韓國 컴퓨터情報學會 論文誌
第11卷 第5號, 2006. 11.

2006-11-5-1-1

ASH를 이용한 Pathrate에서의 Local Mode 검출 알고리즘

황 월*, 김 용 수*

A New Algorithm Based on ASH in Local Modes Detection of Pathrate

Yue Huang*, Yong-Soo Kim*

요 약

효율적인 네트워크 운용을 위해 트레픽를 측정하는 일은 중요하다. 흔히 용량(capacity)은 트레픽 부하가 없을 때 경로가 제공할 수 있는 최대처리량 또는 경로상의 모든 링크 간의 최소 전송율로서 정의된다. Pathrate는 현재 가장 널리 사용되는 네트워크 용량 측정 도구 중의 하나로써 네트워크의 일시적인 부하에 관계없이 정확한 측정을 할 수 있고 수년간의 개발과 보완으로 성능도 안정되어 있다. Pathrate에서의 Local Mode 검출에는 통계적 방법이 사용되는데 본 논문에서는 ASH(Averaged Shifted Histogram)을 이용한 Local Mode 검출 알고리즘을 제시하고, 구현을 통해 기존의 방법보다 더 나은 결과를 얻었음을 보였다.

Abstract

Network measurement is a vital part of network traffic engineering. In a network, the metric "capacity" characterizes the maximum throughput the path can provide when there is no traffic load, or the minimum transmission rate among all links in a path. Pathrate is one of the most widely used network capacity measurement tools nowadays. It's famous for its accurate estimation result and non restriction of the temporal network traffic condition. After several years of development, its performance becomes more stable and reliable. Extant local modes detection algorithm in pathrate is based on statistic methodology histogram. This paper suggests a new algorithm for local modes detection based on ASH (Averaged Shifted Histogram). We have implemented this algorithm and will prove it can accomplish the same task as the original one with a better result.

▸ Keyword : pathrate, local mode 검출, 히스토그램, ASH(Averaged Shifted Histogram)

• 제1저자 : 황 월, 교신저자 : 김용수
• 접수일 : 2006.09.12, 심사일 : 2006.10.16, 심사완료일 : 2006.11.18
* 경원대학교 소프트웨어대학

Ⅰ. Introduction

As one of the greatest inventions by human beings, Internet has been changing our society with its magic power since it came out and makes our works and lives more efficient and convenient. Its contribution to human society blots out the rays from any other inventions in the human history. However, its dramatic developing and expanding speed goes on at the cost of increasing complicacy. People begin to pay more attention to the Internet's performance capability while they are enjoying the services provided by it, such as the network traffic measurement, model establishment, traffic control, and so on. Network capability estimation provides us a way to understand the network behavior. The purpose is to obtain relevant data of the network behavior and try to find out the main factors that affect the network capability. Once we get them, we can optimize transport performance, overlay network routing, and peer to peer file distribution [1].

Perhaps the most common complaint from Internet users is that the response time takes a long time. When confronting this complaint, we should find out what indeed lead to the decrease of network capability, because many factors can result in this unpleasant performance of network capacity, such as the speed of reading data on sender host, the speed of writing data on receiver host, the bottleneck speed of network path from sender to receiver, and the utilization of the path, etc. [2]. Sometimes, the hosts may have capable performance, and the networks that the two hosts are connected to may also have sufficient capability, while due to the limit capacity of the link between sender and receiver, the whole performance is poor. A sensible check is required to see where the bottleneck is. Pathrate and pathload can be used to estimate the capacity and the available bandwidth of the network respectively, which are respectively introduced in [3] and [4].

Ⅱ. Pathrate's Working Discipline

Pathrate was released in April 2001. It is a free software developed and maintained by Professor Constantinos Dovrolis of Georgia Institute of Technology. This is an end to end Internet paths capacity estimation tool, using UDP packets for probing and exchanging the control information through TCP connection. The estimation result can be obtained by using the technology of packet pair, packet train, and statistical method histogram. During the measurement period, several different statistic estimations are taken. They are distributed into three main phases.

