

Adaptive Parameter Collection in Dynamic Distributed Environments

Zhenghua Fu, Nalini Venkatasubramanian
Information and Computer Science, University of California, Irvine
{zfu,nalini}@ics.uci.edu

abstract Cost-effectively collecting distributed state information is a challenging problem. It perhaps has no *single* perfect answer since different distributed application environments pose different requirements from the information collection process. In any case, knowledge of the environment, in terms of traffic models, load models etc. play a key role in determining tradeoffs between accuracy (needed to ensure QoS requirements) and cost-effectiveness. In this paper, we develop an adaptive information collection algorithm that utilizes network traffic knowledge characterized using a time series model. The algorithm utilizes an information collection architecture consisting of a directory service integrated into the middleware layer with monitoring modules distributed across the network. The cost-effectiveness of the proposed information collection algorithm proposed is verified in simulations over diverse network traffic patterns, i.e. Internet WAN (TCP), MPEG(multimedia) and web access traffic traces. Our results show that the proposed adaptive information collection algorithm compensates for inaccuracies in network traffic predictions in a cost-effective manner.

1 Introduction

In this paper, our objective is to explore effective middleware infrastructures which can be used to support efficient QoS-based resource provisioning algorithms. MM applications have QoS requirements which translate to system level resource requirements (e.g., network link bandwidth, server resources). Resource provisioning algorithms (e.g., QoS based network routing, server selection) utilize current system resource availability information to ensure that applications meet their QoS requirements. Given a *reasonably* good approximation of the current utilization map, network control tasks such as connection admission control, load balancing and congestion control, can be performed highly effectively. Such optimized resource provisioning policies are becoming more crucial with the exponential growth of Internet based services.

In this paper, we consider a management framework consisting of a directory service that holds system information which can be used by resource provisioning algo-

rithms. An important issue in managing the directory information is that of maintaining accurate and current system state information. For this purpose, we study parameter collection mechanisms that collect information from the network and server components that are being provisioned. In highly dynamic environments, where system conditions are constantly changing, it is critical to provide effective approximations of the current system state. There are tradeoffs between accuracy of directory information and the update overhead costs that must be considered in engineering such a parameter collection protocol. The work described in this paper attempts to address this tradeoff by developing simple and efficient policies to collect and approximate the current system state information in highly dynamic network environments.

We propose an adaptive parameter collection protocol using the notion of a dynamic range to describe the residual capacities of resources being provisioned, and describe a simple algorithm to relax and tighten the range. While the proposed approach to parameter collection is general and applies to a variety of resources being provisioned, (e.g., web servers, remote disks, processor capacity, etc.), for concreteness, we focus on using the parameter collection process developed for network traffic measurement and provisioning. Network traffic parameter collection is important for network control tasks such as connection admission control, load balancing for network and hosts, etc.

Network Traffic Measurement Traffic monitoring and measurement is one of the key issues in large scale network management research. The focus of the existing work has been on designing measurement architectures and more recently on traffic modeling and prediction [4, 5, 6, 7, 8] as well as on system architectures to facilitate effective implementation of traffic prediction in real network environments [2, 9]. Existing techniques have taken a simplistic approach towards parameter collection – a monitoring module samples the residual capacity of a network link at fixed intervals and updates the directory using the latest sampled value [9]. The sampling rate determines the quality of the information stored in the directory – in highly dynamic traffic, the monitoring module has to sample at a very high frequency to prevent information from being outdated. Instead of us-

ing a snapshot value, an alternative mechanism proposed in [3] is to partition the link capacity into static intervals, and update the directory with an interval number [3]. This interval based policy, however, does not improve the cost-effectiveness of the monitoring process. First, given the interval, only a uniform distribution can be inferred from it, thus limiting its practical significance. Moreover the size of the static interval does not directly correlate to a sampling period.

Our Approach In contrast to existing approaches, the adaptive parameter collection protocol developed in this paper provides a novel way to cost-effectively measure the network state information. Using statistical analysis techniques, we quantify and estimate network traffic fluctuation. We propose a *2-phase* information collection process that uses simple statistical analysis techniques based on time-series. First, we derive an adaptive range such that the deviation between the predicted and observed values remains in the range with high confidence. Based on the size of the adaptive range and the confidence level, we determine a bound on the rate at which the resource needs to be sampled. The parameter collection process dynamically adjusts the range as well as the sampling rate based on the burstiness of the incoming traffic. To study the cost-effectiveness of our adaptive range-based parameter collection algorithm, we apply our approach to the measurement of network traffic from three different scenarios – TCP, MPEG and WEB. We also develop a metric to statistically measure the effectiveness of using the collected range to perform admission control. Our performance evaluation results indicate that the adaptive range based parameter collection is significantly more cost-effective than static range-based approaches in most of the traffic traces studied. In addition, the robustness of our algorithm in nonstationary traffic environments allows it to be deployed in scalable measurement architectures.

2 A Model for Predicting the Range and Sampling Interval

Our objective in this section is to develop an analytic model for predicting the network traffic for a link. The model developed will be used in the following section to design an adaptive parameter collection/monitoring process. The parameters collected by the monitoring process are used to predict the residual capacity of a resource (e.g., a network link) in the future.

