How accurate are your traffic measurements?

Hiroshi TSUNODA
Tohoku Institute of Technology
35-1, Yagiyama Kasumi-cho, Taihaku-ku,
Sendai-shi, Miyagi, 982–8577 JAPAN
E-mail: tsuno@m.ieice.org

Glenn Mansfield KEENI
Cyber Solutions Inc.
ICR Bldg, 6-6-3, Minami Yoshinari, Aoba-ku,
Sendai-shi, Miyagi, 989-3204 JAPAN
E-mail:glenn@cysols.com

***Abstract*—Traffic measurement provides a useful insight into the dynamics of the network. Though traffic measurement is widely deployed, the quality and nature of the statistics obtained has not been closely scrutinized. Two of the simple aspects of traffic measurement, namely, volume measurement and peak measurement have significant implications in accounting, operations, security and quality of service management. In this paper, we take a closer look at the measurement practices widely deployed and discuss the inaccuracies inherent in the measurement.**

I. Introduction

Traffic measurement is widely deployed as an essential aspect of network monitoring and management. It provides a useful insight into the dynamics of the network. Two of the primary aspects of traffic measurement, namely, volume measurement and peak measurement have significant implications in accounting, operations, security and quality of service management.

For accounting management, the operator is required to have the user-wise bandwidth consumption figures as accurately as possible. Billing will be based on these figures. Further, in cases where the tariff is not linear but a step function of the traffic volume, the inaccuracies, if any, get amplified in the billed amount. In the operations and/or security management context, if some user is consuming an unfair share of the network bandwidth, a network administrator would want to be notified of it and would want to be able to identify the user and coerce the user's traffic to within normal limits. Moreover, with the rapid growth of multimedia applications such as movie-streaming and videophone, sophisticated quality of service management is required. The operator needs to understand the accurate bandwidth usage in the *service level agreement* (SLA) context, and would want to detect and track the SLA violation as quickly as possible [1].

Figure 1 illustrates the assumed network environment in this paper. A provider network provides the end users reachability to the external network via access networks, such as ADSL, FTTH, and cable internet. End users possess customer premises equipment (CPEs) (e.g., routers, switches, modems, and set-top boxes etc). The end user's network is connected via a CPE. An access router in the provider network is a terminating point for the access networks. A cable modem terminating system (CMTS) in cable internet, a digital subscriber line access multiplexer (DSLAM) and an edge router

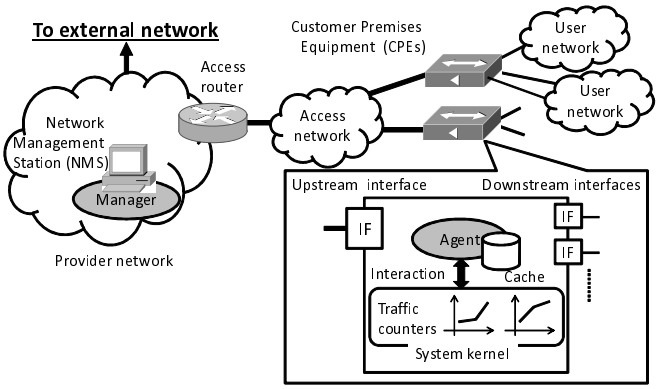

Fig. 1. Assumed environment

are examples of an access router. For network management purposes, a manager is located on a network management station in the provider network, and agents are deployed on the CPEs.

In large distributed networks, traffic measurement, by its nature is distributed. To obtain the bandwidth utilization by an end user, generally the CPE connecting the end-user's network to the provider's network is probed/queried. In general, related counters of traffic volume, errors and other statistics are in the system kernel of the CPE. An agent on the CPE provides the value of such counters to a remote manager in response to a request from the manager. Some (management) protocol is used for this request and response. The manager calculates bandwidth utilization based on the retrieved data. In this paper we limit ourselves to this method of traffic measurement and discuss the inaccuracy of the bandwidth utilization estimate.

The remainder of the paper is organized as follows. In Sec. II, we explain the basic procedure for the measurement of bandwidth utilization and briefly describe the causes of inaccuracy of the measurement. We describe related works in Sec. III. Section IV discusses the importance and difficulties of the accurate peak measurement. The various factors related to the measurement error in the volume measurement are explained in Sec.V. In Sec.VI, we discuss the new challenges, traffic measurement in mobile and cloud environments and generality of the problems pointed out in this paper, followed by conclusions in Sec. VII.

II. Basic procedure for network traffic measurement

As shown in Fig.1, typically, an agent maintains counters of various facets of the network traffic. Examples are traffic volume, in number of packets and/or number of bytes, errors etc. These counters are in general cumulative. A manager samples the counters, computes the delta of the two samples and thereby computes the bandwidth utilization for the interval between the two samples.

