A New Technique for Link Utilization Estimation in Packet Data Networks using SNMP Variables

S. Amarnath and Anurag Kumar*
Dept. of Electrical Communication Engineering
Indian Institute of Science, Bangalore - 560 012, INDIA
(email: anurag@ece.iisc.ernet.in)

Abstract

On-line estimation of performance measures, such as the utilization of a digital communication link, is useful in the monitoring and management of packet data networks. In this paper, we define a notion of link utilization when the packet arrival process at the link is nonstationary. We develop a scheme for accurate monitoring of the link utilization, using the easily available SNMP variables. Simulation results and on-line measurement results on a link carrying live traffic in a wide area data network are provided.

1 Introduction

Most network management systems for packet switched data networks are based on information provided by SNMP (Simple Network Management Protocol) [4], a popular protocol for gathering traffic measurements from network elements (routers, bridges, hosts, etc.,) and exercising elementary control actions. SNMP provides raw information in the form of simple counters or status variables. In our research, our aim has been to turn the raw data provided by SNMP into useful information about the data network (see [3]).

In this paper we report the results of our efforts to develop algorithms for estimating the utilization of a digital communication link in a wide area packet switching data network. We use some of the SNMP variables to estimate the link utilization when the traffic carried by the link is nonstationary. We first point out the problem with the usual *sliding window* based measurement; then we propose, analyse and evaluate an *adaptive window* based technique.

2 Definition of Link Utilization

The packets arriving at a link for transmission can be viewed as "customers" arriving to a queue. At any time, the link (i.e., the server of this queue) is either busy or idle. The link occupancy thus forms a $\{0,1\}$ valued process; a sample path is shown in Figure 1. Let us denote the queue length process by $\{X(u), u \geq 0\}$; i.e., $X(u)$ is the queue length at time u. The server occupancy process is given by $\{1_{\{X(u)>0\}}, u \geq 0\}$, where $1_{\{.\}}$ is the indicator of the set $\{.\}$. The long run average occupancy (or utilization) ρ of the link is given by,

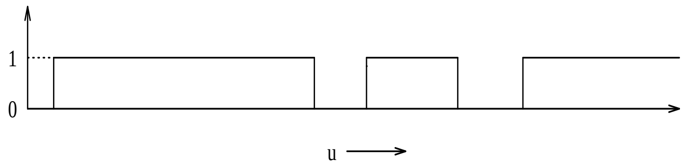

Figure 1: Server occupancy.

$$\rho = \lim_{t\to\infty} \frac{1}{t} \int_0^t 1_{\{X(u)>0\}} du$$

This long run utilization is useful only if the queue operates in a "stationary" environment. In practice, the rate of the arrival process is time varying; for example, the server utilization could be as high as 90% for some time, while a bandwidth "hogging" application like file transfer is on, and go down to a low value after it finishes. Clearly, the long run utilization gives the wrong picture in this case. This motivates us to seek a definition of utilization for a link with a time-varying packet arrival rate. Such a notion of utilization would be useful in adaptive traffic control algorithms.

To establish such a definition, we assume that the arrival process has a finite number of states, in each of which it has a well defined long-run average rate. A $MMPP$ (Markov Modulated Poisson Process) [1] is an example of such a process, in which arrivals occur as a Poisson process whose rate is governed by a finite state, continuous time Markov chain.
Definition: *We define the utilization $\rho(t)$ at time t as the long run average utilization of the server if the arrival process were to stay in the state it is in at time t.*

In this paper, we are interested in obtaining a good estimate $\hat{\rho}(t)$ of $\rho(t)$. We wish to obtain this estimate using easily available information, like SNMP variables. Note that more detailed information (such as packet arrival epochs and packet lengths) is usually not available.

3 Link Model

Throughout this paper, for the purpose of analyzing the estimation schemes we develop, we assume the arrival process to be a two state $MMPP$, whose modulating Markov chain has the transition rate diagram shown in Figure 2. When the arrival process is in state $i \in \{0,1\}$, the packet arrival process is Poisson with rate λ_i, with $\lambda_0 < \lambda_1$. If we denote the

*This work was supported by the Department of Electronics (Govt. of India), through its Education and Research Network (ERNET) Project.