Our parameter collection algorithm maintains a dynamic range such that the deviation of the predicted value from the real observation lies within the range with high probability. Let Z_t be a random variable representing the load at a given link at time t . Let $\hat{Z}_t(l)$ denote the predicted

value at time $t + l$ based on the parameters collected at or before time t . Thus, the error in the prediction is specified by $e_t(l) = |Z_{t+l} - \hat{Z}_t(l)|$. Our algorithm determines the tightest possible range $[\hat{Z}_t(l) - R^*, \hat{Z}_t(l) + R^*]$ such that the actual value (Z_t) lies within the range with high probability. More formally, we define the notion of an ε -optimal range as follows:

Definition 1 (ε -optimal range) *Given a confidence level ε , at time t , we define an ε -optimal range as $[Z_t - R^*, Z_t + R^*]$, where*

$$R^* = \min \left\{ R \mid Pr(|Z_{t+1} - \hat{Z}_t(1)| > R) < \varepsilon \right\}$$

Notice that by the definition above, the ε -optimal interval guarantees that the deviation based on the parameters collected upto time t at the next time step $t + 1$ is bounded within the range $[\hat{Z}_t(1) - R^*, \hat{Z}_t(1) + R^*]$. It is expected that the error in prediction (deviation) will increase with time. That is, the estimation of residual resource capacity at a time $t + l$ ($\hat{Z}_t(l)$) based on parameters collected at or before time t , will increase with l unless new samples of actual value Z_t are obtained. We next define an optimal sampling interval as the time unit T^* such that the deviation between the predicted value and the actual value lies within the interval R^* with high probability for T^* units. Formally, the optimal sampling interval is defined as follows:

Definition 2 (Optimal Sampling Interval) *Given an optimal ε -range as $[Z_t - R^*, Z_t + R^*]$, we define an optimal sampling interval to be T^* , where*

$$T^* = \max \left\{ T \mid Pr(|Z_{t+T} - \hat{Z}_t(T)| > R^*) < \varepsilon \right\}$$

In the next section we will develop an adaptive parameter collection protocol based on the estimation of range parameter R^* and the optimal sampling interval T^* . In the remainder of this section, we establish how the monitoring process can determine R^* and T^* for a network link.

For this purpose, we need an analytic model for network link traffic. It is well known that internet traffic is difficult to model using simple models since the inter-arrival process of the packets is not independent. Instead, the traffic arrival patterns have long-lived correlations that impact traffic patterns for a significant duration in the future [4]. Such long range dependencies in the auto-covariance gives rise to the "self-similar" phenomenon of the traffic series in different time-scales [10]. Such self-similar traffic patterns can be generated by multiplexing a large number of ON/OFF sources which have their ON and OFF period follow a *perato* distribution[5]. However in practice, the heavy-tailed distribution can easily give rise to large variances, which poses difficulties for data analysis and prediction [4]. Since our goal is to draw statistical inferences from history values, we model the incoming network

traffic using a time-series model. Specifically, we fit the *Auto-Regressive Moving Average* (ARMA) model for this purpose. In the ARMA model [21], a stationary stochastic process Z_t is modeled as a linear combination of Gaussian random variables, a_t , that follows a normal distribution $N(0, \sigma_a^2)$

$$Z_t = \mu + \sum_{i=0}^{\infty} \psi_i a_{t-i} \quad \psi_0 = 1, |\psi_i| < 1 \text{ for } i > 0 \quad (1)$$

where μ is the mean of Z_t and ψ_i is the ARMA model parameter.

Suppose we are at time t , the least square error prediction of Z_{t+l} can be estimated by setting future noise input a_{t+1}, \dots, a_{t+l} to be their expected value to zero. That is,

$$\hat{Z}_t(l) = \mu + \sum_{i=l}^{\infty} \psi_i a_{t+l-i} \quad (2)$$

The forecast error of leading time l can be estimated as follows:

$$e_t(l) = |Z_{t+l} - \hat{Z}_t(l)| = \sum_{i=0}^{l-1} \psi_i a_{t+l-i} \quad (3)$$

and its mean and variance is given as

$$E[e_t(l)] = 0 \quad VAR[e_t(l)] = \sigma_a^2 \cdot \sum_{i=1}^{l-1} \psi_i^2 \quad (4)$$

The random variable a_t is called the noise input at time t . It is also be called an estimation residue, since $e_t(1) = a_t$, it is the deviation between actual value Z_t and the prediction $\hat{Z}_{t-1}(1)$. During the process of ARMA model estimation, the estimation residual series will be tested to see if it fits a normal distribution. If not, the ARMA model needs to be re-fitted, and parameters recalculated. In practice, there are standard statistics toolbox to do model identification, estimation and model diagnostics. In the following discussion, we assume that estimation residue series a_t does follow $N(0, \sigma_a^2)$ and the model fits well.

Based on the ARMA model for network traffic, we can establish a bound on the dynamic range parameter R^* and the sampling interval T^* discussed earlier.

Proposition 1 (Lower bound for R^*) *Let ε be a confidence level between 0 and 1, and let R^* be defined as:*

$$R^* = \min \left\{ R \mid Pr(|Z_{t+1} - \hat{Z}_t(1)| > R) < \varepsilon \right\}$$

If Z_t follows the ARMA model, the following holds:

$$R^* \geq \phi(1 - \varepsilon/2) \cdot \sigma_a \quad (5)$$

where $\phi(\cdot)$ is the inverse cdf of $N(0, 1)$.

This proposition tells us that based on parameters collected at or before time t , the most accurate possible ε -range for next step prediction is $[\hat{Z}_t(1) - R^*, \hat{Z}_t(1) + R^*]$. From (5), we see that R^* has a simple linear relationship with σ_a . Intuitively, the smaller the estimation residual, σ_a , the better the parsimony the incoming traffic exhibits, which enables us to derive a better prediction. For a formal proof, please refer to [24].

Due to the bursty nature of network traffic, series a_t exhibits highly dynamic *local* behavior. During the network burst period, such as addition or removal of large number of connections, the variance of a_t rises, which makes ε -range larger. While during other period as traffic becomes predictable and conforms to the ARMA model, the range narrows gradually providing better approximation.

In order to capture this local behavior, later in section 3, we will use $\hat{\sigma}_a(t)$ in stead of σ_a to derive R^* .

$$\hat{\sigma}_a(t) = \left(\frac{1}{M} \cdot \int_{t-M+1}^t a_x^2 dx \right)^{1/2} \quad M > 0, \quad (6)$$

In the above equation, M is the memory size which determines how long ago should the prediction model remember. A small M makes the ε -range sensitive to higher frequency traffic components, while a larger M causes the change in R^* to be more gradual in both directions. We will discuss the impact of varying M on the measurements in the context of the TCP, MPEG and WEB traffics in the section on performance analysis.

In the following proposition, we establish the closed form equation for the optimal sampling interval T^* when the traffic follows the ARMA model.

