

# 時戳精度對網路量測準確度之影響探討

黎明富、呂紹宇

長庚大學電機工程學系

mfli@mail.cgu.edu.tw, m0121024@stmail.cgu.edu.tw

## 摘要

本文探討時戳精度(Precision)對網路量測準確度的影響，我們以網路可用頻寬估測的技術為例，來說明時戳精度對網路可用頻寬量測的影響。目前 ICMP 協定所定義的時戳精度只有到毫秒(ms)，在高速網路環境下，封包的傳送時間可能只有數十微秒( $\mu$ s)，甚至更低。因此，採用毫秒精度的時戳來進行網路量測，自然就會產生較大的量測誤差。本文將透過模擬來比較不同時戳精度下之網路可用頻寬估測之準確度，並提出時戳規格之修改方式，以做為下一代網路時戳標準之參考。

**關鍵詞：**網路量測、時戳精度、可用頻寬估測、ICMP 協定、下世代網路。

## Abstract

The impact of timestamp precision on network measurements is studied in this work. The network available bandwidth estimation technique is taken as an example to investigate the measurement accuracies under different timestamp precisions. Currently, the ICMP timestamp precision is limited to a millisecond. However, in high speed networks, the packet transmission time may be in only a few microseconds or even less. Therefore, network measurements based on millisecond-precision timestamps may result in a larger estimation error. In this paper, the estimation accuracies of available bandwidth under various timestamp precisions are compared using simulations. Moreover, a proposal to revise the ICMP timestamp specification is presented for measurement applications in next generation networks.

**Keywords:** Network Measurement, Timestamp Precision, Available Bandwidth Estimation, ICMP Protocol, Next Generation Network.

## 1. 前言

隨著科技的日新月異，網路漸漸融入人們的生活中，從以往只能坐在固定的地方使用電腦上網，到現在無論身處何地都能透過網路連到世界的任何一個角落，要如何有效地使用有限的網路頻寬就成為一個重要議題。尤其網路頻寬往往不是這麼的穩定，所以準確的頻寬估測就變得相當重要。以網路可用頻寬(Available Bandwidth)估測方法為例，目前的估測方法大部份都採取主動探測法，即利用探測封包來得知網路可用頻寬。在文獻[1]中，作者採用 ICMP (Internet Control Message Protocol) TIMESTAMP REQUEST (Type 13)封包來探測整段網路的瓶頸鏈路，由於在接收端收到 TIMESTAMP 封包後會回覆 TIMESTAMP REPLY (Type 14)封包，藉由封包上記錄的時戳經由演算法進而算出可用頻寬。近年來由於網路電視 IPTV(Internet Protocol Television)[2]的快速發展與普及，但是每個使用者的網路環境又是如此參差不齊，所以伺服器就必須針對不同的使用者給予不同的畫面品質，於是有了可調式視訊編碼(Scalable Video Coding, SVC)[3]技術的產生。然而為了確保伺服器能夠針對不同使用者的網路環境送出最適當的畫面，如何快速找出最準確的有效頻寬就是一項很重要的技術。

頻寬估測主要分為被動式及主動式兩大類，被動式主要是利用延遲(Delay)或是封包遺失率(Loss Rate)等等相關參數推估目前的可用頻寬，而主動式是實際發出探測封包，經由通過各節點所記錄的各種參數預測可用頻寬，由於主動式較被動式精確且快速，目前做法大都採取主動式如 Delphi[4]、IGI[5]、Spruce[6]、MoSeab[7]、Pathrate[8]、TOPP(Train of Packet Pair)[9]、SLoPS[10, 11]、PathChirp[12]、PathVar[13]、PathMon[14]，其中[4][5][6][7][8]屬於間隙模型(Gap Model)，利用探測封包間的間隔變化來推估容量及可用頻寬。文獻[9][10][11][12][13][14]等所採用的方法則屬於速率模型(Rate Model)，利用改變探測封包的速率來推估出可用頻寬。本文將採用文獻[1]之可

Type	Code	ICMP Header Checksum
Identifier		Sequence Number
Originate Timestamp		
Receive Timestamp		
Transmit Timestamp		