In Fig. 2, v_t denotes the value of the cumulative traffic counter at the agent at time t. The manager polls the agent periodically at some interval Δt and retrieves the corresponding sample v_t at time t. If the counter corresponds to upstream or downstream traffic, from two samples v_{t_i} and $v_{t_{i+1}}$ the bandwidth utilization (Bw) between t_i and t_{i+1} is calculated as

$$Bw = \frac{\Delta v}{\Delta t} = \frac{v_{t_{i+1}} - v_{t_i}}{t_{i+1} - t_i}. \quad (1)$$

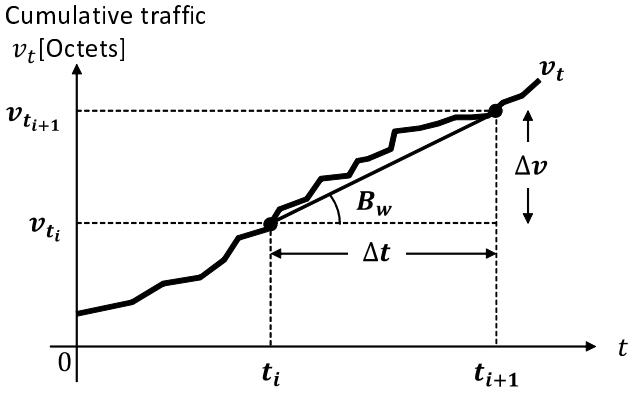

Fig. 2. Basic concept for calculating bandwidth utilization

The obtained Bw is averaged over the time interval Δt. Thus the value of Δt affects the measurement of bandwidth utilization peaks. One important consideration in setting Δt is the management traffic overhead. In order to keep the management traffic overhead low Δt is kept large. Currently, typical values of Δt in network management tools range from tens of seconds to several minutes.

The measurement accuracy is implicit in Eq.2. For accurate measurement Δv and the corresponding Δt must be accurately measured. In real world situations this turns out to be difficult. In the ideal case, v_{t_i} is the value of the counter at time t_i. However, in reality, in the absence of explicit time tags, t_i can only be approximately estimated. Fig. 3 shows a sequence diagram of the i-th polling cycle. According to this figure, v_{t_i} should be the value at time t_i^{GCnt}, the time at which the agent looked up the traffic counter. However, in the absence of explicit time tags, the manager will not know the exact value of t_i^{GCnt}. As an alternative to t_i^{GCnt}, a manager will generally use t_i^{SRq} (t_i^{RRq}), the time when the manager sent the request to (received the response from) the agent, as an approximation. We denote the value used at the manager by $\widehat{t_i}$. The manager will have reasonably accurate values of t_i^{SRq} (t_i^{RRs}). However, since the request/response latency, (d_i^{Rq} and d_i^{Rs}), between the manager and the agent, and, the request processing time at the agent (d_i^{Prc}) are all variable depending on the network conditions and processing load at the agent, t_{i+1}^{SRq} - t_i^{SRq} is not equal to t_{i+1}^{GCnt} - t_i^{GCnt}.

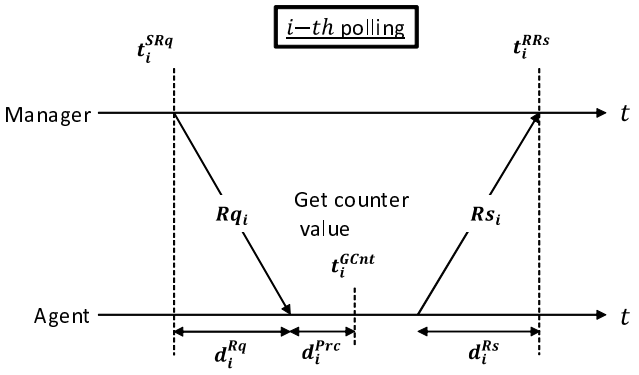

Fig. 3. Sequence diagram of a polling process

Further, depending on agent implementations , the manager will likely receive only an approximation of v_{t_i}. As shown in Fig. 1, the agent, when queried, refers to a traffic counter which is generally some kernel variable and provides the corresponding value to the manager. This reference in general involves multiple lookups of kernel tables. To optimize the load due to such lookups, the lookups are not done in realtime, the looked up value is cached and reused for a small cache-lifetime. A fresh lookup is done if there is no cache, or the cache is older than cache-lifetime. So, the counter value, $\widehat{v_t}$, returned by the agent is only an approximation of the value v_t of the traffic counter at time t.