Proposition 2 (Maximum Sampling Interval for R^*)

Given ε , the maximum sampling interval for range size R^ , which is derived from (5), is*

$$T_t^* = \Psi^{-1} \left(\frac{\hat{\sigma}_a^2}{\sigma_a^2} \cdot \frac{\phi^2(1 - \varepsilon/2)}{\phi^2(1 - \varepsilon)} \right) \text{ where} \quad (7)$$

$$\Psi(t) = \int_0^t \psi_x^2 dx \quad (8)$$

and ψ_x is the continuous form of our ARMA model parameters.

The term $\hat{\sigma}_a^2/\sigma_a^2$ quantifies the level of short term burstiness in traffic. From equation (7), if we fix the global deviation σ_a and a confidence level ε , T_t^* monotonously decrease and increase with σ_a . From equation (5), this implies that holding a smaller ε -range requires higher sampling rate, while holding bigger ranges involves lower sampling requirement. Due to space limitation, for a detailed proof and discussion of the two propositions, please refer to [24].

3 The Adaptive Parameter Collection Algorithm

In this section, we describe our two-phased information collecting algorithm for monitoring residual bandwidth of the network links. The collection algorithm is based on the ARMA model for traffic and uses the range parameter R^* and sampling interval T^* derived in the previous section. Our goal is to maximize the measurement accuracy while minimizing the sampling as well as directory update overhead cost.

Measurement Architecture We assume that network monitoring modules are distributed among the network and that each module monitors a portion of the entire network domain. A monitoring module samples the network to collect the system state information such as current residual bandwidth capacity and update the directories. Notice that the directories may not be placed at the monitoring module. In fact, within an administrative domain, the monitoring modules may be placed near the critical subnets in order to reduce the sampling cost, while the directories may be placed close to the client access points [13] to provide convenient resource provisioning service. Since directory update requires both network bandwidth and computation (CPU cost) at the directory service, the cost of the parameter collection process is proportional to the number of directory updates. In the following discussion, we will assume that monitoring modules have enough processing power for linear model calculation on each monitoring module.

The Adaptive Collecting Process The adaptive collection process is represented using a state transition diagram in Figure 1 with four states: Regular Probing(RP), steadiness Identification(SI), Transient Noise Filtering(NF), and Range Adjusting(RA).

Initially, the monitoring module observes the traffic at the monitored link and determines the prediction model by fitting the parameters into the ARMA model discussed in the previous section. Once the prediction model has been determined, the collection process enters the regular probing state. In the regular probing state, based on the sampling interval (that is, T^* determined in the previous section), the link is periodically probed. If a sampled value falls out of the predicted ε -confidence range (that is, R^*), the module enters a filtering state to prepare to potentially expand the range; otherwise, if the value conforms to the predicted range, the monitoring module enters the steadiness identification state to attempt to tighten the range.

We next explain the details of the collection process by discussing the actions taken at each state and the resulting transition in the state-transition diagram shown in Figure 1.

[A.] Upon entering a Regular Probing state at time t , the monitoring module calculates $Z_a = \hat{Z}_t(1)$ as the expected next step prediction with equation (2). Furthermore,

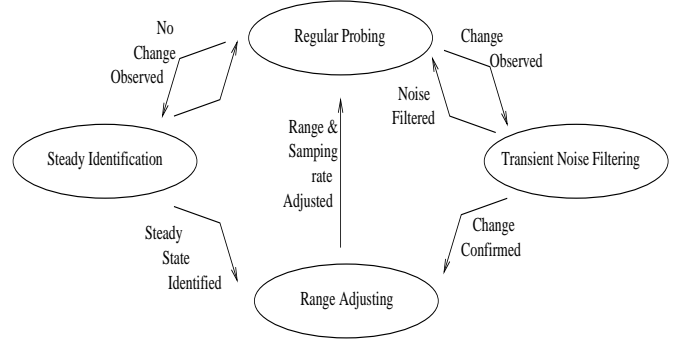


Figure 1: State Transition Diagram Representing the Parameter Collection Process

as discussed in the previous section, it determines the current value for range parameter $R_a = R^*$ and the sampling interval $T_a = T^*$ based on Z_a . When the sampling time is due, the monitoring module sends out probes to corresponding network node to collect link state information. Upon getting the sample, s , with the link's state value $v(s)$ and a time stamp $t(s)$, the monitoring module writes the estimation residual $a_t = v(s) - Z_a$ into its memory table. Then the monitoring module compares $v(s)$ with R_a to determine the state transition.

[B.] If the value $v(s)$ fall into the existing active range R_a , no change is observed. Proceed to the Steadiness Identification (SI) state to see if the link state has been steady for some time. If $v(s)$ fall out of R_a , change is observed, we proceed to the Transient Noise Filtering (NF) state.

[C.] In the SI state, we decide if range R_a should be tightened to provide higher accuracy. To do this, we calculate $\hat{\sigma}_a^{new}$ based on samples currently in the memory table by equation (6). Suppose the old estimation variance that generated the current active range R_a is $\hat{\sigma}_a^{old}$, if $\hat{\sigma}_a^{new} / \hat{\sigma}_a^{old} \leq 1 - TH_s$, where TH_s is a threshold between 0 and 1 which serves as one of the administrative tuning knobs¹, we anticipate more steadiness for the link. Therefore we tighten current range accordingly in the Range Adjusting(RA) state. If otherwise, the new estimation variance hasn't been significantly reduced, the process goes back to the RP state and waits for more input.