圖 1、可記錄時戳之 ICMP 封包格式。

用頻寬估測方法來探討時戳精度對網路量測準確度之影響。

以目前 ICMP 協定標準來說，封包上記錄之時戳精度只有到毫秒(ms)，圖 1 即為一個具有可記錄時戳欄位之 ICMP Type 13/14 封包的格式，其中 Original Timestamp 為傳送端送出此封包之當地時間、Receive Timestamp 為接收端收到此封包之當地時間、而 Transmit Timestamp 為接收端回傳封包至傳送端之當地時間。雖然在低速網路下(如頻寬 10Mbps)，封包經過中繼節點時需耗時較長(通常超過一毫秒)，以現行 ICMP 封包之時戳精度來說，並不會有太大問題，但在高速網路下(如頻寬 100Mbps)，封包經過中繼節點時之重新傳送時間往往只需數十微秒( $\mu s$ )，這時如果使用原本 ICMP TIMESTMAP 毫秒精度之時戳標準，則會因為所記錄之時戳精度不足而造成估測上的誤差。目前很多網路設備的時間精度已可達到微秒等級[15][16]，但若相關的網路協定時戳精度沒有進行修改，依然無法解決高速網路環境下量測誤差的情況，這就是本論文所要探討與解決的問題。

## 2. 時戳精度對可用頻寬估測準確度之影響

本文採用 NS2 網路模擬軟體來進行相關實驗，由客戶端(Client)發送探測封包經過所有節點，透過探測封包經過節點上所記錄之時戳，經由文獻 [1] 所提出之 GNAPP(Gaps of Non-Adjacent Probing Packets)即可估測出其瓶頸鏈路容量(Capacity)  $C$  之大小、背景資料流(Cross Traffic)流量  $\lambda$  之大小、及可用頻寬(Available Bandwidth)。圖 2 為本實驗之網路架構圖，而在背景資料流方面，我們分別模擬了 CBR (Constant Bit Rate)、Pareto ON-OFF、及 Poisson 程序三種不同型態的背景資料流，針對瓶頸鏈路容量、背景資料流、及可用頻寬進行估測，並跟實際值比較，以計算估測誤差。

NS2 模擬的探測封包是可記錄時戳精度至微秒的封包，至於毫秒精度之時戳則是將原始

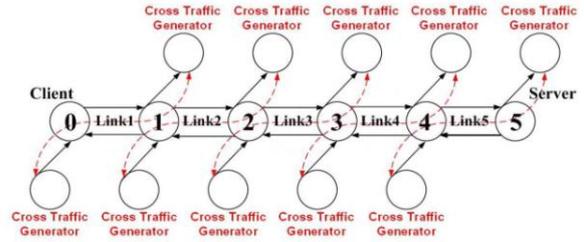


圖 2、NS2 模擬之網路環境架構圖。

微秒精度之時戳以四捨五入的方式得出毫秒精度之時戳模擬數據，然後對此兩組數據分別利用 GNAPP 演算法[1]去估測，並進行比較。

在此節中，我們將考慮低速及高速兩種網路環境，並對圖 2 的 Downlink (從 Server 到 Client)路徑之可用頻寬進行估測，低速的網路環境 Link 1、Link 2、Link 4、Link 5 之頻寬均設定為 20 Mbps，而在高速網路環境中則設定為 100 Mbps。Link 3 為瓶頸鏈路，其頻寬在低速與高速的網路環境中分別設定為 15 Mbps 與 75 Mbps，而 Link 3 之背景資料流在低速與高速的網路環境中則分別設定為 10 Mbps 與 50 Mbps。探測封包之大小  $L$  均設定為 1500 Bytes。探測週期  $T$  設定為 5 秒，每一探測週期會進行 5 回合的探測，每一回合會送出  $N = 60$  個探測封包。由於文獻 [1] 所提出的是雙向探測機制，我們將 Uplink 及 Downlink 設定為兩個獨立的通道，Uplink 當中的所有鏈路(包含 Link 3)參數  $C$  及  $\lambda$  均設定為和 Downlink 的 Link 1、Link 2、Link 4、Link 5 一樣，這樣能確保整段通道的瓶頸鏈路在 Downlink 的 Link 3 上。本實驗我們主要是針對 Downlink 進行探測，當然這也能同樣應用在探測 Uplink 上，相關模擬參數詳列於表 1 及表 2。

