

# Adaptive Bandwidth Provisioning Envelope based on Discrete Temporal Network Measurements

Balaji Krithikaivasan, Kaushik Deka<sup>1</sup> and Deep Medhi  
School of Computing and Engineering  
University of Missouri-Kansas City, Kansas City, MO-64110, USA

**Abstract**—We propose an adaptive bandwidth provisioning mechanism based on packet-level measurements (done at discrete time intervals) without any assumption on the statistical property of the traffic. Our approach consists of two stages: traffic forecasting, followed by bandwidth estimation. For the first stage, we have developed a Probability-Hop Forecasting algorithm (based on ARIMA model) to forecast traffic based on online traffic measurements in a temporal management information base framework. For the second stage, we present several bandwidth provisioning schemes that allocate or deallocate bandwidth depending on the traffic forecast generated by our forecasting algorithm. We have introduced an utility function (along with other measures) to evaluate the overall effectiveness of our bandwidth provisioning framework. Through the use of real-world data, we have found that our approach works well for highly fluctuating data.

## I. INTRODUCTION

It has been known that network traffic is dynamic in nature due to a variety of factors. An important network management issue is how to determine bandwidth needed in a network to maintain QoS guarantees in this dynamic environment. A common approach is to engineer the network for a given QoS by allocating bandwidth for the peak load over a time window (say, 24 hours). This approach has the drawback that during the non-peak time, the network bandwidth is under utilized. With the emergence of capabilities that can allow different bandwidth to be allocated at different time instant, identifying a good prediction method for bandwidth on demand becomes an increasingly important issue. With such schemes, unutilized bandwidth may be released for use by other services.

There is limited work on understanding network dynamism in defining control strategies to perform updates for capacity adjustment or dynamic bandwidth provisioning. Most such schemes in the literature are rooted on models that makes assumptions on the nature/dynamics of traffic. For instance, analyzing network performance under non-stationary traffic has been addressed for a queueing system with time-dependent Poisson traffic using a fluid-flow-based approach (see [1], [2], [3]). Groskinsky et. al [4] work investigates adaptive capacity control schemes in such a dynamic environment using a time-dependent fluid-flow model. A summary of different bandwidth access control schemes for heterogeneous traffic can be found in [3]. Some of these schemes have coarser granularity in that they are only-aimed at connection-level performance. In Afek et. al. [5] work, for a time varying traffic inside a ATM virtual circuit, hidden markov model based

bandwidth estimation for future time interval is proposed. In their work, bandwidth is loosely defined in terms of number of ethernet frames that could be served by a server in a unit time. Also, in the context of ATM virtual path management, there have been some work [6], [7], [8] on dynamically adjusting the size of a virtual path by controlling the number of active virtual circuits inside the virtual path. These works assume the virtual circuit connections to follow a Poisson arrival process with exponential holding time. Moreover, they consider the connections to be of equal bandwidth. In recent years, neural network based approach to bandwidth prediction has received some attention [9], [10], [11]. However, none of these works consider the predictive dynamic adjustment of available bandwidth for generically measured traffic data.

In this work, we attempt to gain a deeper insight into network dynamism from the finer granularity of traffic, namely from measurements at the packet level, and address the issue of adaptive bandwidth provisioning from this nodal perspective. Our assumption is that the packet level measurement is done in discrete time intervals since in most realistic networking environment, measurement on the activity cannot be done continuously. It may be noted here that actual measurements from data traffic found Poisson model to be not applicable at the packet level [12]. Further, there are many recent works in the literature that has postulated that packet level network traffic has self-similar properties and shows long range dependencies [13]. Our interest here is to look at some way to predict bandwidth (*not* purely do the traffic forecast) without making any assumption about the traffic.

Nevertheless, the measured information for Internet traffic on a link on a periodic basis (say every five minutes or one minute) is found to be quite chaotic. In fact, in most forecasting environment with non-stationary data, statistical techniques such as the Autoregressive Integrated Moving Average (ARIMA) model seems to work quite well [14], [15]. Since, our interest here is to provision a bandwidth envelope for future traffic, we have found that ARIMA traffic forecasting model can be modified to suit our need. In essence, our provisioning goal has led us to a two-stage approach where we developed a probability-hop forecast algorithm from an ARIMA model, followed by bandwidth provisioning techniques.

It may be noted that going from discrete-time traffic measurements to bandwidth prediction, we do follow a simple principle in deriving the estimation method: be liberal with

<sup>1</sup>Currently affiliated with Demos Solutions, Norwell, MA-02061, USA.

allocation, and be conservative with deallocation. Further, to assess our approach, we consider the average utilization measure and average data loss and show how effective are the proposed schemes. However, we do not address the issue of minimizing the signaling cost in this paper. Nevertheless, it is an important goal that affects how and when bandwidth provisioning is needed to be done.

To assess our methodology, we have collected incoming packet data traffic from the link that connects the University of Missouri-Kansas City to MOREnet for connectivity to the rest of the Internet. While this temporal measurement is collected every five minutes and our analysis is done for such data sets, our approach can work for other time window-based data as well. We have specifically considered three disparate data sets to see how far we can generalize the results.

The rest of the paper is organized as follows. In Section II, we present the overview of ARIMA modeling of time series along with the identification of ARIMA models for various data sets used in our study. We also discuss here about the exact ARIMA forecasting. In Section III, we explain our probability-hop forecasting algorithm to predict data traffic for future time instants. We also present some results on the effectiveness of probability-hop forecast algorithm in Section III-D. Next, we move on to describe several adaptive bandwidth provisioning schemes in Section IV. Finally, in Section V, we present experimental results for various data sets followed by our summary and conclusion.

## II. ARIMA MODELING AND FORECASTING

### A. Overview and Identification

The data rate at a particular interface is a time-varying process due to the combination of a number of factors like routing, application throughput, management information flow etc. The data rate can be assumed to be a realization of the underlying network dynamics which is stochastic in nature. When sampled at fixed periods,  $N$  such observations yield a temporal time series which is non-stationary, as having no natural mean. Formally, such an equally spaced time series for a stochastic process  $z_t$  can be described by a  $N$  dimensional random variable  $(z_1, z_2, \dots, z_N)$  with the joint probability distribution denoted by  $p(z_1, z_2, \dots, z_N)$ .

AutoRegressive Integrated Moving Average (ARIMA) models are proved to be quite powerful to model a class of non-stationary data otherwise called homogeneous non-stationary [16]. In a way, ARIMA process can be thought of as a device that transforms a highly dependent and homogeneous non-stationary process  $z_t$  into a sequence of uncorrelated random variables  $a_t$ , i.e., in transforming the process into a white noise. The general form of the ARIMA( $p, d, q$ ) model is :

$$\Phi(B)\nabla^d z_t = \theta(B)a_t \quad (1)$$

where  $\Phi(B) = 1 - \Phi_1 B - \dots - \Phi_p B^p$  is the autoregressive operator of order  $p$ ,  $\theta(B) = 1 - \theta_1 B - \dots - \theta_q B^q$  is the moving average operator of order  $q$ ,  $\nabla^d = (1 - B)^d$  is the  $d^{th}$  order difference operator, and  $B$  is the backward-shift operator,

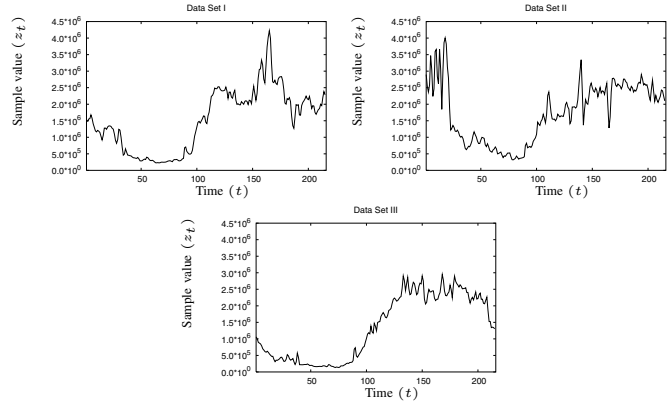


Fig. 1. Data sets

i.e.,  $Bz_t = z_{t-1}$ . Also,  $\varphi(B) = \Phi(B)\nabla^d$  is the stationary autoregressive operator. In an explicit manner, equation (1) can be specified as follows:

$$z_t = \varphi_1 z_{t-1} + \varphi_2 z_{t-2} + \dots + \varphi_{p+d} z_{t-p-d} + a_t - \theta_1 a_{t-1} - \theta_2 a_{t-2} - \dots - \theta_q a_{t-q}. \quad (2)$$