[D.] In the RA state, the range size is adjusted according to the memory table. Consider the case of range tightening triggered by the SI state. The new size R_a (set equal to the range parameter R_t^*) is computed using equation (5) as follows:

$$R_a \Leftarrow \phi(1 - \varepsilon/2) \cdot \hat{\sigma}_a^{new}, 0 > R_a > C \quad (9)$$

where C is the capacity of the link. And the new sampling

¹In our simulation, TH_s is related with windows size M ; the bigger the M , the closer to 0 for the TH_s , vice versa.

interval, by equation (7), is adjusted to be

$$T_a \leftarrow \Psi^{-1} \left(\frac{\hat{\sigma}_a^2}{\sigma_a^2} \cdot \frac{\phi^2(1 - \varepsilon/2)}{\phi^2(1 - \varepsilon)} \right), T_a \geq T_{min} \quad (10)$$

where $\Psi^{-1}(\cdot)$ is defined in equation(8) in section 2. T_{min} is a *cut off* limit imposed on adaptive sampling interval. By setting this parameter, the network administrator is able to limit the maximum sampling cost. Notice that in the RA state, the current prediction model to determine if a goodness of fit test fails and execute the parameter refitting process as outlined in [21, 2].

[E.] The collection process enters the NF state from the regular probing state due to an ε -range miss. In practice there could be various reasons for the ε -range miss: measurement error, a transient burst or a significant load level change. In order to 1) filtering out high frequency traffic components 2) assist the underlying model to adapt to a confirmed change, the monitoring module sends multiple probes over a given probing period P_p , at a probing rate P_r . After the probing period, we count the number of samples with values out of the active range R_a . Theoretically, the ratio of the outliers over the total numbers of samples collected should be less than ε . Therefore, if the ratio is less than this bound, the samples collected during the probing period can be discarded. otherwise, the observed change is confirmed and the range parameter R_a needs to be expanded. In this case, we record the samples during probing period to the memory table, and proceed to the Range Adjust state to calculate the new value of the range parameter R_a and the sampling interval T_a .

[G.] If a significant change is confirmed by NF state, a burst series should be recorded in the memory table. Depending on its size M , the $\hat{\sigma}_a$ will increase more or less (equation (6)), causing both the range size and the next step sampling interval to increase in RA state. This puts the measurement in a low cost and low accuracy operation. When more steady states are identified, it approaches by gradually improving the accuracy with more active sampling. From the later simulations, we show that this is an effective method to achieve cost-effectiveness and prevent oscillation in nonstationary traffic.

3.1 Algorithm Discussions

Identifying Bursts in the NF State. To identify a burst in traffic in the NF state, we used a ratio of outliers samples (that is, samples which were out of the range R_a based on current prediction) to the total number of samples collected and compared the ration to ε . In practice, given the short probing period, the ratio of outlier in the sample will have a large variance. Let O_i be a random variable that takes a value 0 or 1 depending on whether the i th sampled value fall out of range R_a from the predicted model (that

is, it is an outlier). From the discussion in Section 2, $Pr\{O_i = 1\} = \varepsilon$ and $Pr\{O_i = 0\} = 1 - \varepsilon$. The expected value of O_i is ε and the variance is $\varepsilon(1 - \varepsilon)$. The ratio can be written as $RATIO = (\sum^n O_i)/n$, where n is the total number of samples in the probing period and the variance of the ratio $VAR(RATIO) = [\varepsilon(1 - \varepsilon)]/n = \sigma_r^2$. By Chebyshev's inequality, with less than probability $1/k$, $RATIO > \varepsilon + \sqrt{k} \cdot \sigma_r$. In practice, we found that combined with proper choice of probing period, taking $k = 3$ is very effective in identifying a significant state change of the measured traffic.

Directory Update Cost. Since the monitoring module and the directory are distributed, sending a range information over the network and performing a directory updates have a significant cost. To reduce the number of updates needed, the monitoring module can send, in stead of the 1 step prediction $\hat{Z}_t(1)$, send in 1 to l step predictions $\hat{Z}_t(1), \hat{Z}_t(2), \dots, \hat{Z}_t(l)$ along with the prediction range size R_a . Thus, no updates is needed as long as $R_a(t)$ doesn't change. A directory update occurs only because R_a adjusted or l steps exceeded. In these cases, the monitoring module updates the directory with new 1 to l step predictions and R_a . At the directory service side, it simply steps the range prediction according to the pre-informed range size and 1 to l step predictions. For step i , the active range should be $[\hat{Z}_t(i) - R_a, \hat{Z}_t(i) + R_a]$. Later in the simulation, we express this technique as *update reduction*.

4 Performance Evaluation

In this section, we evaluate our prediction model and the adaptive information collection protocol using 3 diverse traffic traces obtained from TCP-WAN traffic loads, multimedia (MPEG) traffic and Web request traffic.

4.1 Experimental Setup

Before we present the performance (simulation) results, we briefly report the parameters used in the simulation, table 1, as well as some implementation details. In calculating T_a , we need to approximate a continuous model from the discrete sampled values; we use a value of one second as the smallest unit of measure and use discrete summation for the integral value.

In the simulation, we evaluate the adaptive information collection algorithm with 3 different traffic traces that are of varying nature - (a)TCP(WAN) traffic [18], (b)MPEG(multimedia) traffic [19] and (c)pure Web accesses [20]. The tcp trace obtained from Lawrence Berkeley Labs (LBL) is collected on a WAN backbone link connecting LBL and the Internet using *tcpdump*. The mpeg

trace is collected by recording the number of bytes for each frame generated by a mpeg decoder playing back the movie *star war*. The web trace was collected at EPA, by recording every request/reply between the web server and the rest of the world during a whole day. A more detailed description of these traffic traces can be found at [24].

4.2 Performance Results

As described in section 3, the design objective of our algorithm is to try and provide better accuracy whenever possible (i.e. during relatively stable states), and tradeoff accuracy for cost-effectiveness when in the bursty states (since it is extremely expensive and difficult to predict accurately in the presence of bursts anyway). It is therefore critical to effectively identify bursty states and quickly adapt to them. Unless the underlying prediction model is perfect, using a large history table generally isn't practical, since its long memory of bursty states prevents it from identifying steady states effectively. In other words, long memory (larger window size) sacrifices the accuracy that can be obtained from steady state traffic to try and accommodate more bursty states, which is contrary to our design goal. Furthermore, the accuracy of state prediction as derived from the analytical time series model can vary. The goal of the adaptive information collection algorithm is to allow for variations in predictions (which are likely to occur in highly dynamic and intractable environments) without sacrificing significantly on either performance or accuracy.