首先探討在低速網路環境下，時戳精度對可用頻寬估測準確度之影響。圖 3 至圖 5 分別為 CBR、Pareto ON-OFF、Poisson 背景資料流環境下可用頻寬之估測結果，其中  $M$  為移動平均(Moving Average) 參數、 $W$  則為所計算之非相

表 1、低速網路環境下相關模擬參數設定。

$N$	60
$L$	1500 Bytes
$M$	30
$W$	25
Link 1: $C/\lambda$	20 Mbps/1 Mbps
Link 2: $C/\lambda$	20 Mbps/1 Mbps
Link 3: $C/\lambda$	15 Mbps/10 Mbps
Link 4: $C/\lambda$	20 Mbps/1 Mbps
Link 5: $C/\lambda$	20 Mbps/1 Mbps

表 2、高速網路環境下相關模擬參數設定。

$N$	60
$L$	1500 Bytes
$M$	30
$W$	25
Link 1: $C/\lambda$	100 Mbps/1 Mbps
Link 2: $C/\lambda$	100 Mbps/1 Mbps
Link 3: $C/\lambda$	75 Mbps/50 Mbps
Link 4: $C/\lambda$	100 Mbps/1 Mbps
Link 5: $C/\lambda$	100 Mbps/1 Mbps

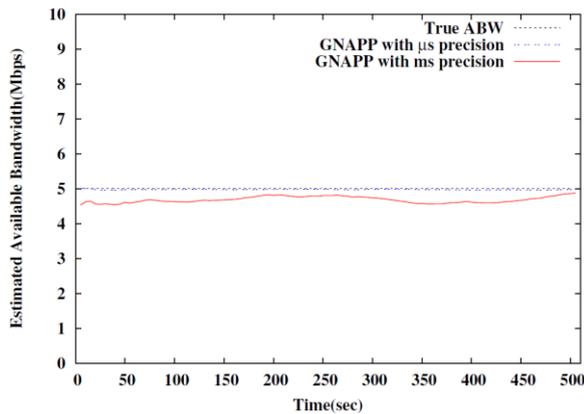


圖 3、低速網路環境中 CBR 背景資料流下之可用頻寬估測結果( $M=30, W=25$ )。

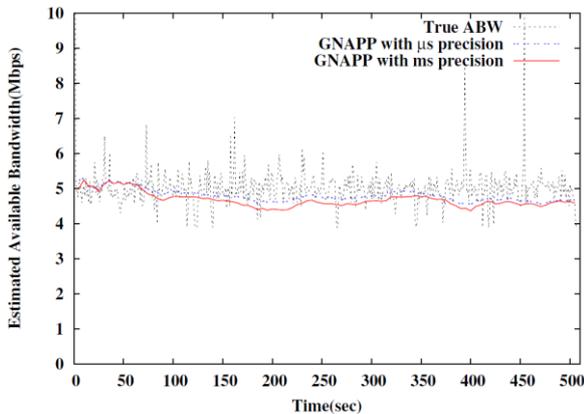


圖 4、低速網路環境中 Pareto ON-OFF 背景資料流下之可用頻寬估測結果( $M=30, W=25$ )。

鄰探測封包間的最大間隔數(以封包數計) [1]。由於 CBR 為最穩定的背景資料流，所以可以得到最準確的估測結果。

由模擬結果可知在 CBR 背景資料流下，微秒精度的時戳所得出的估測值幾乎完全與實際值吻合，而毫秒精度的時戳所得出的估測值則與實際值有些許誤差。Pareto ON-OFF 和 Poisson 背景資料流下的可用頻寬估測值就不如 CBR 來的準確，這是因為 Pareto ON-OFF 及 Poisson 背景資料流的叢集度(Burstiness) 及