We consider three different data sets in this work shown in Fig. 1 and adopted an iterative approach [16] toward ARIMA model building. We used the GNU R statistical package [17] to do our data analysis. Each data set consists of 18-hour data considering the 5-minute interval. We used the first 96 samples (8-hour data) to do the time series model fitting for all the data sets. The time series behaved as though there were no fixed mean implying the non-stationary nature of the data. Under the homogeneous non-stationarity assumption, we obtained the first order differencing of actual time series  $z_t$  to generate a new time series  $w_t$  where  $w_t = z_t - z_{t-1}$ . This new time series  $w_t$  was found to be stationary in nature in the case of all the three data sets. Therefore, we fixed  $d = 1$  in our ARIMA model fitting. In order to identify the order of autoregressive as well as the moving average components, we examined the autocorrelation function (ACF) plot as well as the partial autocorrelation function (PACF) plot. The ACF plots indicated the order  $q$  of moving average component to be 1 for all the data sets. However, the PACF plot suggested two possible orders for  $p$ , i.e.,  $p = 1$  and  $p = 2$ . As a result, we evaluated both the ARIMA( $p = 1, d = 1, q = 1$ ) and ARIMA( $p = 2, d = 1, q = 1$ ) models for all the data sets. The autoregressive as well as moving average parameter estimates were computed using a maximum-likelihood approach in either case. In order to identify a better fit among these two models, we examined the ACF of residuals and Box and Jenkins' "portmanteau goodness of fit" statistic [16]. These diagnostic tests revealed ARIMA(2,1,1) to be a better fit than ARIMA(1,1,1) for all the data sets. Due to space constraints, we have shown the results of such diagnostic tests for only data set I in Fig. 2. Thus, we have  $p = 2, q = 1$  and  $d = 1$  for all the data sets.

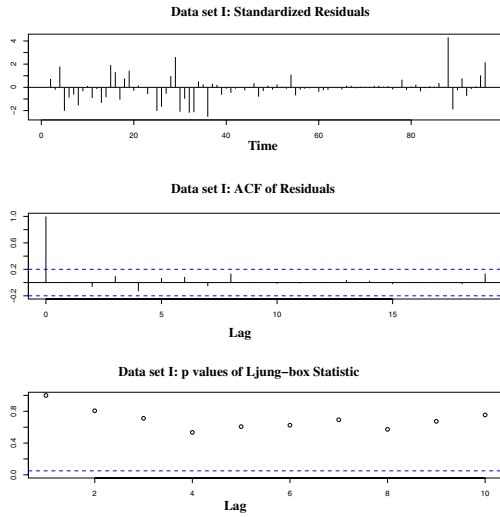


Fig. 2. Data set I: Diagnostic Test

### B. Exact ARIMA Forecast

The utility of identifying and fitting an ARIMA model to a time series data lies in its ability to forecast the future values of time series reasonably close to the actual time series. We use the difference equation approach to forecast for any lead time  $l > 0$  while standing at time  $t$ . Let  $\hat{z}_t(l)$  denote the minimum mean square error forecast at time  $t$  for a lead time  $l$ . It has been shown in [16] that  $\hat{z}_t(l)$  is given by the conditional expectation  $E[z_{t+l}|z_t, z_{t-1}, \dots]$  which we denote here by  $[z_{t+l}]$ . Therefore, from (2) we have

$$\hat{z}_t(l) = \varphi_1[z_{t+l-1}] + \dots + \varphi_{p+d}[z_{t+l-p-d}] + [a_{t+l}] - \theta_1[a_{t+l-1}] - \dots - \theta_q[a_{t+l-q}]. \quad (3)$$

The above difference equation is evaluated by inserting actual  $z$ 's when known, forecasted  $z$ 's for future values, actual  $a$ 's when known, and zeros for future  $a$ 's. The forecasting process is initiated by approximating unknown  $a$ 's by zeros. Once a new observation is available, the residual  $a_{t+1}$  is given by the one-step ahead forecast errors (see [16] for details), i.e.,

$$a_{t+1} = z_{t+1} - \hat{z}_t(1) \quad (4)$$

which can be used to compute the new one-step ahead forecast  $\hat{z}_{t+1}(1)$ . Here, we adopt a recursive approach to update the forecasts at successive time instants by using  $\Psi$  weights as shown in (5).

$$\hat{z}_{t+1}(l) = \hat{z}_t(l+1) + \Psi_l a_{t+1}. \quad (5)$$

These  $\Psi$  weights are estimated as follows [16]:

$$\begin{aligned} \Psi_0 &= 1 \\ \Psi_j &= \varphi_1 \Psi_{j-1} + \dots + \varphi_{p+d} \Psi_{j-p-d} - \theta_j, \quad j > 1 \end{aligned} \quad (6)$$

where,  $\theta_j = 0$  for  $j > q$  and  $\varphi(B)$  is the stationary autoregressive operator. Our initial assessment with the ARIMA model on the data sets indicated that the exact forecasts generated gives a fixed weight to past history and has a lag in capturing

transient network dynamics. Such an exponential averaging scheme is therefore not fully capable of extrapolating temporal time series which is often chaotic in nature. This motivated us to introduce a probability limit-based approach based on the exact forecast from the ARIMA model. Further, our interest is not just to forecast traffic; rather, to estimate the bandwidth needed for *future traffic*.

### III. PROBABILITY-HOP FORECASTING

Assuming random shocks (residuals)  $a$ 's to be Normally distributed as suggested in [16], it follows that given information up to time  $t$ , the conditional probability distribution  $p(z_{t+l}|z_t, z_{t-1}, \dots)$  of a future value  $z_{t+l}$  of the process will be Normal with mean  $\hat{z}_t(l)$  and standard deviation  $(1 + \sum_{j=1}^{l-1} \Psi_j^2)^{\frac{1}{2}} \sigma_a$ . Here,  $\sigma_a$  denotes the standard deviation of white noise process  $a_t$  which we replace by the maximum likelihood estimate  $s_a$  in our computation.

For a desired level of probability  $\epsilon$  and a lead time  $l$ , the probability limit of forecasts is given by

$$z_{t+l}(\pm) = \hat{z}_t(l) \pm u_{\frac{\epsilon}{2}} \left(1 + \sum_{j=1}^{l-1} \Psi_j^2\right)^{\frac{1}{2}} \sigma_a \quad (7)$$

where  $u_{\frac{\epsilon}{2}}$  is the deviate exceeded by a proportion  $\frac{\epsilon}{2}$  of the standard Normal distribution  $N(0, 1)$ .

The probability-hop forecast algorithm uses three curves - Upper Probability Curve (UPC), Exact Forecast Curve (EFC) and Lower Probability Curve (LPC) to sensitize the forecast mechanism. The algorithm has been designed keeping in mind the final goal of an effective bandwidth provisioning (described later on). The exact forecasts based on the ARIMA model constitute the EFC. UPC follows the upper probability limits of the forecasts, while LPC follows the lower probability limits of the forecasts.

It is frequently the case that the forecasts are needed for multiple lead times, say  $l = 1, 2, \dots, L$  for some  $L > 0$ . However, in this work, we are more interested in forecasting for only one lead time ( $l = 1$ ). In other words, we invoke our forecast algorithm at every instant to make use of the new observation as soon as it is available. With this simplification, (7) reduces to

$$\begin{aligned} \text{EFC} &= \hat{z}_t(1) \\ \text{UPC} &= \hat{z}_t(1) + u_{\frac{\epsilon}{2}} \sigma_a \\ \text{LPC} &= \hat{z}_t(1) - u_{\frac{\epsilon}{2}} \sigma_a \end{aligned} \quad (8)$$

We now illustrate the traffic forecast behavior for the data set I. Fig. 3 shows the EFC, UPC and LPC for 50% probability limit forecasts (along with the actual measurement) starting from time instant  $t_{95}$ . For the clarity of presentation, we split the 10 hours of data ( $t_{95}$  to  $t_{215}$ ) into four intervals: (95, 120), (121, 150), (151, 175) and (176, 215) in Fig. 3. It is seen here that the EFC follows the time series quite narrowly however, often with a lag time. In particular, whenever there is a sudden rise or drop in the time series,  $z_t$  falls outside the band. Our probability-hop forecast algorithm helps in these undesirable instants (spikes) by shifting the forecast toward the UPC. In this way, the magnitude of forecast errors can be reduced.