4.2.1 Cost-effectiveness study

For comparison purposes, we introduce a *static* range based measurement algorithm here. The static algorithm, briefly described in Section 1 uses predefined fixed ranges. It samples the traffic with a fixed interval and represents a sampled value using a range in which it falls. It uses this range

param.	description	setting
ε	the confidence level	0.05
M	size of the sliding window (1sec-per-entry)	specified
T_{min}	the lower cut off limit for $T_a: T_a \geq T_{min}$	specified
T_{max}	the upper cut off limit for $T_a: T_a \leq T_{max}$	$max(14, T_{min})$
P_p	the probing period	$max(10, T_{min})$
P_r	the sampling rate in RP	T_{min}
TH_s	the triggering threshold in SI	0.1
k	the triggering threshold in NF	3.0

Table 1: parameter settings in simulation implementation

approximation until the next sampling. Intuitively, with the static range-based algorithm, a smaller static interval implies greater accuracy and increased directory update overhead. In order to quantitatively evaluate the degree of accuracy in using a range representation, we introduce a metric, *acc* which compares predicted value wrt to the original trace value. It is defined as follows:

Definition 3 (*acc* for adaptive range) For ε -range: $\langle L, U \rangle$, derived by adaptive range based measurement, to approximate a random value Z_{t+1} , the degree of accuracy is defined as

$$acc = Pr\{X = Z_{t+1}\}, \text{ where } X \sim N\left(\frac{L+R}{2}, \frac{R-L}{\phi(1-\varepsilon/2)}\right)$$

In Section 2, we show $\frac{L+R}{2} = \hat{Z}_t(1)$ is the expected next step prediction, and by equation 5 $\frac{R-L}{\phi(1-\varepsilon/2)} = \hat{\sigma}_a$, is the deviation of the estimation residual for next step prediction. So the above normal distribution is the predicted pdf of the next step value Z_{t+1} . Naturally, the probability value at Z_{t+1} in this pdf indicates the quality of this range approximation. Note that the nearer the Z_{t+1} gets to $(L+U)/2$, the higher the *acc*, also the smaller the range size, the higher the *acc* for values within the range.

We now define a similar metric *acc* for the static range.

Definition 4 (*acc* for static range) For a static range: $\langle L, U \rangle$, derived by static range based algorithm, to approximate a random value Z_{t+1} , the degree of accuracy is defined as

$$acc = Pr\{X = Z_{t+1}\}, \text{ where } X \sim uniform(L, U)$$

The choice of a static range in representing a value is based only on the occurrence of the value in that range; hence we represent the next step value as a uniform distribution over the interval of the selected range. In the following, we evaluate the accuracy, sampling cost and directory cost of the adaptive parameter collection algorithm, and compare them with the *static* range based measurement algorithm. For a given traffic trace, we run the measurement program multiple times with different settings for history table size M and cut off sampling interval T_{min} . For each of the measurement runs, we note (a) the *weighted*² average *acc* value, (b) the total number of samples initiated and (c) directory updates performed. The following three sets of performance graphs for TCP-WAN, MPEG and Web traffic were obtained using the described process.

TCP traffic measurement: Due to the effect of multiplexing (multiple connections), the tcp trace is less bursty than the single MPEG and web traces. The performance

²we weight each *acc* based on the sampling interval between two successive measurements

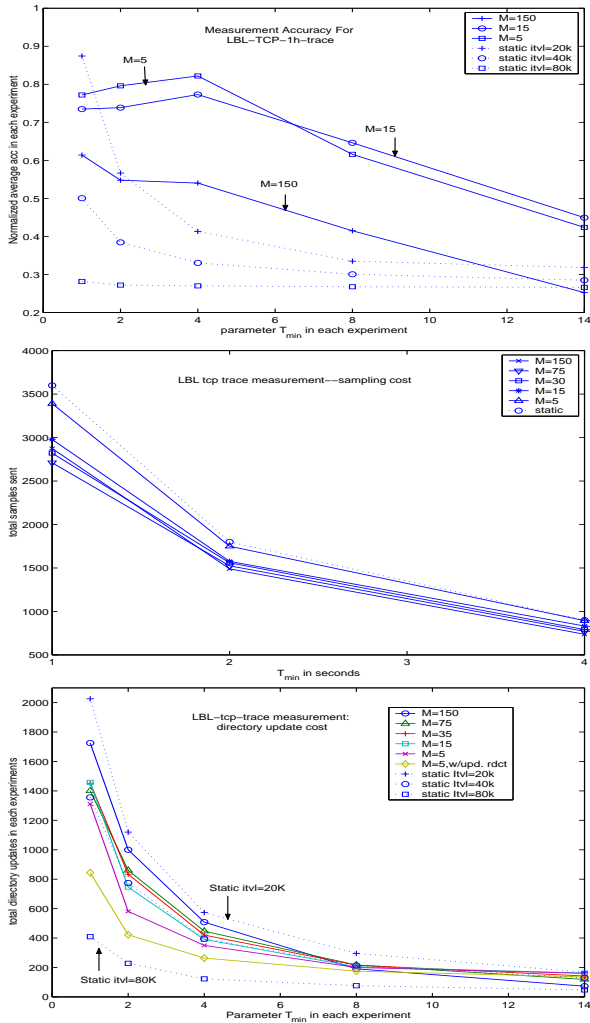


Figure 2: *Performance with TCP-WAN traffic.* Top: The accuracy plot with varying T_{min} and M , compared to the static algorithm with equivalent range sizes; Middle, Sampling overhead; Bottom, The directory update overhead

of our adaptive measurement is very satisfactory both in terms of accuracy and cost overhead. Figure 2 shows the results. We chose static range sizes to match the average range sizes in adaptive measurements to ensure a fair comparison (note that in the static range based algorithm, we can provide higher accuracy with smaller range sizes at the cost of much higher overhead).