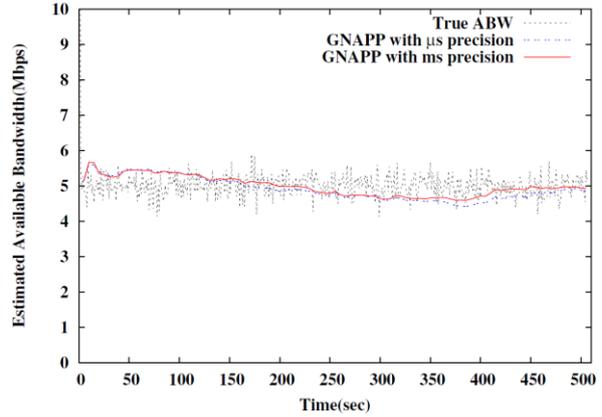


圖 5、低速網路環境中 Poisson 背景資料流下之可用頻寬估測結果( $M=30, W=25$ )。

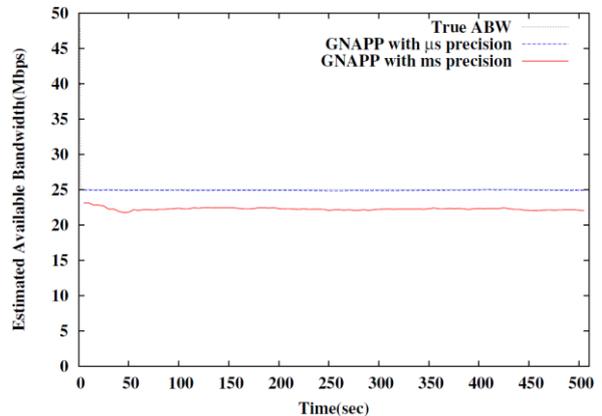


圖 6、高速網路環境中 CBR 背景資料流下之可用頻寬估測結果( $M=30, W=25$ )。

隨機性(Randomness)遠遠較 CBR 背景資料流為高，故所得到的估測誤差也較高。

接下來探討在高速網路環境下，時戳精度對可用頻寬估測確度之影響，模擬結果如圖 6 至 8 所示。圖 6 為在 CBR 背景資料流環境下之估測結果，由圖 6 可看出使用時戳精度為微秒的探測封包依然會得到幾近完美的估測值，但使用時戳精度為毫秒的探測封包所得到的估測值則與實際值有相當大的誤差。這是因為在高速網路環境下，探測封包經過各個中繼節點時只需耗費數十微秒的延遲，若使用毫秒精度的時戳，將無法準確記錄探測封包經過時的時間，因此產生出不小的誤差。在瓶頸鏈路容量為 75 Mbps 的鏈路下就有如此差距，將來在下一代更高速的網路環境下，其誤差將會更大。

圖 7 則為 Pareto ON-OFF 背景資料流環境下之估測結果，圖 8 則為 Poisson 背景資料流環境下之估測結果。從圖 7 及圖 8 可發現 Pareto ON-OFF 及 Poisson 背景資料流下的瓶頸鏈路估測值就不如 CBR 來的準確，其原因如同上