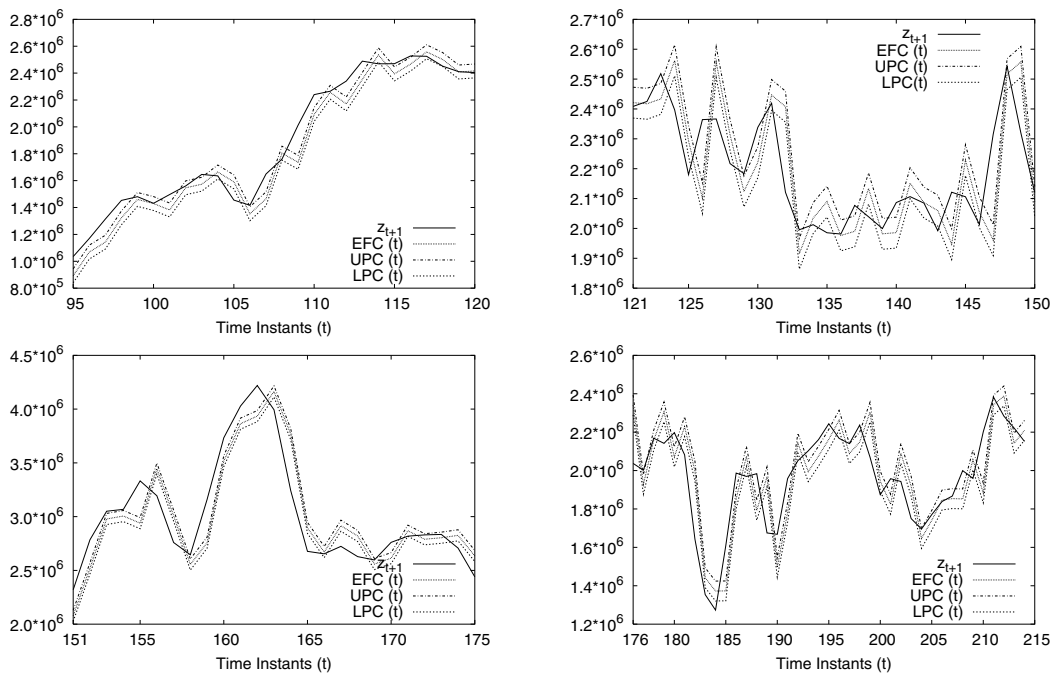


Fig. 3. Probability-hop forecast with 50% limits along with exact ARIMA forecast (Data set I)

#### A. Algorithm

The intuition behind our algorithm is to increase the sensitivity of the effective forecast curve by hopping between the UPC, EFC and LPC at each forecast instant. In other words, our effective forecast value, referred to as the probability-hop forecast, falls on one of these three curves based on what we refer to as “sensitivity criterion”. The sensitivity criterion is defined as the gradient of the weighted sum of two most recent probability-hop forecast errors (difference between the actual time series value and the probability-hop forecast value) with respect to “sensitivity quantum”. The sensitivity quantum represents a fraction of maximum available bandwidth. Also, the idea behind considering weighted sum of errors of last two time instants is to capture the sudden spikes and declines that cannot be effectively captured by classical ARIMA smoothing.

The forecast sensitivity criterion is designed to address the following considerations effectively:

- 1) To have a low mean square error of the forecasts - as close to the theoretical Minimum Mean Square Error (MMSE) achieved by taking conditional expectation of the ARIMA equation at time  $t$  for lead time 1, namely  $E[z_{t+1}]$ .
- 2) To minimize the negative forecast errors or in other words to minimize the number of under-forecasted values.

The first design consideration above is adhered to by making the probability-hop forecast curve (PFC) conform to the exact forecast curve in regions of stable local variance. Such areas are not punctuated by high system perturbations and the EFC gives a good, close fit to the time series. In addition, we hop on to the LPC whenever the system perturbations show an abrupt

decline. The second consideration is addressed by allowing the PFC to hop on to the UPC whenever the system perturbations indicate a sudden increase.

In order to be more flexible with the level of sensitivities to a sharp rise or a fall in the time series, we allowed the sensitivity quantum to be different in each direction. The sensitivity quantum that controls the sensitivity of forecast curve toward UPC is referred as *sensitivity up quantum* while the one that controls the sensitivity of forecast curve toward LPC is referred as *sensitivity down quantum*. A positive weighted forecast error is measured against the sensitivity up quantum whereas a negative weighted forecast error is measured against the sensitivity down quantum. In order to decide whether the sensitivity criterion is significant enough for a shift in either direction, we used 1.0 as the threshold. This means, if the absolute value of sensitivity criterion is greater than 1.0, a shift is warranted. We would point out here that the effect of choosing a higher (lower) significance factor can be achieved by increasing (decreasing) the sensitivity up/down quantum.

For brevity, we use the following notations in order to present the probability-hop forecast algorithm.

$\Delta S_u$	Sensitivity Up Quantum
$\Delta S_d$	Sensitivity Down Quantum
$e_t$	One-step ahead probability-hop forecast error at time $t$
$w_t$	Weight associated with probability-hop forecast error at time $t$
$\varepsilon$	Weighted sum of two recent probability-hop forecast errors
$PF_t$	Probability-hop forecast for time $t$

The input parameters to the algorithm are sensitivity up

quantum and sensitivity down quantum ( $\Delta\mathcal{S}_u$  and  $\Delta\mathcal{S}_d$ ), weights associated with the forecast errors for last two time instants ( $w_t$  and  $w_{t-1}$ ) and the bounds on the probability-hop forecast value. See Algorithm (1) for formal description of probability-hop forecast. Next, we present guidelines for choosing a particular value for the parameters involved in our algorithm.

### B. Choosing $\Delta\mathcal{S}_u$ and $\Delta\mathcal{S}_d$

Recall here that both  $\Delta\mathcal{S}_u$  and  $\Delta\mathcal{S}_d$  represent a fraction of the maximum available bandwidth except that we associate a direction with it. It follows from the definition that as we increase  $\Delta\mathcal{S}_u$  ( $\Delta\mathcal{S}_d$ ), the sensitivity of the PFC toward UPC (LPC) decreases. Consequently, the PFC will follow the EFC closely. This means that the mean square error of forecasts will tend toward the theoretical minimum and the misses or number of under-forecasted values will increase. On the other hand, as we decrease  $\Delta\mathcal{S}_u$  ( $\Delta\mathcal{S}_d$ ), the PFC becomes highly sensitive to transient network dynamics thereby capturing most of the sudden “spikes” (“drops”). As a result, the PFC will shift frequently to UPC as well as to LPC which increases the mean square error of forecasts. So a match between a low mean square error and a desired level of sensitivity is sought in the selection of the sensitivity quantum in both directions.

---

#### Algorithm 1 PHForecast( $\Delta\mathcal{S}_u, \Delta\mathcal{S}_d, w_t, w_{t-1}, \text{UB}, \text{LB}$ )

---

```

PFt-1 ←  $\hat{z}_{t-2}(1)$ 
PFt ←  $\hat{z}_{t-1}(1)$ 
while true do
  et-1 ← zt-1 - PFt-1
  et ← zt - PFt
  ε ← et-1wt-1 + etwt
  EFC ←  $\hat{z}_t(1)$ 
  UPC ←  $\hat{z}_t(1) + u_{\frac{\epsilon}{2}}\sigma_a$ 
  LPC ←  $\hat{z}_t(1) - u_{\frac{\epsilon}{2}}\sigma_a$ 
  if PFt = ( $\hat{z}_{t-1}(1) - u_{\frac{\epsilon}{2}}\sigma_a$ ) then
    PFt+1 ← EFC
  else if  $\frac{\epsilon}{\Delta\mathcal{S}_u} > 1.0$  and UPC < UB then
    PFt+1 ← UPC
  else if  $\frac{\epsilon}{\Delta\mathcal{S}_d} < -1.0$  then
    if PFt =  $\hat{z}_{t-1}(1)$  and LPC ≥ LB then
      PFt+1 ← LPC
    else
      PFt+1 ← EFC
    end if
  else
    if PFt =  $\hat{z}_{t-1}(1)$  then
      PFt+1 ← EFC
    else
      PFt+1 ← UPC
    end if
  end if
  t ← t + 1
end while

```

---

Essentially,  $\Delta\mathcal{S}_u$  and  $\Delta\mathcal{S}_d$  plays a dominant role in controlling the PFC thereby, the “bandwidth envelope”.

### C. Choosing forecast error weights ( $w_t, w_{t-1}$ )

The underlying idea to introduce these weights is to do some level of smoothing of two most recent forecast errors. From the definition of sensitivity criterion (refer III-A), it is evident that the choice of weights play a crucial role in controlling the significance of the probability-hop forecast errors. In our work, we fixed the sum of the weights at 1.0 (i.e.,  $w_{t-1} + w_t = 1$ ,  $0 \leq w_t, w_{t-1} \leq 1$ ). Nevertheless, such a choice still provides us with a lot of possibilities for these weights to experiment. Then, when we tried couple of different combinations here, we found the choice  $w_{t-1} = w_t = 0.5$  performed reasonably well. Thus, we kept these weights fixed at 0.5 in our experiments.

