Journal of Grid Computing manuscript No.
(will be inserted by the editor)

Time-series Forecast Modeling on High-Bandwidth Wide Area Network Measurements

Wucherl Yoo · Alex Sim

Received: date / Accepted: date

Abstract With the increasing number of geographically distributed scientific collaborations and the growing sizes of scientific data, it has become challenging for users to achieve the best possible network performance on a shared network. We have developed a model to forecast expected bandwidth utilization on high-bandwidth wide area networks. The forecast model can improve the efficiency of the resource utilization and scheduling of data movements on high-bandwidth networks to accommodate ever increasing data volume for large-scale scientific data applications. A univariate time-series forecast model is developed with STL and ARIMA on SNMP path utilization measurement data. Compared with traditional approach such as Box-Jenkins methodology to train the ARIMA model, our forecast model reduces computation time by 78.1%. It also shows resilience against abrupt network usage changes. The forecast errors are within the standard deviation of the monitored measurements.

Keywords Data modeling, time series, prediction model, network measurements, network traffic

1 Introduction

With performance advances in large scale experiments and simulations, the data volume of scientific applications has rapidly grown. Even with advances in network technology, it has become more challenging to efficiently coordinate network resources and to achieve the best possible network performance on a shared network. It is also challenging to build a forecast model for network bandwidth utilization with accurate and fine-grained prediction due to the computational complexities. To support efficient resource management and scheduling for ever increasing data volume in extreme-scale scientific applications, we have developed

Lawrence Berkeley National Laboratory
1 Cyclotron Road, Berkeley, CA, USA, 94720
E-mail: {wyoo, asim}@lbl.gov

an analytical model in order to characterize and forecast[1] bandwidth utilization on a high-bandwidth wide area network (WAN), focusing on the traffic for large-scale scientific data movement.

The forecast model can improve the efficiency of network bandwidth resource utilization, and it can help efficient data transfer scheduling and path finding. The goal of this paper is to model the network bandwidth utilization between two sites to support data flow timing, parameter decisions, and network topology or link planning. Our modeling efforts can help systematic data transfer decisions without over/under-provision. One of our previous works proposed a network reservation framework to provide guaranteed bandwidth on ESnet [5]. Our forecast model can complement this type of reservation system, Software Defined Network (SDN) [14], or a system to select alternate paths for large data transfers.

The model needs to be computationally efficient and comparably accurate in order to forecast multiple paths of users' interests. We select a size of an appropriate training set that shows relatively small forecast error for accuracy with manageable computational overhead. In addition, we have studied the effect of variability of the bandwidth usage on the forecast accuracy and mechanisms to make our model resilient against the abrupt usage changes.

The experimental data on Simple Network Management Protocol (SNMP) link utilization has been collected by ESnet [1] in 2013 and 2014 on each router. Our experiments use SNMP data from 6 directional paths connecting a pair of large data facilities described in Sec. 4.1. The SNMP data consists of the size of the bandwidth utilization and time-scale at 30 seconds interval. The maximum size of the bandwidth utilization is calculated for each time interval from the routers in each path, which represents bandwidth utilization for each path. It is well known that Internet traffic has cyclic self-similarity in daily interval. In Sec. 4.2, we also show the daily seasonality is present in the SNMP data.

We have developed the forecast model as a univariate time series model. The first step is to remove the seasonality in the measurement data. We use the Seasonal decomposition of Time series by Loess (STL) [9] in order for this seasonality removal. The STL decomposes the SNMP data into the time series of seasonality, trend, and remainder. After deducting seasonality component, we use the AutoRegressive Integrated Moving Average (ARIMA) on the seasonally adjusted time series. The orders of the ARIMA model are selected in an automated mechanism based on the assumption of stationary time series about the SNMP data. We have observed that there are no significant changes in the average bandwidth utilization in the training dataset window (up to 8 weeks) throughout 2013 and 2014. In Sec. 4.3, we show that our stationary assumption is appropriate for the SNMP data. Our forecast model reduces the computation time for forecast by 78.1% compared to the traditional approach such as Box-Jenkins methodology [7][8], to find the best fitted forecast model using ARIMA. In addition, our model shows resilience against abrupt network usage changes.

The rest of paper is organized as follows. Sec. 2 presents the related work. Sec. 3 demonstrates the model design and implementation. Sec. 4 presents experimental evaluations of the forecast model, and the conclusion is in Sec. 5.

[1] We explicitly make a distinction between forecast and prediction. We use forecast as an estimation of *future* values based on the analytical model built from the past observations. On the other hand, we use prediction as an estimation of values based on an analytical model.