At the very beginning, pathrate calculates several kernel data for the whole estimation process, such as the minimum train spacing, the maximum packet size, and the maximum train length, etc. In the first phase, the primary function is to find out the value of parameter bin width. In histogram, the same data with different bin widths can result different visual diagrams, so bin width is a crucial parameter for the local modes detection algorithm. Pathrate uses packet trains of increasing length with minimum spacing interval to do capacity measurements and calculates bin width based on the difference of the maximum and minimum measurement result in this phase.

The second phase uses a large number of packet pairs with different sizes to uncover all the local modes. Local modes are the local maxima in the bandwidth distribution. One of them corresponding to the capacity of the path is the capacity mode. Pathrate's purpose is to find it out. Local modes detection algorithm is used here for the first time.

Phase three uses a large number of packet trains with maximum train length to estimate Asymptotic Dispersion Rate (ADR). The local modes corresponding to the capacity mode and Post Narrow Capacity Modes become weaker for packet train dispersion as the number of packets per train increases. When the number becomes sufficiently large, bandwidth distribution with packet train becomes unimodal. This unitary mode is called ADR, which is related to the utilization of all links in the path. ADR is a useful metric for monitoring the quality of service that the path offers [5]. We also use the same local modes detection algorithm to detect it. It is less than the capacity, so we can use it as a lower bound for the capacity of the path. We ignore the local modes formed in phase one, whose values are smaller than it. If there is more than one mode whose values are larger than ADR, the one with largest merit will be chosen as the capacity mode. Merit is calculated by using following formula:

$$\text{merit} = \frac{\sum (x[i]-\bar{x})^4}{\sigma^2} \times \frac{\text{mode count}}{\text{total count}} \quad \cdots\cdots\cdots (1),$$

where x[i] is one of the bandwidth estimations in current mode bell(mode bell is the range between two bins with local minimum frequency);

$\bar{x}$ is the arithmetic mean of these estimations;

σ^2 is the population variance of them; mode count is the number of measurements in current local mode; total count is the total number of measurements in the sample. Because all these works are done by statistical methodology, so we can't get an accurate numerical value for the capacity and that's why the final result is printed in form of range.

III. Histogram

3.1 A brief introduction of histogram

A histogram is a great way to get a visual image of data, which gives a lot of information about where the data are clumped, how spread out the numbers are etc. It is the simplest density estimator and is one example of a frequency curve, using tabulation of data in bins. Given a sample x1,x2,……xn contained in interval(a, b), the histogram is constructed over a partition {ti} of (a, b) into M intervals, Bi=$[t_{i-1}, t_i)$, such that a=t0<t1<……<tM=b. Bi is a bin, which is a numerical range we are going to group the data into. The bin width of Bi is denoted by hi= ti - ti 1, and bin count is denoted by vi, so that $\sum_{i=1}^{M} v_i = n$.

Choosing an appropriate bin width is very important in histogram. It's a basic issue with a long history in density estimation. Although this problem is still unresolved, there have been some guidelines and we will adopt one in the following discussion. If the bin width is large, we can get few bins and the density diagram looks rough, which means little distribution information can be gotten from this histogram. On the other hand, if bin width is too small, a large number of bins make the diagram noisy and time consumption to get the distribution. In [6], Professor Stocc brought forward an formula to get an approximate optimal bin width,

$$h = \frac{3.49 \times S}{n^{\frac{1}{3}}} \quad \cdots\cdots\cdots (2).$$

where n is the size of the sample and S is the sample standard deviation of the exploratory data,

$$S = \sqrt{\frac{\sum x^2 - \frac{\left(\sum x\right)^2}{n}}{n-1}} \quad \cdots\cdots\cdots (3).$$