### D. Algorithm Evaluation

In this section, we evaluate our probability-hop forecasting mechanism with the exact ARIMA forecasting by enumerating the mean square forecast error deviation and the fraction of under-forecast instants (i.e., fraction of time instants ( $t$ ) where  $\text{PF}_t < z_t$ ). The maximum available bandwidth of the link that connects our university to MOREnet is 45 Mbps. Thus, the values of  $\Delta\mathcal{S}_u$  and  $\Delta\mathcal{S}_d$  represents a fraction of 45 Mbps here. The various choices that we considered here for  $\Delta\mathcal{S}_u$  and  $\Delta\mathcal{S}_d$  are 2%, 4%, 5%, 8%, 10%. However, we are more interested in the ( $\Delta\mathcal{S}_u, \Delta\mathcal{S}_d$ ) pairs with  $\Delta\mathcal{S}_d \geq \Delta\mathcal{S}_u$ . This is because, having  $\Delta\mathcal{S}_d$  lower than  $\Delta\mathcal{S}_u$  will force the probability-hop forecast to fall onto LPC more frequently as compared to hopping on UPC. Such a behavior has the direct implication of liberal bandwidth deallocation from the provisioning module leading to increased loss ratio.

Here, we evaluated both 50% and 90% probability limits. The results are presented for all the three data sets along with that of exact ARIMA forecast (denoted by EFC) in Table I where “ED” stands for mean square error deviation of probability-hop forecast and “UFI” stands for the fraction of under-forecast instances.

It is evident from Table I that we see a good improvement in UFI with 50% limit forecast as compared to exact ARIMA forecast (see EFC in Table I). The downside is the increase in the forecast error deviation. In data set I and III, we observe further improvement in UFI with 90% limit forecasts. However, in data set II, 90% limit shows a degrading behavior as compared to 50% limit forecasts. Then, increasing  $\Delta\mathcal{S}_d$  for a given  $\Delta\mathcal{S}_u$  also brings down UFI while the ED shows an oscillatory behavior. The improvement in UFI here is due to the decreased sensitivity toward LPC that helps to cut down forecast misses. As we increase  $\Delta\mathcal{S}_u$  along with  $\Delta\mathcal{S}_d$ , the UFI and ED shows convergence toward the observed quantities for EFC. In fact, for the pairs (8%,8%), (8%,10%) and (10%,10%), EDs and UFIs display very close convergence to that of EFC. This possibly indicates that the probability-hop forecast curve follows the exact forecast curve almost all the time. Therefore, we avoid these choices in our evaluation of bandwidth provisioning schemes. In addition, it is clear

TABLE I  
PROBABILITY HOP FORECAST RESULTS

EFC	Data Set I				Data Set II				Data Set III			
	ED <sup>†</sup> = 1.7461		UFI = 57.50%		ED <sup>†</sup> = 2.2134		UFI = 55.00%		ED <sup>†</sup> = 1.5294		UFI = 56.67%	
	50% Limits		90% Limits		50% Limits		90% Limits		50% Limits		90% Limits	
$\Delta S_u, \Delta S_d$	ED <sup>†</sup>	UFI	ED <sup>†</sup>	UFI	ED <sup>†</sup>	UFI	ED <sup>†</sup>	UFI	ED <sup>†</sup>	UFI	ED <sup>†</sup>	UFI
(2%,2%)	1.8034	51.67%	1.9818	51.67%	2.3430	50.00%	3.4863	50.83%	1.5540	48.33%	1.6995	49.17%
(2%,4%)	1.7562	46.67%	1.9154	40.83%	2.5196	44.17%	3.4939	47.50%	1.5667	42.50%	1.7170	36.67%
(2%,5%)	1.7551	46.67%	1.9608	37.50%	2.6430	38.33%	3.2685	44.17%	1.5702	42.50%	1.7948	26.67%
(2%,8%)	1.7965	42.50%	1.9739	28.33%	2.9214	17.50%	3.8172	35.83%	1.5873	41.67%	1.8458	25.00%
(2%,10%)	1.7965	42.50%	2.1361	24.17%	2.9372	17.50%	4.6599	20.83%	1.5873	41.67%	1.8458	25.00%
(4%,4%)	1.7553	58.33%	1.8013	57.50%	2.2989	50.83%	2.6599	50.83%	1.5997	50.83%	1.5831	57.50%
(4%,5%)	1.7541	58.33%	1.7936	56.67%	2.3719	49.17%	2.5439	52.50%	1.6031	50.83%	1.7976	40.00%
(4%,8%)	1.8053	50.83%	1.7647	55.00%	2.7918	27.50%	2.7437	51.67%	1.6146	47.50%	1.8552	35.00%
(4%,10%)	1.8053	50.83%	2.0278	42.50%	2.7814	27.50%	3.6275	38.33%	1.6146	47.50%	1.8552	35.00%
(5%,5%)	1.7567	58.33%	1.7844	58.33%	2.4140	49.17%	2.5946	54.17%	1.5281	56.67%	1.5282	57.50%
(5%,8%)	1.8204	50.83%	1.7812	57.50%	2.7918	27.50%	2.7604	52.50%	1.5294	56.67%	1.5294	56.67%
(5%,10%)	1.8204	50.83%	2.0422	45.00%	2.8023	27.50%	3.6632	40.00%	1.5294	56.67%	1.5294	56.67%
(8%,8%)	1.7461	57.50%	1.7461	57.50%	2.2433	55.00%	2.4158	52.50%	1.5294	56.67%	1.5294	56.67%
(8%,10%)	1.7461	57.50%	1.7461	57.50%	2.2304	55.00%	2.2790	55.00%	1.5294	56.67%	1.5294	56.67%
(10%,10%)	1.7461	57.50%	1.7461	57.50%	2.2304	55.00%	2.2790	55.00%	1.5294	56.67%	1.5294	56.67%

<sup>†</sup> Reported in terms of ( $\times 10^6$ )

from Table I that choosing 50% probability limits as against 90% probability limits adheres more closely to our two major considerations on the effectiveness of a forecast mechanism (i.e., to strike a balance between low mean square error forecasts and the number of under-forecasts). So, we use 50% probability limits in all of our experiments.

#### IV. BANDWIDTH PROVISIONING SCHEMES

A predictive bandwidth provisioning scheme provisions bandwidth for a future time instant based on a forecasted bandwidth value (as we have done here) as well as on desired performance metrics reflecting QoS requirements of the traffic class. In this section, we propose several bandwidth provisioning schemes which take as input the probability-hop forecast generated by our *PHForecast*(.) algorithm (see Algorithm (1)). The underlying assumption of these provisioning schemes is that the bandwidth can be allocated or deallocated only in quanta. We refer to this quantum as “bandwidth quantum”. It is expressed as a fraction of maximum available bandwidth very similar to  $\Delta S_u$  and  $\Delta S_d$ .

The way the probability-hop forecast value maps onto the “bandwidth requirement” at the next time instant is as follows. Let  $\Delta B$  represents the bandwidth quantum and  $C_{max}$  represents the maximum available bandwidth. Then, we determine the interval  $[k\Delta B, (k+1)\Delta B]$  ( $k \geq 0$ ) in which the probability-hop forecast value ( $PF_{t+1}$ ) falls and use the upper bound of the interval ( $(k+1)\Delta B$ ) or the maximum available bandwidth ( $C_{max}$ ) if  $C_{max} < (k+1)\Delta B$  as the bandwidth requirement at time  $t+1$ . In other words, if  $BW_{t+1}$  represents the bandwidth requirement at time  $t+1$  based on  $PF_{t+1}$ , then

we have

$$BW_{t+1} = \min \left\{ \left\lceil \frac{PF_{t+1}}{\Delta B} \right\rceil \times \Delta B, C_{max} \right\}. \quad (9)$$

The bandwidth requirement was thus greater than or equal to the probability-hop forecast value. This has the desirable effect of making the bandwidth provisioned less sensitive to small fluctuations in the forecast within the bounds of size  $\Delta B$ . This is because all the forecast values that fall within the same  $(k\Delta B, (k+1)\Delta B)$  map on to the same bandwidth value. The implicit smoothing of forecasts by the bandwidth provisioning module also has an important byproduct. Every time, a forecast is made, there is a possibility that bandwidth might be newly allocated or deallocated; in other words, bandwidth might be re-provisioned. This re-provisioning operation has a significant cost associated with it and often referred to as the “signaling cost”. The smoothing byproduct of the provisioning module avoids frequent re-provisioning which leads to an overall reduced signaling cost. Therefore,  $\Delta B$  has direct implications on the signaling overhead.