From Figure 2, we report a significant accuracy improvement (100 percent to 200 percent) of our adaptive range based algorithm over the static range based algorithm. In addition, the overhead costs both in terms of sampling cost and directory updates are reduced. For the adaptive measurement, a smaller history table size (smaller sliding window) leads to better accuracy when smaller sam-

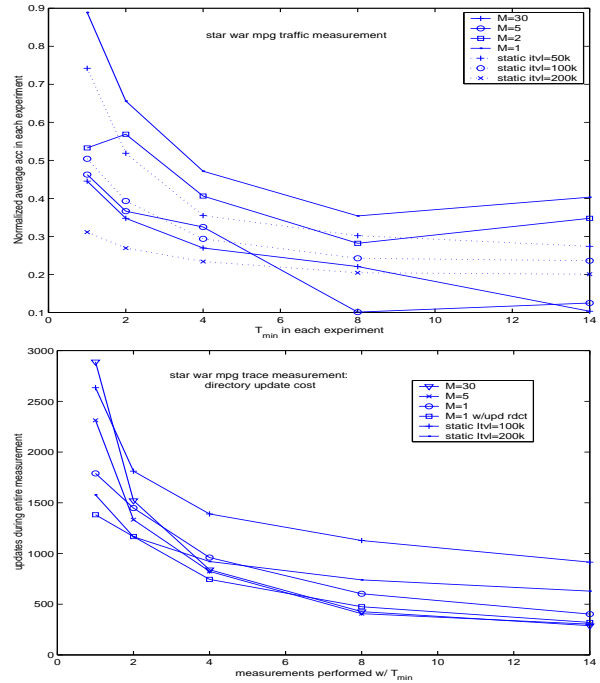


Figure 3: *Performance with MPEG traffic.*

pling intervals (higher sampling rates) are allowed. With an increased sampling intervals (lower sampling rates), measurements with larger history table (larger sliding window) have slightly better accuracy because of the bigger range size. As expected, with the static algorithm, a smaller interval yielded higher accuracy and caused increased directory update overheads. Note that the sampling interval remains unaltered for different interval sizes of the static algorithm.

MPEG traffic measurement: The single (video) mpeg trace is relatively more bursty than the tcp trace. Even though the prediction model we are able to derive has only a modest fit, we expect that the collection algorithms will compensate for inaccuracy in prediction by identifying steady states and suppressing bursts. Again, we choose the static range size in accordance to the average size of adaptive measurements.

As expected, from Figure 3 we found that the improvements in accuracy with the adaptive algorithm over the static algorithm are not as dramatic as in the TCP-WAN trace. With an extremely small memory size, $M = 1, 2$, we observe 50 percent to 100 percent accuracy improvements over the static algorithm with comparable cost overheads. When M is extremely small, the next step prediction always anticipates exactly the same traffic state as the current one, quickly reducing the adaptive range in relatively steady states. Whenever a burst is identified through the probing process, the algorithm's burstiness meter $\hat{\sigma}_a/\sigma_a$ rises directly, causing the sampling interval T_a to

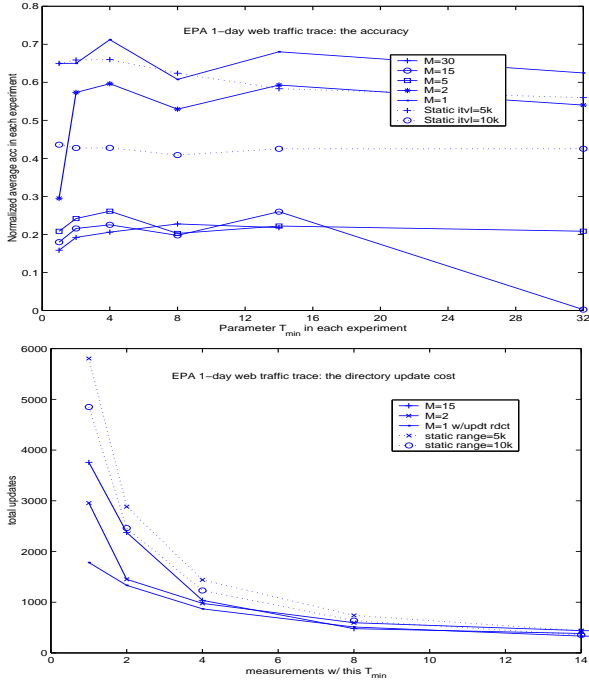


Figure 4: Performance with web traffic.

increase, preventing oscillation during the bursts. This process proactively catches the steady states and simply blocks the influences from bursty states, therefore yielding a better overall performance.

Web traffic measurement: The 1 day trace of the EPA web server is extremely bursty, especially during less busy hours when the load changes from 0 to a few hundred kilobytes per second easily. Although 90 percent of the values are well below average, the tail distribution can give rise to extremely large values. The ARMA prediction model isn't satisfactory due to its large estimation residual variation. We include this traffic to test the robustness of the proposed algorithm when the source traffic isn't stationary. Similar to the tcp and mpeg trace measurements, the 5K and 10K static range sizes are chosen according to the average range size of the adaptive measurements.

Figure 4 shows that the adaptive algorithms with small values of M shows slightly higher accuracy levels as compared to the static algorithms, however the directory update costs of the adaptive algorithm is much smaller than the static. With minimal prediction ability from the estimated model, the adaptive algorithm accommodates the enhanced dynamics by manipulating the range size to ensure reduced directory update overheads. In large scale distributed measurement infrastructures, the overhead of directory updates can be critical since frequent updates cause the directory service to become a bottleneck, reducing the performance of the system.