Consequently, the manager computes the bandwidth utilization based on the $\widehat{v_t}$ and $\widehat{t_i}$ as follows.

$$\widehat{B_w} = \frac{\widehat{\Delta v}}{\widehat{\Delta t}} = \frac{\widehat{v}_{t_{i+1}} - \widehat{v}_{t_i}}{\widehat{t}_{i+1} - \widehat{t}_i} \quad (2)$$

The inaccuracy in the computed traffic rate is given as

$$B_{w_{err}} = \widehat{B_w} - B_w. \quad (3)$$

III. Related works

Several methods are used for accurate traffic measurement. Offline measurements using traffic traces is useful for experimenting, testing and diagnosis. Arbitrary level of detail, precision and accuracy may be diagnosis. An arbitrary level of detail, precision and accuracy may be achieved in offline measurements, by analyzing the traffic traces. For example, in [2], the packet-size statistics of IPTV traffic, such as packet size distribution and the frequency of retransmission and reordering, are analyzed with offline measurements using traffic trace.

As another example, deep packet inspection, a technology for analyzing packet payloads in real-time, has been used for

the purpose of detecting and restricting the communication of P2P applications [3].

In these cases the only source of inaccuracy will be the packets that were dropped by the traffic trace collector. Generally speaking, however, CPEs do not have enough resources in terms of storage and processing power. Thus, this method is not a good candidate for routine measurements.

Per-flow traffic measurement is an important way for getting accurate user-wise or application-wise statistics. It is useful for identifying *elephant flows* (flows that include a huge number of packets) and for usage-based billing. However, the main problem of the per-flow measurement is its lack of scalability [4]. Cisco NetFlow [5] uses a packet sampling technique for improving the scalability, but it introduces significant measurement error [6]. The current trend is to implement measurement functions in high-speed but expensive SRAM [6].

Use of the Internet Protocol Detail Records (IPDR) [7] mechanism has been spreading in recent years, especially in the area of cable internet. IPDR defines a data format and a protocol for exporting network measurement and management information [8]. Due to its flexibility, efficiency and scalability, IPDR is expected as a promising technology for the future network management and measurement.

In the context of the traffic measurement using simple network management protocol (SNMP) [9], a High Resolution Traffic Measurement MIB (HRTM MIB) is proposed [10]. The main concept of the MIB is to aggregate values of the traffic counter at the agent and to fetch the aggregated data in bulk manner. This bulk transfer of the aggregated data results in the reduction of polling frequency, bandwidth consumption between the manager and the agent, and processing cost at the agent. The concept of HRTM MIB is also published as the Managed Object Aggregation MIB (AggrMIB) in RFC 4498 [11]. As an application of AggrMIB, [12] tackles the inaccuracy problems in the network management in a mobile environment. During the monitoring from an agent on a mobile node, such as a car, accuracy suffers from the intermittent characteristics of wireless link connectivity and the agent activity. Hence, a new polling technique called tagged and persistent polling method has been proposed for improving accuracy. In the proposed method, the agent adds time-tags to the data and stores the tagged data in a time sequenced manner at regular intervals. Since the agent itself stores pairs of data and time information (i.e., v_{t_i} and t_i in Fig. 2), the manager can retrieve the stored data later even if the wireless link is disconnected between the manager and the agent temporarily. If we can adopt this method for traffic monitoring of mobile nodes, the manager will be able to get an accurate view of traffic variation on mobile nodes. Although the use of AggrMIB is a promising approach in the SNMP context, in this paper, we limit our scope to the typical situation where the agent does not have explicit time-tagged data for retrieval.

IV. Granularity of the measurement

In this section, we discuss the problems on the fine-grained measurement of bandwidth utilization. It is known that the Internet traffic has burstiness over a wide span of time scales [13]. Traffic bursts generate a sharp shortlived increase in the rate of network traffic. As quantified in [14], the variability of link load on small time scales is larger than on large time scales. To measure and detect such peak-rates is important in network management. In order to observe the traffic peak-rate on small time scales, the fine-grained measurement with frequent polling is required.

Below are some examples of situations where the fine-grained traffic measurement is required.

Fault management
: Some network faults manifest themselves in the traffic patterns. For example, if a network interface has a problem of intermittent connectivity, then the traffic pattern seen on that interface in enough detail i.e. with appropriate polling intervals, would reveal this in periods of zero traffic. But if on the other hand the polling interval is larger than the periods of zero traffic, the traffic will be seen as an average over the polling interval and the fault will not be noticed.

Configuration management
: Peak rate information is required for network design [15]. The network administrator should be aware of the peak traffic rate that the network will need to handle. The operator will then put in place checks to ensure that these peak rates are not exceeded.

Accounting management
: Currently, users are charged on the basis of traffic volume (average rate), peak rate, and/or utilization time. As we will see later, bursty traffic can adversely affect the performance and health of networks. The burstiness of the traffic can be an additional criteria for accounting.