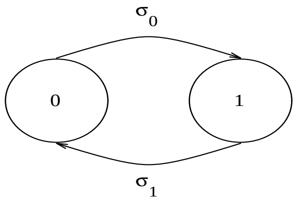

Figure 2: Arrival process for the simulation model.

fraction of time spent in state 1 by ϕ, then $\phi = \frac{\sigma_0}{\sigma_0+\sigma_1}$. When the arrival process is in state $i \in \{0,1\}$, we denote the link utilization by ρ_i. We assume that the packet transmission times are exponentially distributed with mean $\frac{1}{\mu}$. Clearly, $\rho_i = \frac{\lambda_i}{\mu}$. The "Poisson and exponential" assumptions are only to facilitate some analysis, and are not necessary for the development of the estimation schemes. Observe that in our model, $\rho(t) = \rho_i$ where $i \in \{0,1\}$ depending on the state of the Markov chain at time t.

In the following sections, we develop schemes for estimating $\rho(t)$. To be able to analyze the performance of these schemes, we define a performance metric called the *mean error*. If we let ϵ denote the mean error, we define,

$$\epsilon = \lim_{t\to\infty} \frac{1}{t} \int_0^t E[|\hat{\rho}(u) - \rho(u)|]du$$

where $\hat{\rho}(t)$ is the estimate at time t, and $\rho(t)$ is the actual utilization of the link at time t.

4 Sliding Window Scheme

With a fixed size window of W_s seconds, we can estimate $\rho(t)$ at time t as the average occupancy of the link in the previous W_s seconds. Throughout this paper, we let $\hat{\rho}_W(t)$ denote the estimate

$$\hat{\rho}_W(t) = \frac{1}{W} \int_{t-W}^t 1_{\{X(u)>0\}}du$$

Hence the sliding window estimate is $\hat{\rho}_{W_s}(t)$. Choosing small values for W_s would lead to noisy measurements. From Figure 1, we can see that choosing a very small value for W_s would give us mostly '1's and '0's. Choosing large values for W_s would result in averaging over long periods. Most of the time, W_s would straddle periods in which the arrival process is in different states and hence measure a wrong value of utilization. A natural question to ask is whether there is an optimal value of W_s that minimizes the measurement error.

To answer this question, we simulate an $MMPP/M/1$ queue. In the simulation the window slides in small discrete steps, yielding $\hat{\rho}_{W_s}(t_i), i \geq 1$. An estimate of the mean error is given by,

$$\epsilon = \frac{1}{N} \sum_{i=1}^{N} |\hat{\rho}(t_i) - \rho(t_i)|$$

where, N is the number of time steps, $\hat{\rho}(t_i)$ is the estimate at t_i, and $\rho(t_i) \in \{\rho_0, \rho_1\}, \rho_0 < \rho_1$, depending on the state of the system at that t_i. We simulate our model for different values of ϕ and σ_1. We choose the value of $\frac{1}{\sigma_1}$ to be $5000 \times \frac{1}{\mu}$ and $500 \times \frac{1}{\mu}$, where $\frac{1}{\mu}$ is the mean packet transmission time.

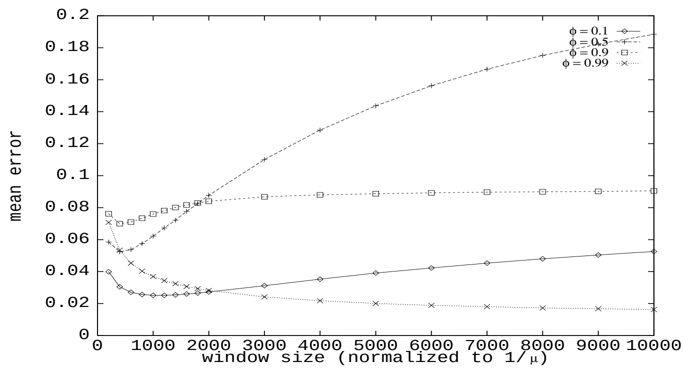

Figure 3: Mean error vs. sliding window size, with $\frac{1}{\sigma_1} = 5000 \times \frac{1}{\mu}$, $\rho_0 = 0.2$ and $\rho_1 = 0.7$, for different values of ϕ.