In [7], Freedman and Diaconis suggested a formula as following:

$$h = \frac{2 \times IQR}{n^{\frac{1}{3}}} \quad \cdots\cdots\cdots (4).$$

IQR here indicates the inter quartile range of the sample data, which can be gotten in following steps: rearrange the given sample in increasing order $x_1', x_2' \ldots\ldots x_n'$; assume that:

$$\frac{n-1}{4} = i \cdots j \quad \text{......} (5).$$

$$IQR_1 = x_{i+1}' \times \frac{4-j}{4} + x_{i+2}' \times \frac{j}{4} \quad \text{......} (6).$$

$$IQR_3 = x_{n-i-1}' \times \frac{4-j}{4} + x_{n-i-2}' \times \frac{j}{4} \quad \text{......} (7).$$

$$IQR = IQR_3 - IQR_1 \quad \text{......} (8).$$

And Wand provides a modification to satisfy an asymptotic optimality condition in [8]. The formula is given by:

$$h = \frac{3.49 \times \hat{\delta}}{n^{\frac{1}{3}}} \quad \text{......} (9).$$

where $\hat{\delta}$ =min(S, $\frac{IQR}{1.349}$). This expression for h is based on normal scale bin width selection. The normalizing constant (c=1.349) is used to guarantee that IQR/c is an asymptotically unbiased estimator of δ whenever the underlying data are normally distributed, i.e,

$$E\left(\frac{IQR}{1.349}\right) = \delta \quad \text{......} (10).$$

The practical importance of this method is that these data determine the width of the bin using traditional measures of scatter and the sample size, not based on heuristic experience anymore.

3.2 An improvement of histogram - ASH

In histogram, bin origin is another design parameter which can cause a great impact in vision. That means besides bin width, different choice of bin location t0 can also give out different histograms. Averaged shifted histogram (ASH) is brought forward to reduce the impact of the uncertainty of choosing an origin, smooth the histogram and reduce the bias. ASH calculates m histograms with the same bin width h, shifts them by $\frac{jh}{m}$ to the right and takes an average to reduce the bias. Therefore the original bin $B_i = [t_0 + (i-1)h, \ \ t_0 + ih)$ is changed into $B_{ij} = \left[t_0 + (i-1+\frac{j}{m})h, \ \ t_0 + (i+\frac{j}{m})h\right)$, $j \in \{1, \ldots, m-1\}$, and $B_i = B_{i0}$. We have to notice that ASH is not simply an ordinary histogram with a smaller bin width. In principle, we take two steps to get an ASH: form several histograms with equal bin width but different bin locations, and average these shifted histograms [9] by the formula:

$$\hat{f}_{ASH}(x) = \frac{1}{m}\sum_{i=0}^{m-1}\hat{f}_{ih}(x) \quad \text{......} (11).$$

Although ASH becomes much smoother than ordinary histograms, it still depends on the selection of bin width.

Ⅳ. Comparison Between Original And Proposed Algorithms

4.1 Basic idea and flaws in original algorithm

Histogram is applied to the original local modes detection algorithm. It is iterative, because one execution of the algorithm can only get one mode value. Its main steps are shown in Figure 1.

```
Step 1: check whether all the data are marked; if they are, the algorithm is finished;
Step 2: from the smallest unmarked value, partition the remainder data into several
bins with width of bin_wd; find the bin with the largest frequency, i.e. local mode;
Step 3: current bin points to the largest frequency bin; search the left part of this bin:
if   the left bin has less measurements than the current one's
     it belongs to the current mode bell, current bin = the left bin;
else
     it belongs to another new mode bell, break;
Step 4: search the right part of the largest frequency bin in the same way;
Step 5: mark all the data unavailable in this mode bell; return local mode's value.
```

그림 1 : 원래의 local mode 검출 알고리즘
Figure 1: Original local modes detection algorithm

As we have seen, the idea is simple. When looking into the codes of pathrate, we could find there are two hidden flaws existing in it. The first one is about the ascertainment of local modes. In the algorithm, the process of searching local modes is simply based on histogram. As we mentioned above, different histograms can be generated if we choose different origins, even the shift is very small. So data with bias existing in certain bins may have a great effect on the local modes detection. That's why ASH is chosen instead of histogram to detect the local modes. Figure 2 illustrates how the local modes change with a slight shift of origin. We use frequency polygons instead of histogram in order to get a better visual observation.