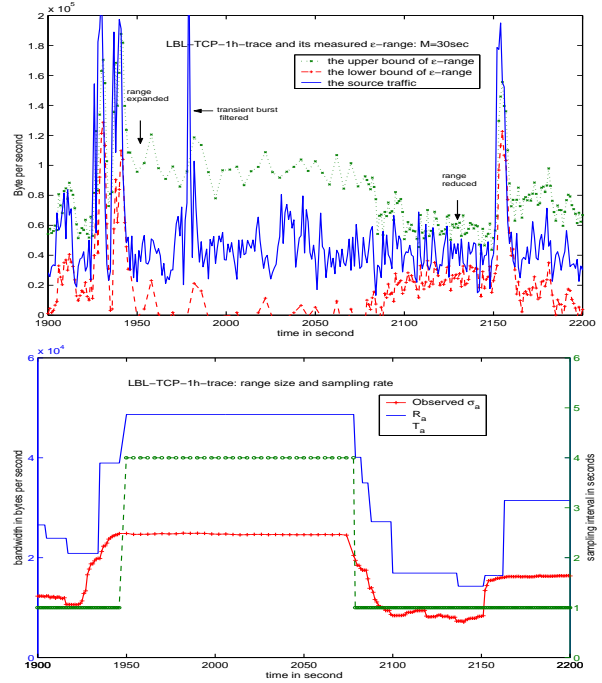


Figure 5: Measurement using long memory ($M=150$ sec), Top: Source traffic and estimated ε -range; Bottom: Short-term burstiness level σ_a and its influence on range size and sampling interval

4.2.2 The impact of M (sliding window size)

Figures 5 and Fig 6 show a 5 minute clip obtained from the entire TCP traffic trace and illustrates the impact of history table (moving window) sizes at $M = 150$ and $M = 15$ respectively. From the first graph (LHS of Figure 5), we observe that the next step prediction $\hat{Z}_t(1)$, which is in the middle of the two bounds (upper and lower), generally follows the source traffic. The range size expands upon arrival of bursts and reduces when it stays steady for for a while. However, due to the relatively large window size ($M = 150$), the changes observed were not dramatic, because a large window size records history information from past bursts longer (i.e. long memory). The impact of the long memory (large moving window) can be clearly observed in the second graph (RHS of Figure 5) where the rise and fall of σ_a takes about 150 seconds. This can be compared to Figure 6, where a much smaller history table is used ($M = 15$), and the same burst caused σ_a to fluctuate much more quickly.

From the graphs in Figure 5, we also observe that large transient bursts are filtered due to the averaging effect caused by the large number of history values. Furthermore, we notice that the sampling rate increases upon range tightening and decreases upon range relaxation. In Fig 6 the calcu-

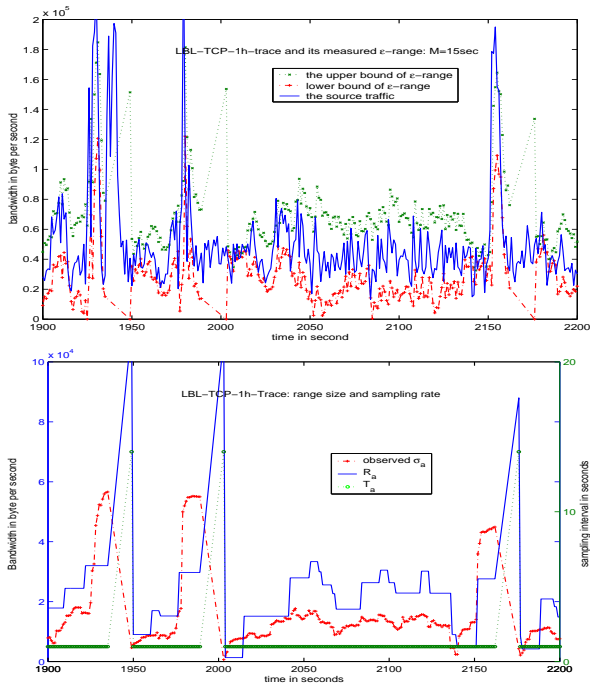


Figure 6: Measurement using shorter memory ($M=15\text{sec}$)

lated range is much smaller on average. This is due to the fact that the source traffic is heavy tailed in burst distribution, with relatively steady behavior most of the time, with a few extremely bursty states. With a smaller history table, bursts slide out of the memory (moving window) quickly, resulting in smaller values of $\hat{\sigma}_a$ most of the time. We observe similar results (i.e. impact of sliding window size) with the MPEG and Web traces as well and do not discuss them in this section due to space limitations.

4.3 Performance Summary

Since network traffic can be highly unpredictable, the design goal of our algorithm is to (a) tradeoff accuracy for cost savings during bursty states and (b) provide better accuracy during relatively steady states. For most of the traffic traces studied, the proposed adaptive range based algorithm significantly improves the measurement accuracy with a much smaller overhead cost compared with static range based algorithms. We also found that setting the sliding window to a smaller size generally improves the performance for realistic network traffics. This is because as the traffic gets more and more unpredictable, quick response to traffic state changes becomes more important than better predictions. We also noticed that with less stationary traffic, the adaptive range based algorithm gracefully reduces accuracy while providing higher cost-efficiency. Such robustness is especially desirable in large scale network systems.

5 Related Work and Future Research Directions

Time series analysis has been used frequently to model network traffic. The modeling of modern network and telecommunication traffic using heavy tails have been studied [4], and several methods are proposed including a time series approach, which uses the classical *Yule-Walker estimation* to obtain the heavy tailed autoregressions. The ARMA model is used for network traffic prediction [7] to show how multiplexing can be used to improve traffic predictability. In [6], the memory (moving window) size is studied. By illustrating the existence of a persistent critical time scale, an optimal memory size is derived, so that the concerned overflow probability P_f , the probability that actual arrivals exceed the estimated arrivals, is minimized. This memory size can be conservative for our purposes since our goal is to maximize the overall accuracy of the estimation while minimize the sampling and directory update cost overhead.

A lot of measurement architectures have proposed to determine traffic parameters [12, 2, 13, 1, 9] for network traffic, for instance [12] develops methods to estimate the distance between any two points in the internet. [2] implements a linear model based information collecting and prediction infrastructure, and defined service APIs for upper level applications which need such support in establishing network connections. Placement strategies for monitoring modules are proposed in [13]. Here, the efficiency of the sampling process is enhanced by emphasizing *hot points* in typical ISP network topology. In the diffserv architecture [17], state information collected in the directory is used by the bandwidth broker (BB) to statistically guarantee the negotiated QoS agreement. Possibly, the directories of adjacent domain can exchange aggregated state information to maintain a more accurate snapshot of the system. We are also currently focusing on developing additional notions of soft state [23] suitable for QoS provisioning.