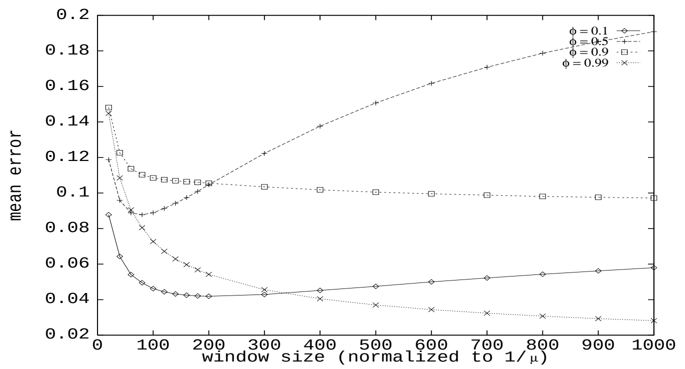

Figure 4: Mean error vs. window size, with $\frac{1}{\sigma_1} = 500 \times \frac{1}{\mu}$, $\rho_0 = 0.2$ and $\rho_1 = 0.7$, for different values of ϕ.

We let $\rho_0 = 0.2$ and $\rho_1 = 0.7$. Figure 3 shows the mean error vs. sliding window size for $\frac{1}{\sigma_1} = 5000 \times \frac{1}{\mu}$ and different values of ϕ. The window sizes are normalized with respect to the mean service time of packets on the link.

As we had expected, for a fixed value of ϕ the mean error is large for a small window size. The mean error reduces with increasing window size, reaches a minimum and starts increasing on further increasing the window size. This is because, when the window size is large, most of the time the window would straddle periods in which the arrival process is in different states. We see, from Figure 3, that choosing a value of about $700 \times \frac{1}{\mu}$ for W_s would give us a good performance for the entire range of ϕ.

Figure 4 shows the mean error vs. sliding window size for $\frac{1}{\sigma_1} = 500 \times \frac{1}{\mu}$ and different values of ϕ with $\rho_0 = 0.2$ and $\rho_1 = 0.7$. From this figure we see that choosing a value between $100 \times \frac{1}{\mu}$ and $150 \times \frac{1}{\mu}$ for W_s would yield good performance for the entire range of ϕ.

Thus we see that it is not possible to find a value for W_s that would do uniformly well for all values of $\frac{1}{\sigma_1}$. In the next section we propose and develop an adaptive window scheme that overcomes this problem.

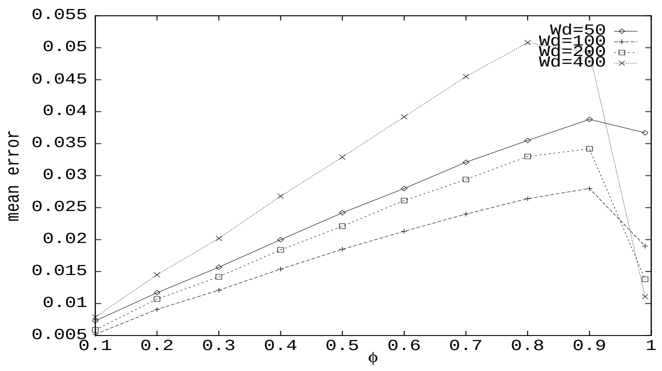

Figure 5: Mean error vs. ϕ, with $\frac{1}{\sigma_1} = 5000 \times \frac{1}{\mu}$, $\rho_0 = 0.1$ and $\rho_1 = 0.9$, for different values of W_d.