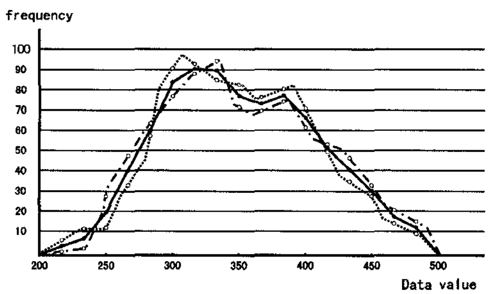


그림 2 : 원점 이동으로 인한 히스토그램의 변화
Figure 2: Same data generate different histograms with shifted origins

The second flaw is about the boundary bin. In the programming codes, averaged shifted principle has been used for the detection of mode bell's boundaries, but the shift is based on data in the exploratory sample rather than on bins. On one side, this kind of calculation is a little nuisance; on the other side, the shift only works on one mode bell but not on the overall distribution. That indicates the mode bell who is detected earlier than the others has the priority to occupy the boundary bins which should be shared with its neighbor mode bells. The unfairness is especially obvious when its neighbor mode bell has nearly the same size as its. This effect can be found in merit calculation which we mentioned in formula (1). Figure 3 can illustrate this problem. According to this algorithm, there are 3 mode bells: mode bell 1, (x3, x5); mode bell 2, (x2, x3); mode bell 3, (x1, x2). The range (x3, x4) belongs to mode bell 1. Is it fair to mode bell 2?

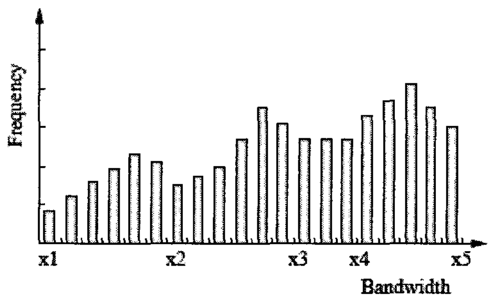


그림 3 : 영역 (x3, x4)은 mode bell 1에 속한다. mode bell 2에 대해 바르게 처리되고 있는가?
Figure 3: The range (x3, x4) belongs to mode bell 1. Is it fair to mode bell 2?

4.2 Implementation of the new algorithm

We propose a new algorithm based on ASH for local modes detection. We take all the measurement results in consideration and choose the bin width based formula (9). Although there are two complex parameters in comparison in the formula, the pathrate programming codes have already rearranged the measurements data in increasing order and calculated the standard deviation of these data before calling the local modes detection algorithm, so what we have to do is just quoting the parameters we need rather than doing the calculation over again. It's suggested parameter m should be no more than 10; if larger than 10, the process of calculation becomes complexly, but it makes little sense on smoothing of the histogram. Here m is chosen as

5. Because in the reality, the measurement results for our estimation didn't cover a large interval and the sample size is very large, 5 is enough. After the execution of this algorithm, all the existing mode bells are returned in order of their bandwidth. We release the basic idea of the new algorithm as follows:

Step 1: distribute two struct arrays **bins** and **modes** to record the frequency and boundaries of each bin and mode bell; at first, choose 0 as the histogram origin;
Step 2: partition the increasing ordered bandwidth estimation results into several bins with width of **bin_wd**;
Step 3: if bins partition has been done for 5 times
    go to **step 4**;
  else
    shift the bin origin to the right slightly, go to **step 2**;
Step 4: get the averaged histogram of these 5 histograms; if two adjoining averaged bins have the same frequency, they are combined into one bin;
Step 5: do comparison on all the bins' frequencies to find out the local minimum and maximum value. Bandwidth estimation results corresponding to the local minimum values are the mode bell's boundaries and the maximum value is the number of measurements in local mode.

그림 4 : 새로운 local mode 검출 알고리즘
Figure 4: New local modes detection algorithm

In the new algorithm, an averaged histogram is generated based on 5 initial histograms with the same measurement data and bin width but shifted origins. Histogram curve becomes smoother and more apparent with this method, so the first problem is mended. When averaged histogram is created, the adjoined averaged bins that have the same frequency are combined into one. When the mode bells are created, the boundary bin of two adjoined mode bells is shared by each other instead of belonging to the "stronger" one. These two disciplines make the local modes detection fair to every mode bell, so the second problem is improved. How the proposed algorithm works will be shown in the next section.