Performance management
: Peak rate of traffic gives the administrators important information about the symptoms of performance degradation. High peak rate indicates the existence of traffic bursts and the bursts will be the cause of packet losses and jitter. Packet losses result in retransmission and degraded TCP throughput. Jitter degrades the quality of real-time contents.

Security management
: The traffic bursts, in some cases, indicate the existence stealthy DoS attacks. Low-rate TCP-Targeted denial of services attacks [16], also known as pulsing DoS attacks, use sharp, shortlived traffic bursts. If we can make a precise traffic pattern capturing traffic bursts, it will be a good signature of such attacks.

However, despite the importance of the fine-grained measurement, typical values of Δt in Fig. 2 are not small enough in network management tools because such polling consumes various resources such as network bandwidth between the

manager and the agent, CPU time on the agent [10], etc.. For example, MRTG [17], a widely deployed tool for traffic measurement, uses five minutes as the minimum value of Δt. But a large value of Δt may result in bursty nature of the network traffic, if any, may passing unnoticed. In other words, sharp bursts of split second duration, cannot be captured using the typical values of Δt used in network management tools.

Small Δt is good for understanding the traffic characteristics more accurately. However, a smaller Δt will increase the error rate due to inaccuracy in estimating the Δt, as we will see in the next section.

Another solution would be to build into the agent a peak measurement function. In this case the peak-rate could be measured for intervals δt, smaller than Δt, for each Δt. This peak-rate would be the representative peak traffic rate for the interval Δt. To the best of our knowledge, such implementations are not widely deployed.

V. Factors of Measurement Errors in Volume Measurement

A. Ambiguity of measured value acquisition time

In this section, we discuss the factors that contribute to errors in traffic volume measurement.

The manager polls the agent periodically. Although the interval of the polling is determined by the manager, the actual value of the interval is fluctuates due to various factors.

Figure 4 illustrates two successive (i-th and $i+1$-th) polls, and Table I summarizes the definitions of variables illustrated in Fig. 4.

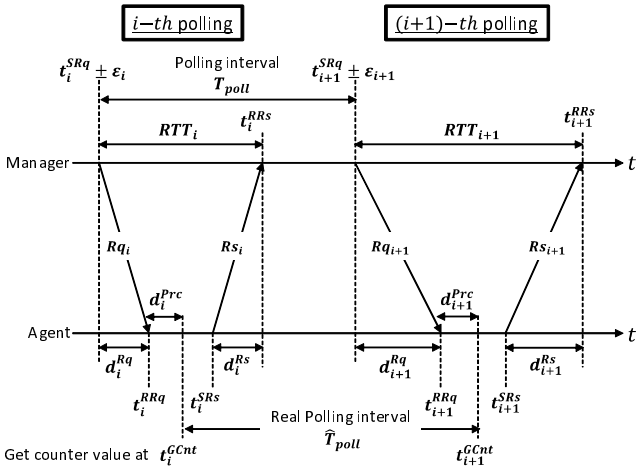

Fig. 4. Timestamp variation

In i-th and $i+1$-th polling, the manager sends request to the agent at time t_i^{SRq} and t_{i+1}^{SRq}, respectively. Only these two values can be controlled by the manager, and other values are beyond the manager's control. For periodic polling, the manager determines the constant value of T_{poll} and adjusts t_i^{SRq} and t_{i+1}^{SRq} based on T_{poll}. The actual times at which the agent accesses the counter are t_i^{GCnt} and t_{i+1}^{GCnt}, respectively.

TABLE I
Definitions of Variables

Variables	Descriptions
t_i^{SRq}	Time at which request Rq_i is sent by manager
t_i^{RRq}	Time at which request Rq_i is received by agent
t_i^{SRs}	Time at which response Rs_i is sent by agent
t_i^{RRs}	Time at which response Rs_i is received by manager
t_i^{GCnt}	Time at which agent does a traffic counter lookup.
d_i^{Rq}	One way delay experienced by request Rq_i
d_i^{Prc}	Processing time for request Rq_i
d_i^{Rs}	One way delay experienced by response Rs_i
RTT_i	Round trip time for the i-th poll
T_{poll}	Polling interval. The interval between two successive request sent by the manager.
$\widehat{T}_{poll}$	Actual polling interval. The interval between two successive traffic counter lookups by the agent.