2 Related Work

The self-similarity of historical network measurements has been studied in LAN [23], WAN [28], and the Internet [12]. It allows to use past history to forecast near-term future network traffic. Qiao et al. [30] presented an empirical study of the forecast error on different time-scales, showing that the forecast error does not monotonically decrease with smoothing for larger time-scale.

Benson el al. [6] studied network traffic patterns in data centers using SNMP data. Yin el al. [36] proposed a mechanism to predict application-layer data throughput. Balman et al. [5] proposed a network reservation framework to provide guaranteed bandwidth. Our forecast model complements these works by providing traffic forecast information.

Available bandwidth can be estimated by sending probe packets as proposed from measurement tools: Pathload [20], pathChirp [31], IGI [18], and Spruce [35]. Shriram et al. [34] conducted a comparison study on the estimation of available bandwidth from various measurement tools in network simulator (ns2) [2]. Croce et al. [11] proposed bandwidth estimation techniques from large-scale distributed systems. Aceto et al. [3] proposed an end-to-end available bandwidth measurement infrastructure. Our model focuses on the forecast of the available bandwidth using passive measurements from routers instead of estimations from probing packets.

Several prediction models of TCP data transfers have been proposed. Throughput prediction models were proposed for large TCP transfers [16][25]. Mirza et al. [26] used a machine learning mechanism to predict TCP throughput. While these works are restricted to predict TCP data transfers, our model forecasts on aggregated throughput for a network path.

Several models have been proposed to forecast network traffic. Sang et al. [32] proposed a short-term (a few minutes) forecast model using ARMA with 1 second time-scale data. Papagiannaki et al. [27] proposed a long-term (1 year) forecast model of Internet backbone traffic using ARIMA with 1 week time-scale data. Krithikaivasan et al. [21] proposed a mid-term (1 day) forecast model using ARCH model with 15 minute time-scale data. Our model focuses on mid-term (1 day) forecast of the bandwidth utilization using 30 second time-scale data. The number of forecast points is orders of magnitude more than the aforementioned models due to the smaller time-scale. Therefore, our forecast model requires more computation for the accuracy than these proposed models. Our model overcomes these challenges by seasonal adjustment and stationary assumption, which are not discussed in the previous models.

3 Model Development

We have developed the forecast model as a univariate time series model. A forecast model estimates the future values using the observed SNMP data up to time n $(x_1, x_2, \cdots, x_n)$. The forecast of h steps ahead is denoted as $\hat{x}_n(h)$ at time $n+h$. When the observed value (x_{n+h}) is available at time $n+h$, we calculate the forecast error denoting $e_n(h)$ as:

$$e_n(h) = x_{n+h} - \hat{x}_n(h) \tag{1}$$

3.1 Logit Transformation

The theoretical maximum value of the possible size of the SNMP is 10^{10} bits, and the minimum value is 0 bit within one second on the current 100 Gbps ESnet network. As the SNMP data is collected every 30 second, we normalize the traffic size by dividing by 30. Logit transformation is applied to the SNMP data x to set the lower and upper bounds based the limits of possible values. Time series data x containing n observations is transformed to time series data y with lower bound a and upper bound b (10^{10} bit/s) as denoted as Eq. 2. The lower bound a is approximated to 1 bit/s instead of 0 bit/s. While there are very few observed cases when no transfer occurs, this lower bound approximation can be ignored in the 100Gbps networks.

$$\begin{aligned} x &= \text{time series } x_t = x_1, x_2 \cdots, x_n \\ y &= \text{time series } y_t = y_1, y_2 \cdots, y_n \\ y &= \text{logit}(x) = \log(\frac{x-a}{b-x}) \end{aligned} \tag{2}$$

3.2 Seasonal Adjustment

After the logit transformation defined in Eq. 2, the transformed SNMP data y is seasonally adjusted. Removing seasonal components from the time series allows the analysis of the non-seasonal trend. This is essential to project the trend and the seasonality of the past history to the future values. We use the Seasonal Decomposition of Time Series by Loess (STL) [9] for this seasonal adjustment. The STL decomposes the logit transformed SNMP data into the components of the seasonality S, the trend T, and the remainder R as denoted as Eq. 3.

$$y = y_t = S_t + T_t + R_t \tag{3}$$

The STL applies a sequence of smoothing from Loess (Locally Weighted Regression Fitting) [10]. This smoothing sequence progressively refines and improves the estimates of the seasonal and trend components. There exist several parameters to derive the STL model. The seasonal cycle is evaluated with the possible choices such as minute, hour, day, and week. The smoothing windows for the seasonality (n_s) for trend (n_t) are evaluated with different values. After the decomposition, we seasonally adjust SNMP data by deducting seasonality component denoted as $y\prime = y_t\prime = y_t - S_t = T_t + R_t$.