##### A. Instantaneous Bandwidth Provisioning (*IBP*)

This scheme is an utopian scheme where the bandwidth is re-provisioned at every forecast instant unless the probability-hop forecast for next time instant maps onto the currently provisioned bandwidth. Therefore, it is highly sensitive to the forecasts generated. In fact, such a scheme is useful in terms of identifying the lower bound on the average data loss as well as the upper bound on the signaling overhead. Given the probability-hop forecast value  $PF_{t+1}$  at time  $t$ , the instantaneous bandwidth requirement for the next time instant  $t+1$  denoted by  $IBP_{t+1}$  will be equal to the  $BW_{t+1}$  given by (9).

If the probability-hop forecast at two successive instants, say  $t$  and  $t + 1$  maps onto same bandwidth requirement, we will have  $\mathcal{IBP}_{t+1} = \mathcal{IBP}_t$  and hence, there will be no need of re-provisioning at time  $t$ .

### B. Stabilized Bandwidth Provisioning (SBP)

The major disadvantage of the instantaneous provisioning scheme is the considerable amount of signaling overhead associated with it. Thus, it is desirable to wait for a preset time period, say  $\hat{t}$ , before any other bandwidth adjustments are made since the last adjustment and in particular, bandwidth deallocation. This is the underlying idea of the *SBP* scheme. We use a Hold Down Timer (HDT) to delay the deallocation and it is reset after every time period  $\hat{t}$ . So, at any time instant  $t$ , bandwidth will be re-provisioned either if  $\mathcal{BW}_{t+1}$  exceeds the currently provisioned bandwidth or if HDT expires whichever comes first. But, whenever bandwidth is re-provisioned, the timer is reset. If  $t_l$  is the last instant when bandwidth was re-provisioned then,

$$\mathcal{SBP}_{t+1} = \begin{cases} \max\{\mathcal{IBP}_{t+1}, \mathcal{SBP}_t\} & t - t_l < \hat{t} \\ \mathcal{IBP}_{t+1} & t - t_l = \hat{t} \end{cases} \quad (10)$$

where  $t_l$  will be reset to  $t$  if  $\mathcal{SBP}_{t+1}$  is assigned the value of  $\mathcal{IBP}_{t+1}$ . If  $\mathcal{SBP}_{t+1} = \mathcal{SBP}_t$  after HDT is reset then, there will be no need of re-provisioning at time  $t$ .

### C. Stabilized Bandwidth Provisioning with Local Maxima (SBPL)

The motivation behind this variation of the *SBP* scheme is to follow a more conservative approach while deallocating bandwidth at the end of the hold down timer period. Instead of re-provisioning bandwidth based purely on the instantaneous requirement immediately after the HDT is reset, we also consider the local maximum of the bandwidth requirement in the last timer period while deallocating bandwidth. In other words, if the bandwidth requirement for the next HDT period (estimated at  $t$ ) is greater than the current bandwidth provisioned then, provisioning is done based on  $\mathcal{IBP}_{t+1}$ . Otherwise, the current provisioned bandwidth is deallocated upto the higher of the  $\mathcal{IBP}_{t+1}$  and the local maxima in the last hold down timer period. If  $\mathcal{BW}^{max}$  denotes the local maxima in the last HDT period then,

$$\mathcal{SBPL}_{t+1} = \begin{cases} \max\{\mathcal{IBP}_{t+1}, \mathcal{SBPL}_t\} & t - t_l < \hat{t} \\ \max\{\mathcal{IBP}_{t+1}, \mathcal{BW}^{max}\} & t - t_l = \hat{t} \end{cases} \quad (11)$$

where  $t_l$  will be reset to  $t$  if  $\mathcal{SBPL}_{t+1}$  is assigned the value of  $\mathcal{IBP}_{t+1}$  in the first case. It will be always reset in the second case.

## V. EXPERIMENTAL RESULTS

To gain insights from our experiments with various data sets, we first state the suitable performance metrics followed by our evaluation and general observations.

### A. Performance Metrics

The bandwidth provisioning schemes discussed above can be evaluated under the umbrella of the following performance metrics:

- 1) *Average Utilization ( $\mathcal{U}_{avg}$ )*: The average utilization is computed as the average of the ratio of the bandwidth utilized (bandwidth provisioned less current data rate) to the bandwidth provisioned. If  $z_t$  represents the data rate and  $\mathcal{BP}_t$  represents the provisioned bandwidth at time  $t$ , the average utilization is computed as follows:

$$\mathcal{U}_{avg} = \frac{1}{T} \sum_{t=1}^T \max\left\{\frac{\mathcal{BP}_t - z_t}{\mathcal{BP}_t}, 1.0\right\} \quad (12)$$

- 2) *Average Loss Ratio ( $\mathcal{LR}_{avg}$ )*: The loss ratio gives a measure of the bytes dropped at the interface due to under-provisioning of bandwidth. With reference to the above notations, the average loss ratio is computed as follows:

$$\mathcal{LR}_{avg} = \frac{1}{T} \sum_{t=1}^T \max\left\{\frac{z_t - \mathcal{BP}_t}{\mathcal{BP}_t}, 0\right\} \quad (13)$$

- 3) *Signaling Frequency ( $\mathcal{SF}$ )*: The signaling frequency helps to evaluate a bandwidth provisioning scheme in terms of how often bandwidth allocation/deallocation is done. It directly affects the signaling cost that will be incurred. Though, minimizing the signaling frequency (cost) is not the major objective of the provisioning schemes described here, we present it in order to bring out the impact of provisioning schemes on signaling overhead.

- 4) *Utility Function ( $\mathcal{UF}$ )*: This function helps us evaluate the overall utility of our provisioning framework. It is a linear function that takes into account both the total slack bandwidth and the total under-provisioned bandwidth over the entire forecast period. In other words, it captures the area between the bandwidth envelope and the forecast envelope. As it is undesirable to have under-provisioned bandwidth at any instant, we associate a high cost with the under-provisioned bandwidth in the utility function. Thus, we have

$$\mathcal{UF} = \sum_{\{t: \mathcal{BP}_t > z_t\}} (\mathcal{BP}_t - z_t) + \alpha \sum_{\{t: z_t > \mathcal{BP}_t\}} (z_t - \mathcal{BP}_t) \quad (14)$$

### B. Evaluation

In this section, we evaluate our bandwidth provisioning schemes based on probability-hop forecast as well as the exact forecast for the following choices of  $\Delta\mathcal{B}$ : 2%, 4%, 5%, 8% and 10%. The choices we considered for  $\Delta\mathcal{S}_u$  and  $\Delta\mathcal{S}_d$  are based on the results from Section III-D.

In Fig. 4 and Fig. 5, we present the average utilization ( $\mathcal{U}_{avg}$ ) and average loss ratio ( $\mathcal{LR}_{avg}$ ) respectively for *IBP* scheme. If we look at a particular ( $\Delta\mathcal{S}_u, \Delta\mathcal{S}_d$ ) combination, for a very small  $\Delta\mathcal{B}$ , this scheme shows markedly high utilization