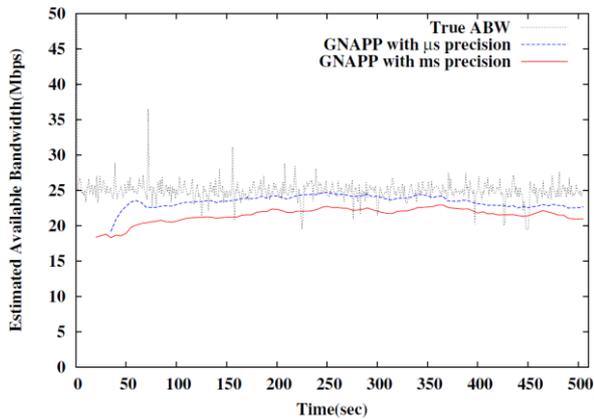


圖 7、高速網路環境中 Pareto ON-OFF 背景資料流下之可用頻寬估測結果( $M=30, W=25$ )。

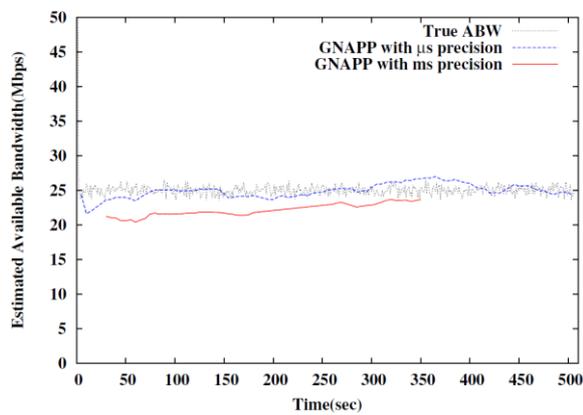


圖 8、高速網路環境中 Poisson 背景資料流下之可用頻寬估測結果( $M=30, W=25$ )。

面敘述。而在不同時戳精度下的估測結果，我們可發現使用微秒精度的估測結果仍然遠較毫秒精度的估測結果為佳。

由上面的實驗可觀察到在低速網路環境下，時戳精度以毫秒記錄跟以微秒記錄所得到的估測準確度並無太大的差距；但在高速網路環境下，時戳精度以毫秒記錄和以微秒記錄所得出的估測準確度就有相當大的差距。這是因為在高速網路環境下，封包通過中繼節點的時間延遲可能只有數十微秒，如果以精度為毫秒的時戳來記錄便會顯得不足，故而造成估測上較大的誤差。在 100 Mbps 的網路環境就有如此的誤差，那在下世代網路環境中將會造成更顯著的誤差。

### 3. ICMP 協定之時戳格式修改與建議

在現行 ICMP 協定中，每一個時戳是由一組 32 位元的空間來存放，ICMP TIMESTAMP 欄位上所記錄的時戳由於不需要記錄到日期的關係，所以捨棄了日期而從每日午夜 (GMT) 開始計算的毫秒數，其可設定之最大毫秒數為

Type = 13/14	Code = 1	ICMP Header Checksum
Identifier		Sequence Number
Originate Timestamp		
Receive Timestamp		
Transmit Timestamp		
Decimal of Originate	Decimal of Receive	Decimal of Transmit

圖 9、增加小數時戳欄位之 ICMP 封包格式。

86,400,000 毫秒(24 小時)，而現行的 32 位元可表示的十進位正數最大為 2,147,483,647，所以並不會發生溢位的問題，但如果想要精準至微秒的話，就必須可記錄到 86,400,000,000，故以現行 32 位元欄位的長度來說，就會顯得不足。

為了解決上述問題，建議的解決方法是在 ICMP 封包增加一列長度總共 32 位元的小數點欄位，如圖 9 所示。將此新增的 32 位元中的前 30 個位元用來記錄各個時戳的小數值，每一時戳各使用 10 個位元來分別記錄 Original Timestamp、Receive Timestamp、和 Transmit Timestamp 無法紀錄到的毫秒以下的小數點時戳。如此一來，便可以將時戳精度提升至微秒的等級，而此時戳精度將可符合下一代高速網路量測之所需。

ICMP Type 13/14 的 Code 欄位目前是都設為 0，代表使用的是毫秒的時戳精度。當要使用微秒精度的時戳時，就可將 Code 欄位設為 1，此時就要在原本三個時戳後面增加一列 32 位元欄位，如圖 9 所示。這種方法將可以相容於以往的設備，因為當 Code 為 0 時表示使用的是傳統的 ICMP 封包格式及毫秒精度的時戳，不需另外更改協定；當 Code 設為 1 時，表示需增設欄位以記錄毫秒以下的時戳，而較新的設備則可記錄更精準的時戳資料，以提供更準確與更廣泛的網路量測應用。