4.3 The application of the new algorithm and the testing

To test the new algorithm, we enlarge the sample's interval against the reality. The sample data are randomly generated varying from 1 to 100. In the first place, let's have a look at how the bin width affects the local modes' distribution. The following Figure 5 to Figure 8 show different histograms with different bin widths. h' is the result of formula (4) and h is the result of formula (9). We didn't take formula (2) into account, because formula (2) and formula (9) have the same result in this execution.

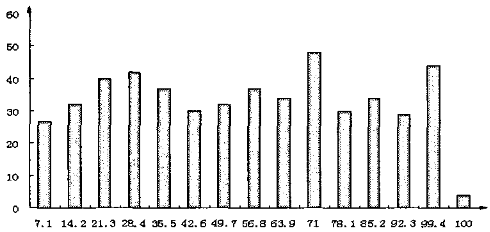


그림 5: bw = h' 시의 히스토그램
Figure 5: Histogram with bw = h'

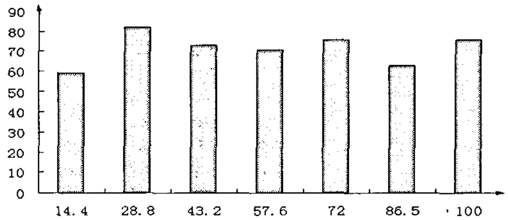


그림 6: bw = h/2 시의 히스토그램
Figure 6: Histogram with bw = h/2

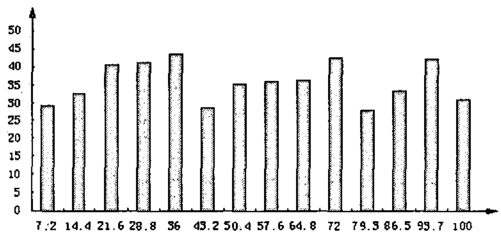


그림 7: bw = h 시의 히스토그램
Figure 7: Histogram with bw = h

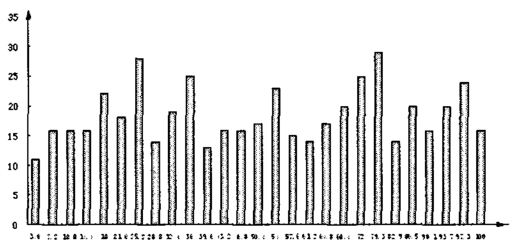


그림 8: bw = 2h 시의 히스토그램
Figure 8: Histogram with bw = 2h

As what the figures show, different bin widths lead to different mode bell distributions. Bin width in Figure 6 is so large that the histogram curve is very rough. We hardly tell where the local mode is. Bin width in Figure 8 is very small, but its distribution is nearly the same as Figure 5 and Figure 6's. More complex calculation makes little sense. Figure 5 has one more bin than Figure 7, but in the mass their distributions are analogical to each other. So bin width gotten from formula (9) is neither too small nor too large. Using it can get a clear data distribution curve with appropriate calculation complexity.

From Figure 9 to Figure14, we can see the impact of choosing different histogram origins and how the averaged histogram is generated.