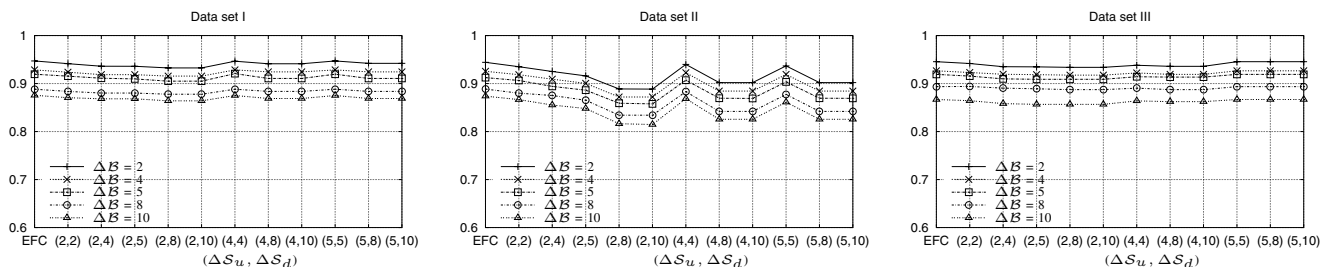


Fig. 4.  $U_{avg}$  for  $IBP$  scheme

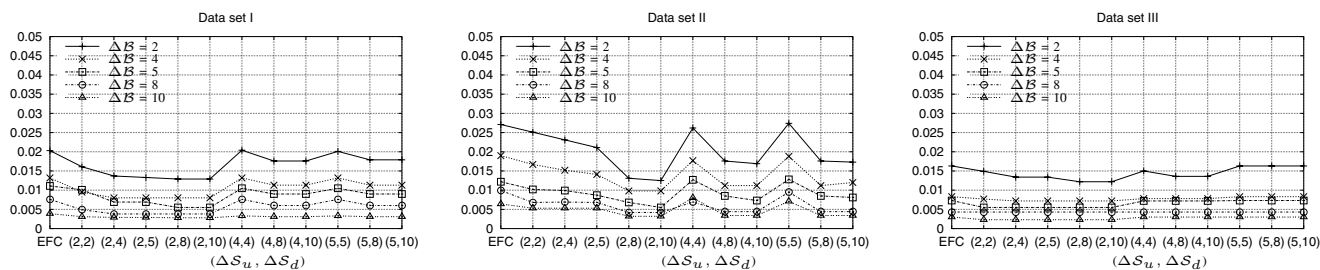


Fig. 5.  $\mathcal{L}_{avg}$  for  $IBP$  scheme

while sacrificing on the loss ratio (for instance, see  $\Delta B = 2\%$ ). This is because, smaller  $\Delta B$  makes  $IBP_{t+1}$  highly sensitive to the probability-hop forecast. As we increase  $\Delta B$  from 2% to 10%, the sensitivity of the bandwidth envelope toward probability-hop forecast decreases. Consequently, we observe a decreasing trend in the average loss ratio which in turn brings down the average utilization. When we look at the loss ratio behavior for  $\Delta B = 2\%$  across different  $\Delta S_u$  and  $\Delta S_d$ , we observe that it is worse when  $\Delta S_u$  and  $\Delta S_d$  are symmetric and higher (see (4%,4%) and (5%,5%)). In fact, it is very close to what we have observed for provisioning based on EFC. On the other hand, if we keep a higher  $\Delta S_d$  than  $\Delta S_u$ , the loss ratio decreases. In particular, such a behavior is more prominent in data set II. We recall here that increasing  $\Delta S_d$  with a given  $\Delta S_u$  decreases the sensitivity of probability-hop forecast toward the lower probability curve. In other words, the PFC will shift between EFC and UPC most of the time and will rarely fall onto LPC. This has the effect of bringing down the loss ratio while compromising on the average utilization which explains the observed behavior. For higher  $\Delta B$  values, we observe a similar behavior with different  $\Delta S_u$  and  $\Delta S_d$  pairs however, in lesser magnitudes due to the higher bandwidth envelope.

The impact of various  $\Delta S_u/\Delta S_d$  on the loss ratio was much evident on data set II than the other two data sets as we look at the results for  $IBP$  scheme. Moreover, our intent here is to show how well behaved the other two provisioning schemes are, as compared to the utopian scheme. Therefore, for  $SBP$  and  $SBPL$  schemes, we illustrate here only the results for data set II.

The  $SBP$  scheme uses a HDT to smoothen the provisioning and make bandwidth adjustments less frequent. This has the considerable effect on bringing down the loss ratio while reducing the average utilization as well. For instance, if we compare the loss ratio behavior for  $\Delta B = 2\%$  in Fig. 4 (data

set II) and Fig. 6, we observe an average gain of nearly 30% from the  $IBP$  scheme to  $SBP$  scheme. At the same time, the average utilization falls down only by 5% on an average. In Fig. 7, we observe that for  $\Delta B = 8\%$  and  $\Delta B = 10\%$ , the impact of sensitivity quanta on the average loss ratio is negligible as against the observation from  $IBP$  scheme (see Fig. 5).

In Fig. 8 and Fig. 9, we present the results for the  $SBPL$  scheme. In this scheme, in addition to smoothing using a hold down timer, we follow a more conservative deallocation by considering the local maximum of the last hold down timer period. For smaller bandwidth quanta, such a deallocation policy helps very well in decreasing the average loss ratio further as compared to the policy guided only by the instantaneous bandwidth requirement. For instance, if we compare the loss ratio for  $SBPL$  scheme (from Fig. 9) with  $SBP$  scheme (from Fig. 7) for a bandwidth quantum of 2%, we observe that the loss ratio from  $SBPL$  scheme completely lies below that of  $SBP$  scheme. At the same time, the compromise here on average utilization is negligible. However, we do not observe any significant change between these two schemes for higher bandwidth quanta.

Next, we tabulate the signaling overhead incurred with all the three provisioning schemes. We present here the results for  $\Delta B = 2\%$  and  $\Delta B = 10\%$  in Table II and Table III respectively. These two extremes can effectively summarize the impact of increasing/decreasing bandwidth quantum on the signaling overhead. We present here the results for data set I and II. The results in the case of data set III were very similar to data set I and hence, we omit them here.

It is evident from Table II that there is a significant reduction in signaling overhead as we move from  $IBP$  to  $SBP$  scheme. Further, for  $SBPL$  scheme, we observe more reduction on the signaling overhead which can be attributed to the more conservative deallocation policy. For  $\Delta B = 10\%$ , we also



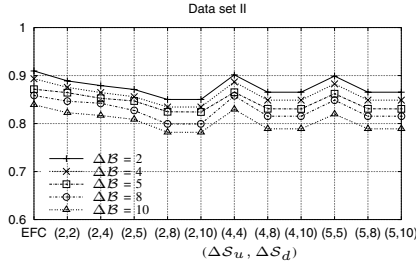


Fig. 6.  $U_{avg}$  for  $SBP$  scheme

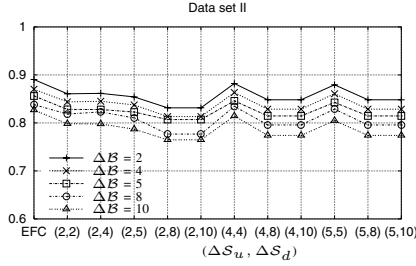


Fig. 8.  $U_{avg}$  for  $SBPL$  scheme

observe a similar behavior (see Table III) across all the schemes. However, the overhead incurred here in general is considerably lower than that of  $\Delta B = 2\%$ . This is due to the greater smoothing effect of  $\Delta B = 10\%$  over the forecasts. In other words, larger the bandwidth quantum, more the number of forecasts that will map onto the same interval.

If we look at the impact of change in sensitivity up and down quantum on the signaling overhead in data set I, we observe a more oscillatory behavior in the case of  $IBP$  scheme as compared to the other two schemes. In general, an increase in  $\Delta S_u$  brings out a gradual reduction in the overhead. We observe here that the signaling overhead is minimum with EFC across all the provisioning schemes. However, we recall here that the data loss is maximum with EFC across all the provisioning schemes (see Fig. 7 and Fig. 9). Moreover, the provisioning schemes are not designed to minimize the signaling overhead, rather to guarantee an acceptable loss ratio while maintaining a reasonably high utilization. Thus, it is possible to expect such a behavior here. We have presented before the fraction of under-forecast instances (UFI) with probability-hop forecast as well as with the exact ARIMA forecast in Table I. Here, we illustrate the fraction of under-provisioned instances (i.e.,  $z_t - \mathcal{BP}_t > 0$ ) denoted by “UPI” along with UFI (50% limits) in Table IV that helps us understand the smoothing effect of bandwidth quantum over the forecast generated. It is evident here that even for smaller bandwidth quantum, the provisioning module avoids under-provisioning fairly well in spite of many forecast misses. This effect is more pronounced in the case of  $\Delta B = 10\%$ . In particular, if we look at the values for  $SBPL$  scheme, UPI reduced substantially as compared to UFI. Note here that UPI is maximum with EFC across all the provisioning schemes.