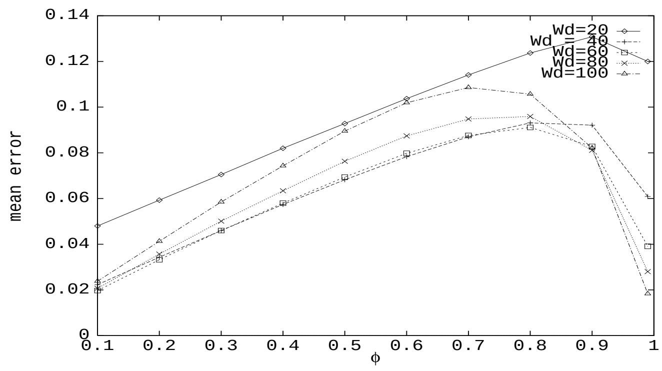

Figure 6: Mean error vs. ϕ, with $\frac{1}{\sigma_1} = 500 \times \frac{1}{\mu}$, $\rho_0 = 0.1$ and $\rho_1 = 0.9$, for different values of W_d;($\eta = 0.25$).

5 Adaptive Window Scheme

When the entire measurement window lies within the same state, we get a more accurate measurement for a larger window, and hence the window can be allowed to grow. Thus, instead of using a fixed size window for measurement, we use a window that varies in size adaptively. We define a minimum size for the measurement window, and start with a window of this minimum size. With time, we let the measurement window grow as long as we "guess" that the traffic is in the same state. When we "sense" that the traffic has moved to another state, we shrink this window to the minimum size and start all over again. We call this variable size window an "adaptive window", and represent it by W_a. To be able to adaptively vary the measurement window, we should be able to detect the epochs when the traffic moves from one state to another so that we can shrink W_a and let it grow again.

In the adaptive window scheme, to be able to detect the change in state, we use a second time window which we call the detection window, which we denote by W_d. W_d is a fixed size window and is basically a sliding window as described in the previous section. At any time t, we will represent the average occupancy over window W_a as $\hat{\rho}_{W_a}(t)$ and the average occupancy over window W_d as $\hat{\rho}_{W_d}(t)$. We use a detection mechanism which declares a change when the measurements over the two windows differ significantly, i.e., we declare a change when,

$$|\hat{\rho}_{W_a}(t) - \hat{\rho}_{W_d}(t)| > \eta$$

Where η is a threshold. It remains to choose appropriate values for W_d and η. We study the performance of this adaptive window scheme by using simulations. We also study the performance of the detection scheme and obtain good values of W_d and η.

5.1 Choosing W_d

We simulate the $MMPP/M/1$ queue to study the adaptive window scheme. Here we assume $\rho_0 = 0.1$ and $\rho_1 = 0.9$. We make measurements at discrete time intervals (every $5 \times \frac{1}{\mu}$) and obtain the mean measurement error as the average of these measurement errors. We use the expression given in Section 4.1 (with $\hat{\rho}(t_i) = \hat{\rho}_{W_a}(t_i)$) to obtain an estimate for the mean error.

Figure 5 shows the values for mean error plotted against different values of ϕ, with $\frac{1}{\sigma_1} = 5000 \times \frac{1}{\mu}$, $\rho_0 = 0.1$ and $\rho_1 = 0.9$. The different curves correspond to different values of W_d. We have assumed a value of 0.25 for the threshold η for the simulation. Window sizes are normalized with respect to mean service time.

From Figure 5, we observe that for small values of W_d the errors are high. As we increase the value of W_d the errors reduce. After a certain point, further increasing W_d results in increasing the errors. This behavior is expected. For small values of W_d, *probability of false alarm*, i.e., the probability of declaring a change in state when there is actually no change, is high. False alarms result in shrinking the adaptive window unnecessarily. This leads to larger errors. For large values of W_d, *the detection delay*, i.e., the time taken to detect the change after it occurs is high. This causes more errors because we make errors in our occupancy measurement before we detect the change.

Figure 6 shows the values for mean error plotted against different values of ϕ with $\frac{1}{\sigma_1} = 500 \times \frac{1}{\mu}$, $\rho_0 = 0.1$ and $\rho_1 = 0.9$. We see that the behavior is the same as before.

From Figures 5 and 6, we infer that a detection window W_d of size about $75 \times \frac{1}{\mu}$ will work well for both cases.