This means that the actual data interval, $\widehat{T}_{poll}$, is the interval between t_i^{GCnt} and t_{i+1}^{GCnt}. Thus, in order to accurately estimate the bandwidth utilization, $\widehat{T}_{poll}$ should be used as Δt in Eq.1. However, the manager cannot know the accurate value of $\widehat{T}_{poll}$, and bandwidth utilization is estimated by using T_{poll} as $\widehat{\Delta t}$ as shown in Eq.2. As a result, Eq.2 becomes

$$\widehat{B_w} = \frac{\widehat{\Delta v}}{\widehat{\Delta t}} = \frac{v_{t_{i+1}^{GCnt}} - v_{t_i^{GCnt}}}{t_{i+1}^{SRq} - t_i^{SRq}} = \frac{v_{t_{i+1}^{GCnt}} - v_{t_i^{GCnt}}}{T_{poll}}. \quad (4)$$

If $\widehat{T}_{poll}$ is always equal to T_{poll}, t_i^{GCnt} and t_{i+1}^{GCnt} will vary due to several factors such as one-way delay from the manager to the agent (d_i^{Rq} and d_{i+1}^{Rq}), drift at the manager in sending requests Rq_i and Rq_{i+1} (ϵ_i and ϵ_{i+1}), and the request processing time at the agent (d_i^{Prc} and d_{i+1}^{Prc}) as shown in Fig. 4. Let $\widehat{T}_{poll} = T_{poll} + e$ where e is error due to the factors described above. If $e > 0$, the actual bandwidth utilization is larger than the estimated one, and vice verse. Obviously, in the case that T_{poll} is large enough compared with e, the inaccuracy due to the fluctuation of $\widehat{T}_{poll}$ can be minimized. Typically, the variations of d_i^{Rq} and d_{i+1}^{Rq} are in the order of milliseconds. Also, the variations of ϵ_i, ϵ_{i+1}, d_i^{Prc} and d_{i+1}^{Prc} are in the order of microseconds. A typical setting of T_{poll} (e.g. five minutes) is generally large enough to make these variations insignificant, but, setting T_{poll} to smaller values like one second for the accurate peak measurement, will make the variations significant.

To minimize the measurement error, a manager is required to accurately estimate $\widehat{T}_{poll}$. In the SNMP context, the Managed Object *sysUpTime* [18] is useful for this purpose. *sysUpTime* indicates the time (in hundreds of a second) since the agent was last re-initialized. In the absence of explicit time tags, the *sysUpTime* object fetched from the agent along with the traffic counter values is a good, not exact, estimate of t^{GCnt}.

B. Discrete nature of the counter information

As described in Sec.II, an agent on the CPE does kernel table lookups and caches the obtained value to reduce the load

due to lookups. Therefore, the counter information provided by the agent is updated discretely.

Figure 5 depicts this problem. Even if the traffic rate is constant and the value of cumulative counter v_t increases linearly, $\widehat{v}_t$, the counter value returned by the agent, looks like a step function. Let r is the constant rate of traffic and $\tau_j (j = 1, 2, 3, 4)$ is the time of the update of the counter value, $\widehat{v}_t$ can be denoted as follows.

$$\widehat{v}_t = \begin{cases} v_{\tau_j} & (t = \tau_j) \\ v_{\tau_{j-1}} & (t \neq \tau_j) \end{cases} \tag{5}$$

The update interval T_{upd} is cache-lifetime which is implementation dependent.

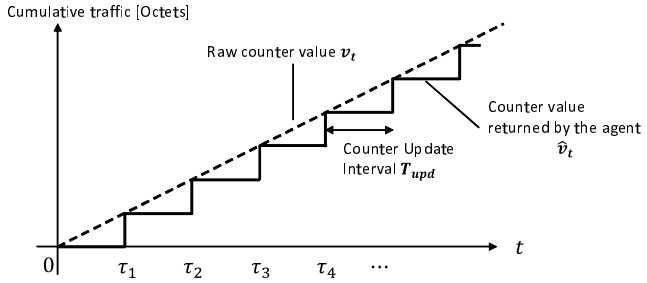

Fig. 5. Discrete nature of the counter information

The effects are shown in Figs. 6 and 7 corresponding to a net-snmp on a Linux device and a Catalyst 3550 device. The figures show that the value of the *ifInOctets* counter is updated every 15 seconds on the net-snmp agent, whereas for the Cisco Catalyst 3550, the value is updated every second.

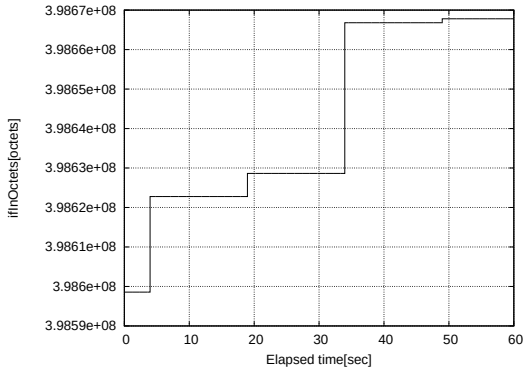

Fig. 6. Example of discrete update (Net-snmp)

In the next experiment, we computes the bandwidth utilization using the measurement results of the polling with the interval of five seconds ($T_{poll} = 5seconds$). Throughout this experiment, a device was subjected to a constant bit rate (10Mbps) UDP stream. There was little background traffic. The bandwidth utilization is expected to be a straight line.