QoS aware routing and resource scheduling rely on efficient traffic measurement. A direct application of our range based measurement technique is the QoS routing protocol proposed in [16], where link-level parameters such as delay, available bandwidth, etc, are modeled as probability distributions. Here, routing decisions are based on a *most probable* path that satisfies the requested QoS. The static range based algorithm used in this paper is proposed in [3] where QoS routing performance is studied in detail using different static range sizes and sampling intervals. The study determined that smaller range sizes perform better when sampling occurs at short intervals, while larger range sizes are more cost-effective with longer sampling intervals. Moving object databases deal with the modeling and tracking of a moving object within a database; information

collection solutions have been proposed in this context that directly compare the cost of information imprecision with the cost of message passing and make decisions based on the difference [22].

In this paper, we established an efficient information collecting process and proposed an adaptive range based measurement algorithm to measure the network traffic load. The performance evaluation indicates that the proposed technique improves the cost-effectiveness significantly over the existing methods under varying network traffic patterns and conditions. Furthermore, our algorithm is highly robust in network traffic conditions for which prediction models are difficult to estimate. We are currently studying the implications of the proposed adaptive parameter collection techniques in the performance of QoS provisioning protocols, e.g to perform combined path and server selection, which allows load balancing not only between replicated servers but also among network links to maximize the request success ratio and system throughput.

This paper uses a time series approach for network traffic analysis. Currently, we are studying a more *structured* way in which the protocol information contained in the network traffic is exploited to assist the measurement. We are also studying how the sliding window size M should be changed according to the knowledge of scaling factor, A simple policy is then able to adjust M according to this estimation.

References

- [1] N.Miller and P.Steenkiste *Collecting Network Status Information for Network-Aware Applications* INFOCOM'99
- [2] P.A.Dinda and D.R.O'Hallaron *An Extensible Toolkit for Resource Prediction In Distributed Systems*, Technical Report, CMU-CS-99-138
- [3] G.Apostolopoulos, R.Guerin, S.Kamat and S.K.Tripathi *Quality of Service Based Routing: A Performance Perspective*, ACM SIGCOM'98
- [4] S.R.Resnick *Heavy Tail Modeling and Teletraffic Data*, The Annals of Statistics, 25(5),pp.1805-1869, 1997
- [5] W.Willinger, V.Paxson and M.S.Taqqu, *Self-similarity and heavy tails: Structural modeling of network traffic*. A Practical Guide to Heavy Tails: Statistical Techniques for Analyzing Heavy Tailed Distributions (eds. R.Adler, R.Feldman and M.S.Taqqu), Birkhauser Verlag, Boston
- [6] M.Grossglauser and D.N.C.Tse, *A Framework for Robust Measurement-Based Admission Control*, IEEE/ACM Transaction on Networking, VOL. 7, No. 3, JUNE 1999
- [7] A.Sang and S.Q.Li *A Predictability Analysis of Network Traffic*, INFOCOM, 2000
- [8] P.A.Dinda and D.R.O'Hallaron, *Host Load Prediction Using Linear Models*, Cluster Computing, To Appear
- [9] Mark Stemm and Randy Katz and Srinivasan Seshan, *A Network Measurement Architecture for Adaptive Applications*, Infocom, 2000
- [10] V.Paxson and Sally Floyd, *Wide-Area Traffic: The Failure of Poisson Modeling*, SIGCOM 1994
- [11] A.Adas, *Using adaptive Linear Prediction to Support Real-Time VBR Video Under RCBR Network Service Model*, IEEE/ACM Transaction on Networking, Vol.6, No.5, Oct.1998,pp.635-644
- [12] P.Francis, Sugih Jamin, Vern Paxson, Lixia Zhang, D.Gryniewicz and Y.Jin, *An Architecture for a Global Internet Host Distance Estimation Service*, INFOCOM 1999
- [13] *GTE Internet Access Services: Managed Hosting with Traffic Distributor*, <http://www.bbn.com/services/hosting/traffic>, 1999
- [14] Z.Fu, N. Venkatasubramania, *Combined Path and Server Selection In Dynamic Multimedia Environments*, ACM SIG Multimedia 1999
- [15] N. Venkatasubramanian and S.Ramanathan, *Load Management for Distributed Video Servers*, ICDCS'97
- [16] D.H.Lorenz and A.Orda, *QoS Routing in Networks with Uncertain Parameters*, In Proceedings of INFOCOM'98
- [17] Z.L.Zhang, Z.Duan, L.Gao and Y.T.Hou, *Decoupling QoS Control from Core Routers: A Novel Bandwidth Broker Architecture for Scalable Support of Guaranteed Services*, SIGCOM 2000
- [18] V.Paxson and S.Floyd, *Wide-Area Traffic: The Failure of Poisson Modeling*, IEEE/ACM Transactions on Networking, 3(3), pp.226-244, June 1995
- [19] M.W.Garrett, W.Willinger, *Analysis, Modeling and Generation of self-similar VBR Video Traffic*, SIGCOM'94
- [20] L. Bottomley, *EPA-HTTP Single WWW Server Trace*, available at '<http://ita.ee.lbl.gov/html/contrib/EPA-HTTP.html>'
- [21] G.E.P.Box, G.M.Jenkins and G.Reinsel, *Time Series Analysis: Forecasting and Control*, 3rd ed. Prentice Hall, 1994
- [22] O.Wolfson, S.Chamberlain, S.Dao, L.Jiang,G.Mendez, *Cost and Imprecision in Modeling the Position of Moving Objects*, In Proceedings of International Conference on Data Engineering, 1998
- [23] S.Raman and S.McCanne, *A Model, Analysis, and Protocol Framework for Soft State-Based Communication*, SIGCOM'99
- [24] Z.Fu and N.Venkatasubramanian, *Adaptive Parameter Collection in Dynamic Distributed Environments*, ICS- Technical Report, UC Irvine, 2001