5.2 Choosing η

We analyze the performance of our detection mechanism using a simulation model. The model is a single server queue. At time 0 arrivals start with a rate λ_0. After a random time T, the state of the arrival process changes and the arrivals occur at a rate λ_1. We assume the arrivals to be Poisson, and the service times to be exponentially distributed with parameter μ. We use the detection window W_d and try to detect the change in state. We study the performance measures: probability of false alarm and detection delay. The window W_d slides in discrete steps of $5 \times \frac{1}{\mu}$. The change time T is a multiple of this time step, and is assumed to be distributed geometrically. Let S be the random variable denoting the time until a change is declared, i.e., $S = \inf\{t : |\hat{\rho}_{W_d}(t) - \rho_0| > \eta\}$ where $\rho_0 = \frac{\lambda_0}{\mu}$. The following performance measures are defined.

The probability of false alarm, P_f, is the probability of declaring a change in state when there is actually no change.

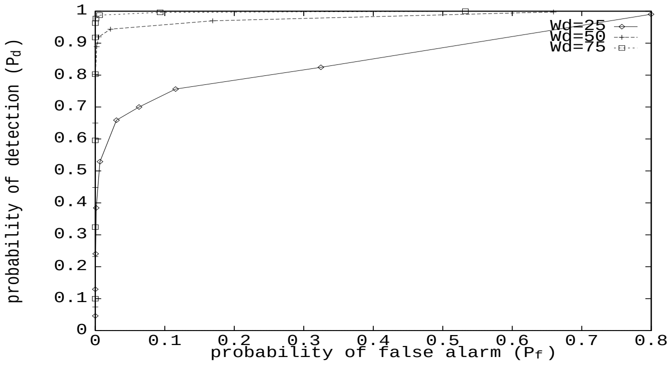

Figure 7: Probability of detection P_d vs. probability of false alarm P_f, for a transition from $\rho_0 = 0.1$ to $\rho_1 = 0.9$, with mean time to change $50 \times \frac{1}{\mu}$, for different values of W_d. η is decreasing to the right along each curve.

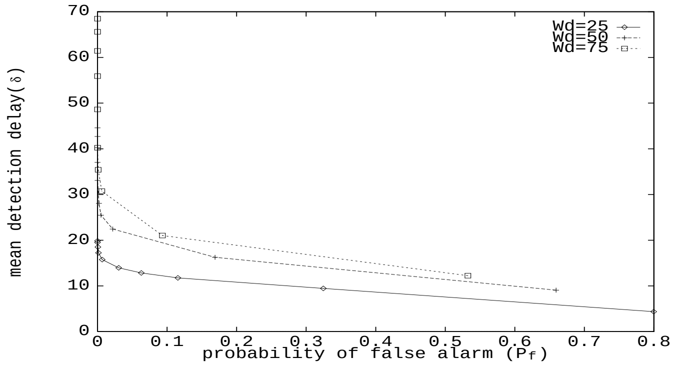

Figure 8: Mean detection delay δ vs. probability of false alarm P_f, for a transition from $\rho_0 = 0.1$ to $\rho_1 = 0.9$, with mean time to change $50 \times \frac{1}{\mu}$, for different values of W_d. η is decreasing to the right along each curve.

Thus,

$$P_f = P(S < T)$$

After the detection window moves into the changed state, it will eventually "detect" the change. We would like to detect the change at least before the detection window moves fully into the changed state. Thus "detection" is defined as the event of the change being detected before the detection window fully moves into the changed state. The probability of detection, P_d, is given by,

$$P_d = P(T \leq S \leq T + W_d)$$

When a change is detected, we would like to obtain the average time it takes to detect a change after it occurs. The mean detection delay, δ, is defined as the expected delay in detection given that the change is detected. It follows that,

$$\delta = E[S - T | T \leq S \leq T + W_d]$$

The conditioning $S \geq T$ is necessary, otherwise a large probability of false alarm can also lead to a small mean detection delay.