Figure 8 shows the estimated bandwidth utilization in the agent of net-snmp. In spite of the constant bit rate, that is not

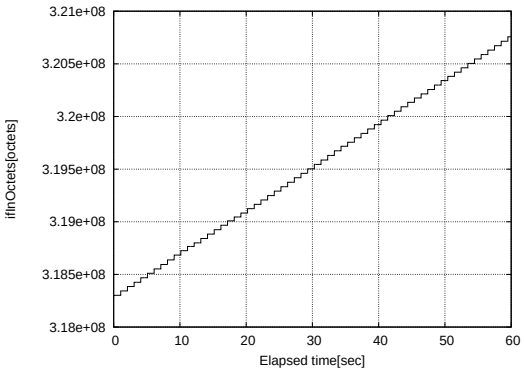

Fig. 7. Example of discrete update (Cisco Catalyst 3550)

the case as can be seen in this figure. This is because that T_{poll} is smaller than T_{upd}. In this case, only every third poll brings the updated value of traffic counter because T_{upd} is three times larger than T_{poll}. Since other polls bring the value which is the same as the previous one, the estimation result becomes zero.

It is clear from this result that T_{poll} should be larger than T_{upd}. However T_{upd} is implementation dependent and is not provided publicly. Thus, network operators have no choice but to set T_{poll} with a large value (e.g., five minutes in MRTG) conservatively and this setting limits the granularity of the measurement as discussed in Sec.IV.

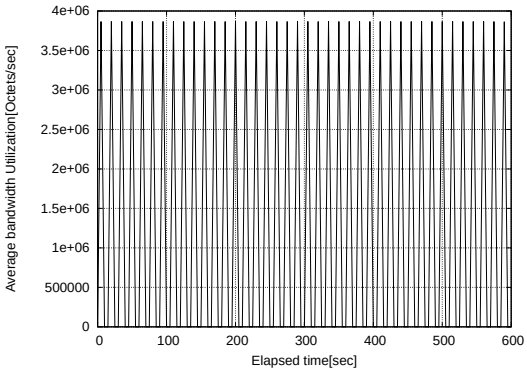

Fig. 8. Estimated bandwidth utilization (Net-snmp) T_{poll} = 5 sec

We conducts the same experiment for Cisco Catalyst 3550. Fig. 9 shows the experimental result. In Cisco Catalyst 3550, T_{upd} is one second and smaller than T_{poll} of five seconds. Nevertheless, we can find several shortlived increase and decrease of the utilization in Fig. 9.

These shortlived increase and decrease are caused by the ambiguity of the measured value acquisition time t_i. Fig.10 explains the reason. We assume that the request from the manager arrives during the target period (from τ_j to τ_{j+1}), and the value acquisition is performed during this period. However, in the actual case, due to the variation of the one-way delay and the request processing time, the value acquisition may be

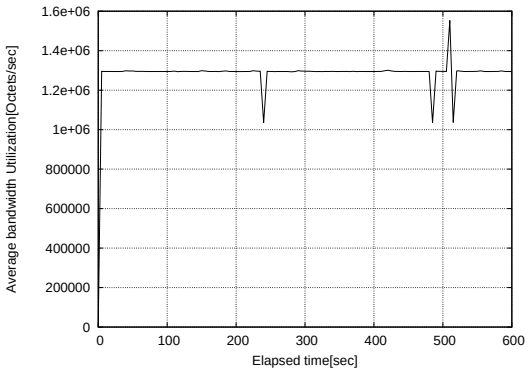

Fig. 9. Estimated bandwidth utilization (Cisco Catalyst 3550) T_{poll} = 5 sec

performed before τ_j or after τ_{j+1}.

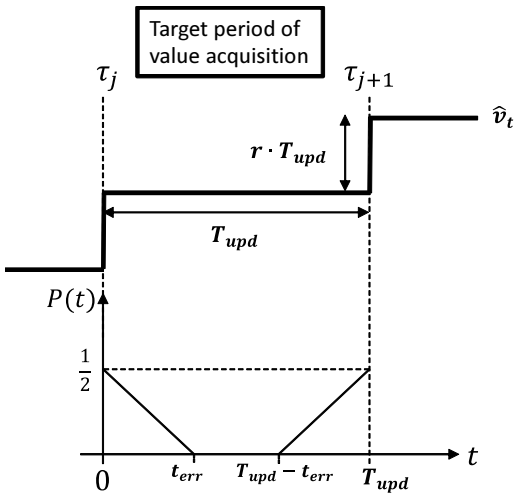

Fig. 10. Estimation of the error occurrence probability

In the lower half of Fig.10, P(t) gives the probability that the value acquisition is performed before τ_j or after τ_{j+1} where t is the planned time of the value acquisition. In this figure, we assume that the actual acquisition time is given by the uniform distribution from $t-t_{err}$ to $t+t_{err}$, for simplicity. In this case, $P(t)$ is given as follows.