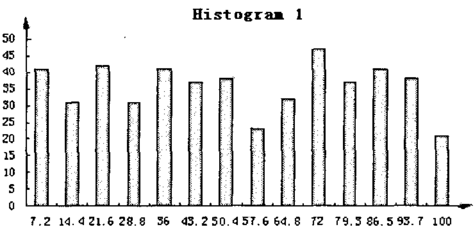


그림 9: 첫 번째 검출하는 히스토그램
Figure 9: Histogram for the first detection

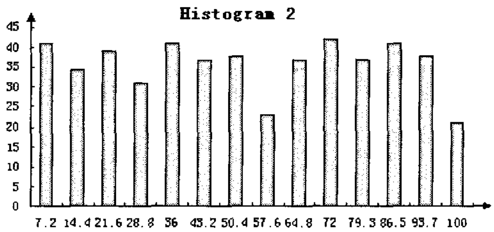


그림 10: 두 번째 검출하는 히스토그램
Figure 10: Histogram for the second detection

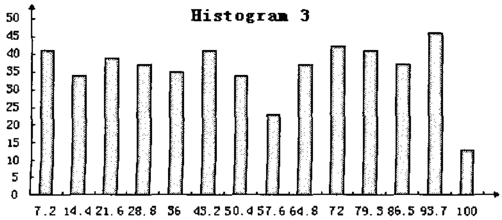


그림 11: 세 번째 검출하는 히스토그램
Figure 11: Histogram for the third detection

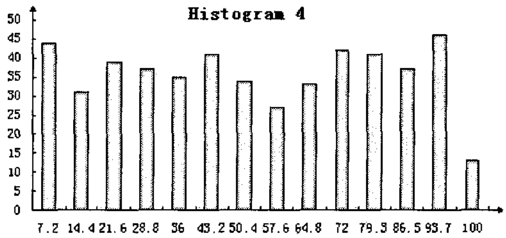


그림 12: 네 번째 검출하는 히스토그램
Figure 12: Histogram for the forth detection

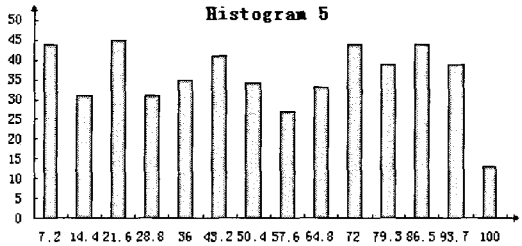


그림 13: 다섯 번째 검출하는 히스토그램
Figure 13: Histogram for the fifth detection

Above six figures illustrate an execution result of the proposed algorithm. Bin width applied to this execution is 7.3501, and the shift parameter is 0.2. From these figures, we could find that the three local modes and mode bells become more obvious in the averaged shifted histogram. So the proposed algorithm can work well.

V. Conclusion

This paper introduces a new local modes detection algorithm based on ASH, which is applied to pathrate. There are two vital parameters that can influence the distribution of the data in histogram. One is bin width and the other is origin. We firstly show the impact of choosing an appropriate bin width, then focus on origin. The new algorithm generates several histograms with shifted origins and gets an average of them. In this way, local modes could become more obvious and precise. New algorithm also pays attention to the mode bell boundary ascription. It makes the boundary bin of two contiguous mode bells to be shared by each other to solve the problem of boundary bin preemption. The resulting figures prove that the new algorithm can accomplish the same task as the original one with a more exact result.

References

[1] R. S. Prasad, M. Murray, C. Dovrolis and K. Claffy "Bandwidth Estimation: Metrics, Measurement, Techniques, and Tools ". In IEEE Network, 17(6), November 2003.

[2] Richard Blum "Network Performance Open Source Toolkit". Wiley Publishing, Inc., Indianapolis, Indiana 2003.

[3] C. Dovrolis, P. Ramanathan, and D. Moore "Packet dispersion techniques and a capacity estimation methodology". In IEEE/ACM Transactions on Networking (TON) Volume 12, Issue 6, December 2004.

[4] M. Jain, C. Dovrolis "Pathload: A Measurement Tool for End to End Available Bandwidth". In ACM SIGCOMM, Pittsburg, PA, 2002.

[5] C. Dovrolis, P. Ramanathan, and D. Moore "What do Packet Dispersion Techniques Measure?". In Proceedings of IEEE INFOCOM, pages 905 914, April 2001.

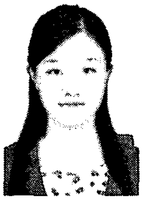
[6] D.W. Scott "On Optimal and Data Based Histograms". In Biometrika, 66, 605 610, 1979.

[7] D. Freedman, and P. Diaconis, "On the histogram as a density estimator: L2 theory". Zeitschrift für Wahrscheinlichkeitstheorie und verwandt Gebiete, 57, 453 476, in 1981.

[8] M.P. Wand "Data Based Choice of Histogram Bin Width". In the American Statistician, 51, 59 64, 1992.

[9] D.W. Scott "Averaged Shifted Histograms: Effective Nonparametric Density Estimators in Several Dimensions". In Annals of Statistics, Vol. 13, No. 3 (Sep., 1985), pp. 1024 1040.

저 자 소 개



Yue Huang

Software College, Kyungwon University

〈Research interest〉 Free software, embedded system.

Yong Soo Kim

Software College, Kyungwon University

〈Research interest〉 OS, free software, embedded system, Performance Management and Capacity Planning.