We choose a value for W_d, simulate the model for different values of η, and obtain the values of P_f, P_d and δ corresponding to the different values of η. Figure 7 shows P_d vs. P_f curves for the case when $\rho_0 = 0.1$ and $\rho_1 = 0.9$ with mean time before transition being $50 \times \frac{1}{\mu}$ units. The different curves correspond to different detection window sizes, and each point along a curve corresponds to a different η. Figure 8 shows the mean detection delay δ vs. probability of false alarm P_f, for different values of W_d. Table 1 shows the numerical values for $W_d = 75 \times \frac{1}{\mu}$. We observe from Table 1 that for threshold $\eta = 0.25$, the false alarm probability is 0.001, the probability of detection is 0.9772 and the mean detection delay is about $35 \times \frac{1}{\mu}$.

We have also obtained P_d vs. P_f curves, and mean detection delay δ vs. probability of false alarm P_f curves, for the case when $\rho_0 = 0.9$ and $\rho_1 = 0.1$, with the other parameters remaining the same as before. Owing to lack of space we do not show these curves here. We have observed from these results that, for $\rho_0 = 0.9$ and $\rho_1 = 0.1$, $W_d = 75\frac{1}{\mu}$ and $\eta = 0.25$ correspond to a false alarm probability of 0.053, detection probability 0.9784, and a mean detection delay of about $32 \times \frac{1}{\mu}$.

η	P_f	P_d	δ
0.05	0.5325	1.0000	12.2365
0.1	0.0933	0.9964	21.0220
0.2	0.0062	0.9876	30.7330
0.25	0.0010	0.9772	35.4215
0.3	0.0001	0.9629	40.2375
0.4	0.0000	0.9177	48.6295
0.5	0.0000	0.8029	55.9060

Table 1: Values for P_f, P_d and δ (in terms of the number of mean service times) obtained using the simulation model when $\rho_0 = 0.1$ and $\rho_1 = 0.9$ with $W_d = 75 \times \frac{1}{\mu}$ for different values of η.

From the above simulation results, we conclude that the detection window size $W_d = 75 \times \frac{1}{\mu}$ and threshold $\eta = 0.25$ yield a good performance for the detection scheme. We also found that the values chosen are good when the mean time before change is assumed to be $500 \times \frac{1}{\mu}$ and $5000 \times \frac{1}{\mu}$.

6 Experimental Comparison of the Schemes

The quantity $\hat{\rho}_W(t)$ can be estimated using SNMP variables. Define $(\hat{\rho}_{ij})_W(t)$ to be the occupancy estimate on the link between nodes i and j, in the direction $i \to j$. If we denote by $N_{ij}(t)$, the number of bytes transmitted on the link $i \to j$ in the interval $[0, t]$, we have,

$$(\hat{\rho}_{ij})_W(t) = \frac{N_{ij}(t) - N_{ij}(t - W)}{W \times C}$$

where C is the capacity of the link in *bytes/sec*.

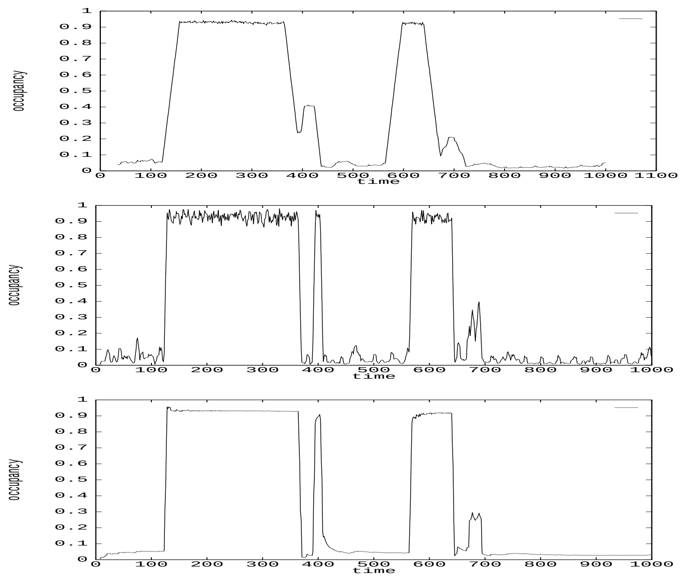

Figure 9: Link utilization as a function of time measured using the sliding window scheme with $W_s = 25$ seconds, $W_s = 5$ seconds, and using the adaptive window scheme with $W_d = 5$ seconds and threshold $\eta = 0.25$.