Finally, we provide the results based on the utility function ( $U$ ) that help us evaluate the overall utility of our bandwidth

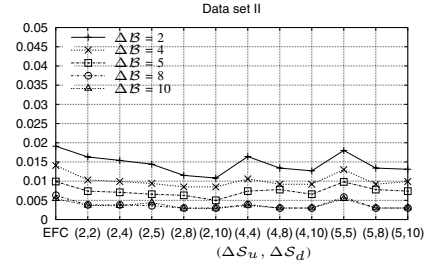


Fig. 7.  $LR_{avg}$  for  $SBP$  scheme

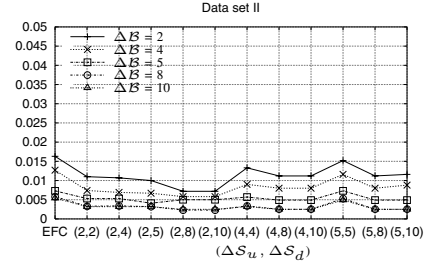


Fig. 9.  $LR_{avg}$  for  $SBPL$  scheme

TABLE II  
SIGNALING FREQUENCY ( $\Delta B = 2\%$ )

$\Delta S_u, \Delta S_d$	Data Set I			Data Set II		
	$IBP$	$SBP$	$SBPL$	$IBP$	$SBP$	$SBPL$
EFC	77.50%	43.33%	33.33%	80.83%	44.17%	33.33%
(2%,2%)	83.33%	46.67%	40.00%	85.83%	48.33%	40.00%
(2%,4%)	85.00%	45.00%	36.67%	84.17%	46.67%	40.00%
(2%,5%)	85.00%	45.00%	36.67%	83.33%	47.50%	40.83%
(2%,8%)	84.17%	45.00%	37.50%	80.83%	46.67%	36.67%
(2%,10%)	84.17%	45.00%	37.50%	80.83%	46.67%	36.67%
(4%,4%)	80.00%	45.00%	35.00%	82.50%	45.83%	35.00%
(4%,8%)	76.67%	45.00%	35.00%	80.83%	45.00%	35.83%
(4%,10%)	76.67%	45.00%	35.00%	80.83%	45.00%	35.83%
(5%,5%)	78.33%	44.17%	34.17%	80.83%	45.00%	36.67%
(5%,8%)	75.00%	44.17%	34.17%	80.83%	45.00%	35.83%
(5%,10%)	75.00%	44.17%	34.17%	80.83%	45.00%	35.83%

TABLE III  
SIGNALING FREQUENCY ( $\Delta B = 10\%$ )

$\Delta S_u, \Delta S_d$	Data Set I			Data Set II		
	$IBP$	$SBP$	$SBPL$	$IBP$	$SBP$	$SBPL$
EFC	36.67%	25.00%	15.00%	27.50%	20.00%	13.33%
(2%,2%)	37.50%	23.33%	15.83%	34.17%	24.17%	19.17%
(2%,4%)	39.17%	21.67%	15.83%	35.00%	25.83%	15.83%
(2%,5%)	39.17%	21.67%	15.83%	40.00%	26.67%	15.83%
(2%,8%)	39.17%	21.67%	15.83%	37.50%	22.50%	13.33%
(2%,10%)	39.17%	21.67%	15.83%	38.33%	22.50%	13.33%
(4%,4%)	36.67%	22.50%	14.17%	30.83%	21.67%	15.00%
(4%,8%)	35.83%	21.67%	14.17%	36.67%	21.67%	14.17%
(4%,10%)	35.83%	21.67%	14.17%	36.67%	21.67%	14.17%
(5%,5%)	36.67%	22.50%	14.17%	34.17%	23.33%	16.67%
(5%,8%)	35.83%	21.67%	14.17%	36.67%	21.67%	14.17%
(5%,10%)	35.83%	21.67%	14.17%	36.67%	21.67%	14.17%

TABLE IV  
UNDER-PROVISIONED INSTANCES

$\Delta S_u, \Delta S_d$	Data set II						
	UFI	$\Delta B = 2\%$			$\Delta B = 10\%$		
		$\mathcal{IBP}$	$\mathcal{SBP}$	$\mathcal{SBPL}$	$\mathcal{IBP}$	$\mathcal{SBP}$	$\mathcal{SBPL}$
EFC	55.00%	44.17%	31.67%	24.17%	10.00%	8.33%	7.50%
(2%,2%)	50.00%	40.83%	25.83%	17.50%	9.17%	6.67%	5.00%
(2%,4%)	44.17%	36.67%	23.33%	17.50%	8.33%	5.83%	4.17%
(2%,5%)	38.33%	32.50%	20.83%	16.67%	8.33%	6.67%	4.17%
(2%,8%)	17.50%	15.83%	11.67%	8.33%	3.33%	2.50%	1.67%
(2%,10%)	17.50%	15.83%	11.67%	8.33%	3.33%	2.50%	1.67%
(4%,4%)	50.83%	40.00%	28.33%	20.00%	10.00%	6.67%	5.00%
(4%,8%)	27.50%	23.33%	15.83%	12.50%	5.00%	3.33%	2.50%
(4%,10%)	27.50%	23.33%	15.83%	12.50%	5.00%	3.33%	2.50%
(5%,5%)	49.17%	41.67%	29.17%	21.67%	10.83%	8.33%	6.67%
(5%,8%)	27.50%	23.33%	15.83%	12.50%	5.00%	3.33%	2.50%
(5%,10%)	27.50%	23.33%	15.83%	12.50%	5.00%	3.33%	2.50%

provisioning framework. The various choices of  $\alpha$  chosen here are: 1, 5, 10, 25, 50, 100. Our objectives are twofold: (1) to show the impact of  $\alpha$  on the utility function, (2) to show the extent of improvement that can be achieved through probability-hop forecast over exact ARIMA forecast. Notice here that our utility function is essentially a cost function i.e., *higher value implies lower utility*. Let us denote the utility function value for the case of bandwidth provisioning based on the exact ARIMA forecast by  $\mathcal{UF}_{\text{EFC}}$ . In order to fulfill our objectives, we chose  $\mathcal{UF}_{\text{EFC}}$  as the reference point and illustrate the fractional increase or decrease in the utility function value with respect to this reference point for different choices of  $\alpha$ .

For all the three provisioning schemes with  $\Delta B = 2\%$  (see Fig. 10), it is evident that for  $\alpha = 1$ ,  $\mathcal{UF}$  value for various  $(\Delta S_u, \Delta S_d)$  pairs lies always above  $\mathcal{UF}_{\text{EFC}}$ . By choosing  $\alpha = 1$ , we do not associate any penalty with the under-provisioned bandwidth. Moreover, it is true that for any given  $\Delta B$ , the total amount of excess bandwidth is upperbounded by the probability-hop based forecast due to the smoothing effect of  $\Delta S_u$  and/or  $\Delta S_d$ . Then, as we increase  $\alpha$ , the value of  $\mathcal{UF}$  decreases gradually. Such an effect is more pronounced with  $\mathcal{IBP}$  scheme as compared to other two schemes. The reason being that the conservative deallocation rule is followed in these schemes which in turn contributes to the excess bandwidth and possibly, decreases the under-provisioned bandwidth as well. In the case of  $\Delta B = 10\%$  (see Fig. 11), the utility of our probability-hop limit based framework becomes prominent for relatively higher values of  $\alpha$ . The smoothing effect of bandwidth quantum contributes to such a behavior.

### C. Observations Summary

The major control parameters underlying our two stage approach of bandwidth provisioning are  $\Delta S_u$ ,  $\Delta S_d$  and  $\Delta B$ . The sensitivity quantum smooths out the small fluctuations in forecast while the bandwidth quantum achieves a second-level of smoothing. These three parameters can be viewed as the

three dimensions of our bandwidth provisioning framework. The sensitivity quantum plays an essential role in calculating effective forecasts by taking advantage of the probability limits. In addition, it helps to offset the effect due to a possible change in the ARIMA parameter estimates that can happen as the time series evolves. The choices of  $\Delta S_u$  and  $\Delta S_d$  used here might not give the best results in a different data set. Nevertheless, one general rule could be to have an higher  $\Delta S_d$  than  $\Delta S_u$ . In this way, we have a liberal allocation and a more controlled deallocation. In fact, an interesting approach could be to change  $\Delta S_u, \Delta S_d$  adaptively. We are still investigating this issue. While  $\Delta S_u$  and  $\Delta S_d$  helps to reduce forecast misses, we have  $\Delta B$  that further controls the data loss as well as the average utilization. Smaller  $\Delta B$  improves the average utilization while increasing the loss ratio whereas larger  $\Delta B$  helps to reduce loss ratio by compromising on the average utilization. Hence, the choice of  $\Delta B$  could be guided by the loss ratio guarantee alone or by the utilization bounds or by both. Moreover, an adaptive approach toward changing  $\Delta B$  could be interesting to investigate further. Finally, it is evident from the numerical results that the dependency between  $\Delta S_u/\Delta S_d$  and  $\Delta B$  is quite non-linear which is another interesting issue to investigate further.

## VI. SUMMARY AND CONCLUSION

In this work, we consider dynamic traffic being offered to a dynamically reconfigurable network link. We have presented the Probability Hop Forecast Algorithm, a variation of exact ARIMA forecasting, to effectively model and predict dynamic, non-stationary traffic. The forecast generation is implemented on a temporal network management environment. We have also described several bandwidth provisioning schemes to maintain QoS requirements of forecasted traffic.