### 4. 結論

根據模擬的數值結果可看出，在高速網路環境下，採用微秒精度的時戳所得到的估測結果遠比採用毫秒精度的時戳之估測結果準確，至於採用毫秒精度的時戳所得到的估測結果在某些情形下之估測誤差甚至可以說已經完全無法忍受了。因此，提高時戳精度來增進量測準確度是勢在必行的，本文建議 ICMP 協定可以將 TIMESTAMP 欄位多增加一列 32 位元的欄位來記錄毫秒以下的小數點時戳，如圖 9 所示，如此便可為提升下一代網路量測的準確度。

## 參考文獻

- [1] M. Li, Y. L. Wu, C. R. Chang, "Available Bandwidth Estimation for the Network Paths with Multiple Tight Links and Bursty Traffic," *Journal of Network and Computer Application*, Vol. 36, No. 1, pp. 353-367, 2013.
- [2] A. Yarali, A. Cherry, "Internet Protocol Television (IPTV)," *IEEE TENCON*, pp. 1-6, 2005.
- [3] M. Mrak, N. Sprljan, E. Izquierdo, "An Overview of Basic Techniques Behind Scalable Video Coding," *46th International Symposium Electronics in Marine, ELMAR-2004*, pp. 597-602, June 2004.
- [4] V. Riberiro, M. Coates, R. Riedi, S. Sarvotham, B. Hendricks, R. Baraniuk, "Multifractal Cross-Traffic Estimation," *ITC Specialist Seminar on IP Traffic Measurement, Modeling, and Management*, Sept. 2000.
- [5] N. Hu, P. Steenkiste, "Evaluation and Characterization of Available Bandwidth Probing Techniques," *IEEE JSAC*, Vol. 21, No. 6, pp. 879-894, Aug. 2003.
- [6] J. Strauss, D. Katabi, F. Kaashoek, "A Measurement Study of Available Bandwidth Estimation Tools," *ACM SIGCOMM Conference on Internet Measurements*, pp. 39-44, Oct. 2003.
- [7] M. Zhang, C. Luo, J. Li, "Estimation Available Bandwidth Using Multiple Overloading Streams," *IEEE ICC2006*, pp. 495-502, June 2006.
- [8] C. Dovrolis, P. Ramanathan, D. Moore, "Packet Dispersion Techniques and a Capacity-Estimation Methodology," *IEEE/ACM Trans. Networking*, Vol. 12, No. 6, pp.963-977, Dec. 2004.
- [9] B. Melander, M. Bjorkman, P. Gunningberg, "A New End-to-End Probing and Analysis Method for Estimating Bandwidth Bottlenecks," *IEEE GLOBECOM'00*, pp.415-420, 2000.
- [10] M. Jain, C. Dovrolis, "End-to-End Available Bandwidth: Measurement Methodology, Dynamics, and Relation With TCP Throughput," *IEEE/ACM Trans. on Networking*, Vol. 11, No. 4, pp. 537-549, Aug. 2003.
- [11] M. Jain, C. Dovrolis, "Pathload: A measurement tool for End-to-End Available Bandwidth," *Passive and Active Measurements Workshop*, Fort Collins, CO, Mar. 2002.
- [12] V. Ribeiro, R. Riedi, R. Baraniuk, J. Navratil, L. Cottrell, "Pathchirp: Efficient Available Bandwidth Estimation for Network Paths," *Proc. PAM*, April 2003.
- [13] M. Jain, C. Dovrolis, "End-to-End Estimation of the Available Bandwidth Variation Range," *ACM SIGMETRICS'05*, pp. 265-276, June 6-10, 2005.
- [14] D. Kiwior, J. Kingston, A. Spratt, "PathMon, a Methodology for Determining Available Bandwidth Over an Unknown Network," *IEEE/Sarnoff Symposium on Advances in Wired and Wireless Communication*, April 26-27, 2004.
- [15] Z. Zhou, L. Cong, G. Lu, B. Deng, X. Li, "HATS: High Accuracy Timestamping System Based on NetFPGA," *International Journal of Future Generation Communication and Networking*, Vol. 3, No. 3, pp. 1-12, Sept. 2010.
- [16] EtherNID™, Accedian Networks.  
<http://www.accedian.com/>