$$P(t) = \begin{cases} -\frac{1}{2t_{err}}t + \frac{1}{2} & (0 \le t \le t_{err}) \\ 0 & (t_{err} < t < T_{upd} - t_{err}) \\ \frac{1}{2t_{err}}t + \frac{1}{2} & (T_{upd} - t_{err} \le t \le T_{upd}) \end{cases} \quad (6)$$

If the planned acquisition time t is randomly selected from 0 to T_{upd}, the measurement error occurs with the probability given by the following equation.

$$\frac{2t_{err}}{T_{upd}} \cdot \frac{t_{err}}{4} = \frac{{t_{err}}^2}{2T_{upd}}. \quad (7)$$

When the measurement error occurs, the value sampled by the manager becomes $r \cdot T_{upd}$ octets smaller or larger than the value that should truly be sampled by the manager. Then the estimated bandwidth utilization becomes $\frac{r \cdot T_{upd}}{T_{poll}}$ octets/second smaller or larger than the actual.

In this experiment (r = 10Mbps = 1.25MBps, T_{upd} = 1 second and T_{poll} = 5 second), the theoretical value of measurement error is $250,000$ octets/second. This value approximately is equal to the amounts of increase or decrease that we can see in Fig.9.

According to the discussion in this subsection, the large value of T_{poll} is better for minimizing the measurement error of the estimation of bandwidth utilization. However, the large value of T_{poll} incurs another problem as discussed in the next subsection.

C. Counter may be reset

The traffic counter maintained by the agent has a finite capacity. In many cases these are 32 bit counters. When the traffic counter reaches the maximum value, at the next increment its value returns to zero. This leads to a problem in computing the traffic rate. There are two approaches here.

a. Ignore such cases.

if $v_{t_{i+1}} \ge v_{t_i}$

$$\widehat{r}_{calc} = \frac{\widehat{v}_{t_{i+1}} - \widehat{v}_{t_i}}{\widehat{t}_{i+1} - \widehat{t}_i}$$

else

$$\widehat{r}_{calc} = unknown.$$

This approach has the drawback of loss of data. The operator is unable to charge the user for the legitimate service provided. In case the counter is small and the traffic is high then this may happen relatively frequently.

b. Guess that the counter has been reset due to

b-1. Counter overflow.

if $v_{t_{i+1}} \le v_{t_i}$

$$v_{t_{i+1}} += CounterMaxvalue$$

$$\widehat{r}_{calc} = \frac{\widehat{v}_{t_{i+1}} - \widehat{v}_{t_i}}{\widehat{t}_{i+1} - \widehat{t}_i}$$

The assumption here is that the counter overflowed only once during the polling interval.

b-2. System reboot. This can be verified from some variable on the agent like *sysUpTime*.

if $sysUpTime\,(t_{i+1}) \ge sysUpTime\,(t_i)$

if $v_{t_{i+1}} \le v_{t_i}$

$$v_{t_{i+1}} += CounterMaxvalue$$

$$\widehat{r}_{calc} = \frac{\widehat{v}_{t_{i+1}} - \widehat{v}_{t_i}}{\widehat{t}_{i+1} - \widehat{t}_i}$$

else

$$\widehat{r}_{calc} = unknown.$$

This case leads to data loss. Since the agent is rebooted everytime the network device is restarted, this data loss can be controlled/caused by a user. If the polling interval is reasonably large a user may

fool the Traffic measurement system and associated accounting system to enjoy a free ride!

VI. Discussion

In this work we have restricted ourselves to the classical case of traffic measurement where the manager 'has access to' the agent and thereby obtains the traffic related data. This situation is rapidly changing with mobile devices and networks. The mobile entities may have poor connectivity. One immediate consequence of this is the wide fluctuations in the latency, which leads to added inaccuracies in the estimation of time. In Cloud environments where dynamic provisioning is the keyword, the measurement parameters themselves will change with time. In such cases it is imperative to have appropriate meters set up within the Cloud which will react quickly enough and provide a simple interface to the measurement manager.

Though we have restricted ourselves to traffic measurement, the same problems are encountered in energy management. In energy management the necessity of monitoring peak-rates is critical as the penalties for breaching the peak-rate limit are heavy. Perhaps for this reason, many energy management meters provide time tagged data.