For our studies, we have chosen three different sets of real-world data. We have identified ARIMA(2,1,1) model to be a better fit for all the three data sets. It is possible that the orders of autoregressive ( $p$ ) or moving average component ( $q$ ) or both may change with some other real-world data. However, according to [16], in practice, it is very rare to come across any time series that has the value of  $p$  and  $q$  more than 2. In addition, it is imperative to consider adequate but parsimonious (with respect to number of parameters involved) models while characterizing a dynamic system [16]. Therefore, our model has a fairly general value. Then, we have chosen a lead time of *one* ( $l = 1$ ) in this work thereby making use of the new observation as soon as it is available for future forecasts. Nevertheless, it would be interesting to consider a lead time greater than one which we are planning to address in our future work.

As shown through our work here, it is indeed possible to create a reasonable bandwidth envelope using the approach we have developed. We have provided numerical analysis to show the effectiveness of our approach. In regard to the overall utility of our bandwidth provisioning framework, we defined an utility function based on a control parameter  $\alpha$  to account for penalty due to data loss. The value of  $\alpha$  might vary for

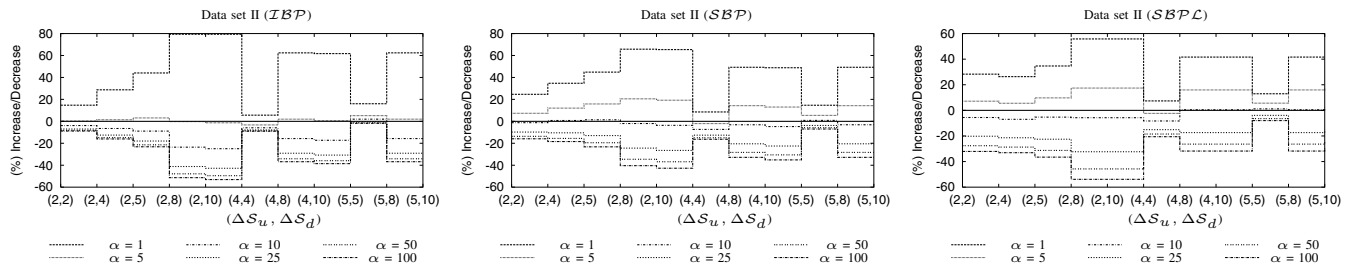


Fig. 10. (%) Increase/Decrease in  $U_{\mathcal{F}}$  with respect to  $U_{\mathcal{F}_{\text{EFC}}}$  ( $\Delta B = 2\%$ )

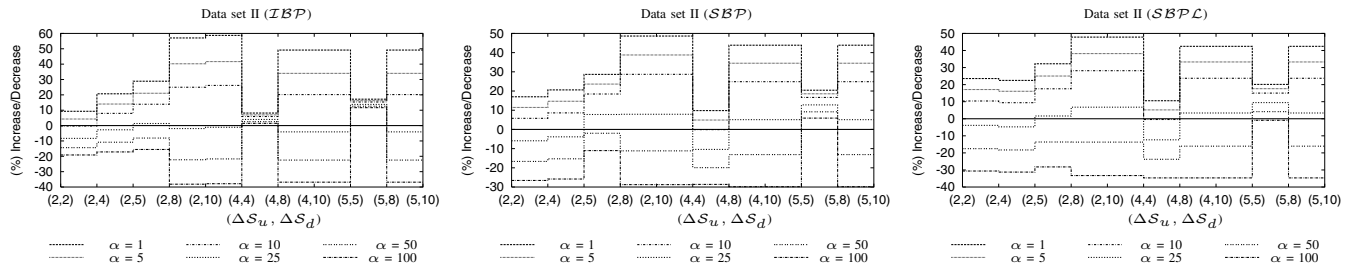


Fig. 11. (%) Increase/Decrease in  $U_{\mathcal{F}}$  with respect to  $U_{\mathcal{F}_{\text{EFC}}}$  ( $\Delta B = 10\%$ )

different networks. However, the choices of  $\alpha$  considered here helped us bring out their impact on the utility function. We hope that our approach can be of value in adaptive traffic engineering.

Finally, an added advantage of using an adaptive technique is that bandwidth adjustment is done proactively for a future time instant based on traffic measurements available till current time. Hence current traffic load is not thawed by the wait generally involved with the update function of a bandwidth adjustment scheme which allocates or deallocates bandwidth. Such a predictive bandwidth provisioning system, which adapts to changes in traffic, can lead to better use of network resources, since idle capacity for a service class can be deallocated for use on a packet-by-packet or byte-by-byte basis by another service class. A possible after-effect is that invoking the update function for bandwidth adjustment too frequently might lead to network vulnerability.

As can be seen from our experimental results, while our approach works quite well most of the time, it is not perfect, i.e., it can miss the target once in a while. The difficulties remain whenever an extremely unexpected turn in traffic is faced. Further, the bandwidth estimation methods discussed here are designed to maintain a good balance between data loss and utilization without considering its impact on the signaling overhead. It would be ideal to develop a method that in addition to adhering the goal dictated by the loss ratio and utilization metrics, also minimizes the signaling overhead in the same framework. We are currently working on such an approach.

## REFERENCES

- [1] P. Chemouil, J. Filiipiak, "Modeling and Prediction of Traffic Fluctuations in Telephone Networks," *IEEE Trans. on Comm.*, vol. 35, no. 9, pp. 931-941, September 1987.
- [2] D. Tipper, M.K. Sundareshan, "Numerical Methods for Modeling Computer Networks Under Nonstationary Conditions," *IEEE Journal on Selected Areas in Comm.*, vol. 8, no. 9, pp. 1682-1695, December 1990.
- [3] W. Wang, D. Tipper, S. Banerjee, "A Simple Approximation for Modeling Nonstationary Queues," *Proc. of IEEE INFOCOM '96*, pp. 255-262, March 1996.
- [4] B. Groszkinsky, D. Medhi, and D. Tipper, "An Investigation of Adaptive Capacity Control Schemes in a Dynamic Traffic Environment," *IEICE Trans. on Comm.*, vol. E84-B, no. 2, pp. 263-274, February 2001.
- [5] Y. Afek, M. Cohen, E. Haalman, Y. Mansour, "Dynamic Bandwidth Allocation Policies," *Proc. of IEEE INFOCOM '96*, pp. 880-887, March 1996.
- [6] S. Ohta, K. Sato, "Dynamic Bandwidth Control of the Virtual Path in an Asynchronous Transfer Mode Network," *IEEE Trans. on Comm.*, vol. 40, no. 7, pp. 1239-1247, July 1992.
- [7] C. Bruni, P. D. Andrea, U. Mocchi, C. Scoglio, "Optimal Capacity Management of Virtual Paths in ATM Networks," *Proc. of GLOBECOM '94*, pp. 207-211, December 1994.
- [8] A. Orda, G. Pacifici, D. E. Pendarakis, "An Adaptive Virtual Path Allocation Policy for Broadband Networks," *Proc. of IEEE INFOCOM '96*, pp. 329-336, March 1996.
- [9] N. Ng, C. K. Tham, "Connection Admission Control of ATM Networks using Integrated MLP and Fuzzy Controllers," *Comp. Networks*, vol. 31, no. 1, pp. 61-79, January 2000.
- [10] Q. Zhang, J. Wu, H. Xi, "Dynamic Bandwidth Allocation over ATM Based on Neural Network Prediction: Analysis and Simulation," *Technical Report*, School of Information Science and Technology, University of Science and Technology of China.
- [11] R. Feraud, F. Clerot, J. Simon, D. Pallou, C. Labbe, S. Martin, "Kalman and Neural Network Approaches for the Control of a VP Bandwidth in an ATM Network," *Proc. of IFIP Networking 2000*, pp. 655-666, May 2000.
- [12] V. Paxson, S. Floyd, "Wide-Area Traffic: The Failure of Poisson Modeling," *IEEE/ACM Trans. on Networking*, vol. 3, no. 3, pp. 226-244, June 1995.
- [13] W. E. Leland, M. S. Taqq, W. Willinger, D. V. Wilson, "On the Self-Similar Nature of Ethernet Traffic," *Proc. of ACM SIGCOMM '93*, pp. 183-193, September 1993.
- [14] N. K. Groschwitz, G. C. Polyzos, "A Time Series Model of Long-Term NSFNET Backbone Traffic," *Proc. of ICC '94*, pp. 1400-1404, May 1994.
- [15] K. Papagiannaki, N. Taft, Z. Zhang, C. Diot, "Long-Term Forecasting of Internet Backbone Traffic: Observations and Initial Models," *Proc. of IEEE INFOCOM '03*, April 2003.
- [16] G. E. P. Box, G. Jenkins, *Time Series Analysis: Forecasting and Control*. Prentice Hall, February 1994.
- [17] *GNU R Project*, <http://www.r-project.org>.