets sample by sample so that *twin local modes* will have similar strength. The second step finds the *twin local modes*. Since the *twin local modes* are very close to each other and have similar occurrence patterns, they can be found easily. If it is difficult to distinguish *twin local modes*, one can send some packet pair probes of size  $(L_1 + L_2)/2$  and use the corresponding local modes for reference. In the third step, valid samples are discovered with help of information about the variance of twin local modes. The path’s capacity can be calculated by (6).

### 3. Simulation Results and Analysis

We simulate PacketTwins in NS-2 [6] on a 6-link path with capacities of {8, 3, 2, 4, 5, 6} Mbps. The nonpersistent cross traffic on each link consists of three Pareto-distributed traffic with the same shape parameter  $\alpha = 1.9$ , different packet sizes (40, 550, 1500 bytes), and corresponding packet weights (0.5, 0.25, 0.25). In each measurement, both CapProbe and PacketTwins send 2000 probe samples. We use a large number of probes because valid samples are rather rare when a path is heavily loaded. For CapProbe, we use 200-byte probes, and for PacketTwins, we use 199- and 201-byte twin probes.

Figure 1 shows the dispersion distributions when all links on the path are 80% utilized. We observe that the dispersion distributions are very complicated, as discussed in [5], but existing packet pair methods do not provide a simple indicator to find the right capacity mode (CM) [5] with packet pair methods.

However, PacketTwins enables us to obtain more information. First, the *twin local modes under PacketTwins* correspond to the *local modes under CapProbe*. Moreover, the *twin local modes* sit on the two sides of the corresponding *local mode of the packet pair* and are equidistant to it. Therefore, if *twin local modes* are hard to distinguish with PacketTwins, we can seek extra information from the dispersion distribution of packet pairs. Second, the dispersion variance of *twin local modes* achieves the maximum value with valid samples. Specifically, the dispersion variance of *twin capacity local modes* around CM (0.8 millisecond) is 0.02773 millisecond which complies with our calculations by (16). The dispersion variances of other *twin local modes* also comply with our analysis; however, they never achieve the maximal value in this scenario.

Figure 2 shows the estimation results derived by PacketTwins and CapProbe under different traffic loads. In each case, 30 independent simulations are performed, and the estimation results have a 95% confidence interval. Clearly, PacketTwins performs much better than CapProbe when the path is heavily loaded. In fact, as long as there are samples for the *twin capacity local modes*, accurate estimation can be achieved by PacketTwins. However, CapProbe may not yield accurate estimations, as valid samples may experience prolonged delays before crossing the bottleneck link.

We also simulate PacketTwins with different probe sizes. Since the path is already heavily loaded, it is observed that large probes perform badly because no valid samples are available in the dispersion distribution. However, when we probe the path with small twin packets, even if the capacity mode is very weak, we can still find corresponding valid samples and estimate the capacity accurately.

## 4. Conclusion

In this letter, we present a novel path capacity estimation approach called PacketTwins. By splitting each *local mode under packet pair* methods into *twin local modes under packet twins*, PacketTwins obtains new filter information about the dispersion variance of *twin local modes*. The results of simulations show that PathTwins is superior to CapProbe because it can achieve more accurate capacity estimation, even when a network is heavily loaded.

## Acknowledgments

This work was supported by the National 863 project (Grant no. 2007AA01Z297) and the National Science Foundation of China (Grant no. 60773137). The author would like to gratefully thank Ling-Jyh Chen for his valuable input.

## References

- [1] R. Kapoor, L.-J. Chen, L. Lao, M. Gerla, and M. Y. Sanadidi, "CapProbe: a simple and accurate capacity estimation technique," *Computer Communication Review*, vol. 34, no. 4, pp. 67–78, 2004.
- [2] K. Harfoush, A. Bestavros, and J. Byers, "Measuring bottleneck bandwidth of targeted path segments," in *Proceedings of the 22nd Annual Joint Conference on the IEEE Computer and Communications Societies (INFOCOM '03)*, vol. 3, pp. 2079–2089, San Francisco, Calif, USA, March 2003.
- [3] K. Lai and M. Baker, "Measuring link bandwidths using a deterministic model of packet delay," *Computer Communication Review*, vol. 30, no. 4, pp. 283–294, 2000.
- [4] V. Jacobson, "Pathchar: a tool to infer characteristics of Internet paths," 1997.
- [5] C. Dovrolis, P. Ramanathan, and D. Moore, "Packet-dispersion techniques and a capacity-estimation methodology," *IEEE/ACM Transactions on Networking*, vol. 12, no. 6, pp. 963–977, 2004.
- [6] "The Network Simulator—ns-2," <http://www.isi.edu/nsnam/ns>.



# Hindawi

Submit your manuscripts at  
<http://www.hindawi.com>