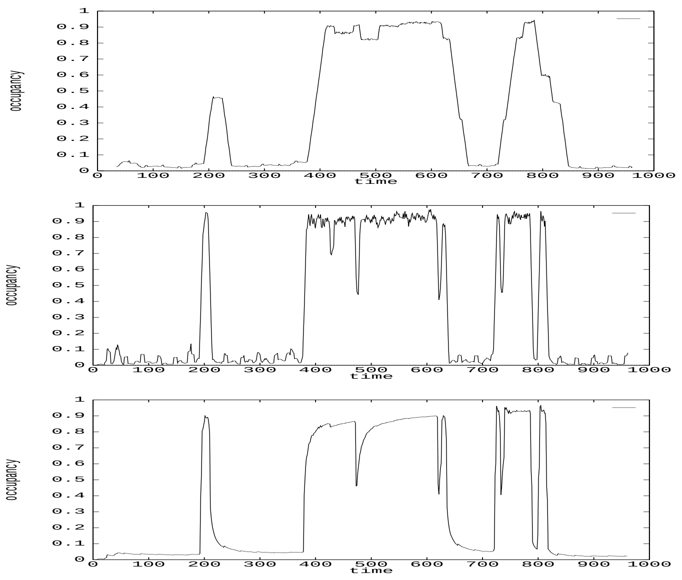

Figure 10: Link utilization as a function of time measured using the sliding window scheme with $W_s = 25$ seconds, $W_s = 5$ seconds, and using the adaptive window scheme with $W_d = 5$ seconds and threshold $\eta = 0.25$.

We monitor the utilization on a $64Kbps$ link connecting the ERNET[1] node at Indian Institute of Science, Bangalore, to the ERNET node in the Air India building in Bombay. Figure 9 shows one set of results. Three file transfers of sizes $1.7 Mbytes$, $98 Kbytes$, and $0.5 Mbytes$ approximately were carried out one after the other. The first file transfer started at around 125 seconds (after the start of the experiment) and finished at 365 seconds. The second one started at 395 seconds and finished at 405 seconds. The third one started at 565 seconds and finished at 640 seconds. The top part of the figure shows the utilization measured using the sliding window scheme with $W_s = 25$ seconds. We observe that we are not able to follow the changes in traffic closely. We see that the second file transfer goes almost undetected. The middle part of the figure shows the utilization monitored using a sliding window of size $W_s = 5$ seconds. Though we are able to detect the changes quickly, the measurement is quite noisy and we often make errors ranging from 10% to 20%. These errors will cause the performance measure (i.e., the mean error that we have defined) of this simple detection scheme to be quite poor. We then show the utilization measured using the adaptive window scheme with a detection window size $W_d = 5$ seconds and threshold $\eta = 0.25$. We observe that we are able to track the changes closely and measure the link utilization accurately.

Similar curves are presented in Figure 10. Three file transfers were carried out in succession. We see that the adaptive window scheme is the most effective; the dips in throughput during a file transfer are probably due to TCP dropping its window and going through a slow restart following a packet loss [2]

[1] ERNET (Education and Research NETwork) is India's academic and research computer network.

7 Conclusion

We have defined a notion of link utilization for a digital transmission link carrying time varying packet traffic. We have demonstrated the inadequacy of the simple sliding window scheme for estimating this utilization, and have developed an adaptive window scheme. The performance of this new scheme has been optimized on a simulation model, and we have shown that the scheme works well on "live" network traffic.

References

[1] H. Bruneel and B. G. Kim, "Discrete-Time Models for Communication Systems Including ATM", Kluwer Academic Press, 1993.

[2] Van Jacobson, "Congestion Avoidance and Control", *Proc. ACM SIGCOMM'88*, Aug. 1988.

[3] Marshall T. Rose, "Challenges in Network Management",*IEEE Network*, Nov. 1993.

[4] Marshall T. Rose, "The Simple Book".