VII. Conclusion

In this work, we have examined the issues of inaccuracies in traffic measurement widely deployed in the current network management area. Traffic measurement is performed through polling from a manager to agents in distributed-manner. To understand the traffic characteristics accurately, a small polling interval is preferred. However, the small interval incurs inaccuracy problems because the manager will not know when the agent looks up the traffic counter exactly. For accurate measurements explicit time tagged data is essential. In the absence of time tagged data, one needs to be careful in choosing the polling interval and be aware of the limited accuracy of the results.

References

[1] MUROOKA Takahiro, HASHIMOTO Masashi, and MIYAZAKI Toshiaki. A High Time-Resolution Traffic Monitoring System. *IEICE transactions on information and systems*, 87(12):2618–2626, 2004-12-01.

[2] Y.J. Won, Mi-Jung Choi, Byung-Chul Park, J.W. Hong, Hee-Won Lee, Chan-Kyu Hwang, and Jae-Hyoung Yoo. End-user iptv traffic measurement of residential broadband access networks. In *Network Operations and Management Symposium Workshops, 2008. NOMS Workshops 2008. IEEE*, pages 95 –100, april 2008.

[3] Milton Mueller and Hadi Asgharia. Deep Packet Inspection and Bandwidth Management: Battles Over Bittorrent in Canada and the United States. *Telecommunications Policy*, In Press.

[4] Cristian Estan and George Varghese. New directions in traffic measurement and accounting: Focusing on the elephants, ignoring the mice. *ACM Trans. Comput. Syst.*, 21(3):270–313, August 2003.

[5] Cisco ios netflow. http://www.cisco.com/en/US/products/ps6601/products_ios_protocol_group_home.html.

[6] Tao Li, Shigang Chen, and Yibei Ling. Fast and compact per-flow traffic measurement through randomized counter sharing. In *INFOCOM, 2011 Proceedings IEEE*, pages 1799 –1807, april 2011.

[7] Ipdr/sp protocol specification version2.3. http://www.ipdr.org/public/DocumentMap/SP2.3.pdf.

[8] Wenxuan Guo, A. Dunstan, and J. Finkelstein. Using ipdr to transfer network management and measurement information. In *Communications Quality and Reliability (CQR), 2011 IEEE International Workshop Technical Committee on*, pages 1 –6, may 2011.

[9] J. Case, R. Mundy, D. Partain, and B. Stewart. Introduction and Applicability Statements for Internet-Standard Management Framework. RFC3410, December 2002.

[10] Glenn Mansfield, Sandeep Karakala, Eep Karakala, Takeo Saitoh, and Norio Shiratori. High Resolution Traffic Measurement. In *Proc. of A workshop on Passive and Active Measurements on the Internet(PAM2001)*, pages 67–73, 2001.

[11] G. Keeni. The Managed Object Aggregation MIB. RFC4498, May 2006. http://www.ietf.org/rfc/rfc4498.txt.

[12] Glenn Mansfield Keeni, Kazuhide Koide, Takeo Saitoh, and Norio Shiratori. A bulk-retrieval technique for effective remote monitoring in a mobile environment. In *Proceedings of the 20th International Conference on Advanced Information Networking and Applications - Volume 01*, AINA '06, pages 889–894, Washington, DC, USA, 2006. IEEE Computer Society.

[13] Hao Jiang and Constantinos Dovrolis. Why is the internet traffic bursty in short time scales? *SIGMETRICS Perform. Eval. Rev.*, 33(1):241–252, June 2005.

[14] Remco Meent van de, Aiko Pras, Michel Mandjes, Hans Berg van den, and Lambert Nieuwenhuis. Traffic Measurements for Link Dimensioning: A Case Study. In *Proc. of 14th IFIP/IEEE International Workshop on Distributed Systems: Operations and Management (DSOM 2003)*, Lecture Notes in Computer Science, pages 106–117, October 2003.

[15] R. van de Meent, A. Pras, M. R. H. Mandjes, J. L. van den Berg, F. Roijers, L. J. M. Nieuwenhuis, and P. H. A. Venemans. Burstiness predictions based on rough network traffic measurements. In *Proc. of the 19th World Telecommunications Congress (WTC/ISS 2004), Seoul, Korea*, page 6, September 2004.

[16] A. Kuzmanovic and E. Knightly. Low-Rate TCP-Targeted Denial of Service Attacks and Counter Strategies. *IEEE/ACM Transactions on Networking*, 14(5), October 2006.

[17] Tobi Oetiker's MRTG - The Multi Router Traffic Grapher. http://oss.oetiker.ch/mrtg/.

[18] R. Presuhn. Management Information Base (MIB) for the Simple Network Management Protocol (SNMP). RFC3418, December 2002.