3.3 Bandwidth Utilization Forecast

The forecast model is developed by using the AutoRegressive Integrated Moving Average (ARIMA) on the seasonally adjusted time series, $y\prime$. The ARIMA model consists of the orders of the autoregressive process (p), the differences (d), and the moving average (q). They are selected in an automated mechanism as follows. First, the stationarity of the time series is confirmed by the Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test [22]. When the stationarity is confirmed, d is selected as 0. Otherwise, d is selected as 1, which is sufficient to make the non-stationary time

series to be stationary in our model. We use the Akaike's Information Criterion (AIC) [4] to automatically select the modeling parameters as shown in the Box-Jenkins methodology [7][8]. The AIC represents the sum of the maximum log likelihood for the estimation and the penalty from the orders of selected model. This combination allows simpler models with less number of orders unless they show severely low likelihood for the estimation. We calculate the AIC with different combinations of p and q incrementing from 1 until the sum of p and q reaches to a certain maximum value. The model choice from the AIC converges, and is asymptotically equivalent to that of the cross-validation [33]. The best model with p and q is chosen with the least value of the AIC.[2] In our case, the maximum sum of p and q is 10, which is the smallest sum that can result in reasonably accurate forecast from the experiments.

After the orders of the ARIMA model are selected, we fit the model with the seasonally adjusted time series $(y_1\prime, y_2\prime, \cdots, y_n\prime)$ and with the training set of n observed data $(x_1, x_2, \cdots, x_n)$. The ARIMA model fitting is to estimate the parameters with the orders of autoregressive process and moving average process (after the orders of differencing if $d > 0$). The forecast of h time steps ahead is computed from the fitted model $(\hat{y}_h\prime)$. Then, the seasonality component is added to these forecast values $(\hat{y}_h)$ as in Eq. 4. The seasonality forecast $(\hat{S}_{n+1}, \hat{S}_{n+2}, \cdots, \hat{S}_{n+h})$ can be estimated by simply repeating the decomposed seasonal component $(S_1, S_2, \cdots, S_n)$.

$$\hat{y}_h = \hat{y}_h\prime + \hat{S}_{n+h} \tag{4}$$

Then, these forecast values are converted to the original scale using the reverse logit transformation as in Eq. 5.

$$\hat{x}_h = (b-a) \cdot \frac{\exp(\hat{y}_h)}{1+\exp(\hat{y}_h)} + a \tag{5}$$

We evaluate the forecast error by the cross-validation mechanism for time series data proposed by Hijorth [17]. The original mechanism by Hijorth computes a weighted sum of one-step-ahead forecasts by rolling the origin when more data is available. Similarly, we compute the average forecast error for 1 week by forecasting one target day ($h = 1, \cdots, 2880$) and rolling 6 more days. We compare this cross-validation results of the forecast errors as Root Mean Squared Error (RMSE) in Sec. 4.

4 Experimental Results

4.1 Experimental Setup

Table 1 describes 6 directional paths used in the experiments. They connect two sites on the ESnet in the US. The paths consist of 6 or 7 links connected with the routers between NERSC in CA, ORNL in TN, and ANL in IL. PID is the path identification: `P1` and `P2`, paths between NERSC and ORNL, `P3` and `P4`, paths between NERSC and ANL, and `P5` and `P6`, paths between ORNL and ANL.

[2] The AIC is combined with the positive value of penalty from the orders and negative log-likelihood.

The measured SNMP data represents the bandwidth utilization during a 30 second interval at routers in each path. We selected the maximum value on a link in each path for an aggregated measurement. The experiments were conducted on a machine with 8-core CPU (AMD Opteron 6128) and 64 GB memory. To reduce overall execution time, we parallelized the computational tasks of parameter searching, fitting, and calculating the forecast error.

The granularity of the SNMP data can be decreased by 30 second time unit into larger scales and aggregating the traffic size, e.g., aggregating and normalizing the traffic into 1 minute, 10 minutes, 30 minute, 1 hour, or 1 day time unit. As the decreased granularity of network traffic results in reducing the variances of the traffic, it can show less forecast error [30]. It also leads to less computation time due to the decreased data size with lower granularity. Our experiments showed the less forecast errors with the decreased granularity of the SNMP data.[3]

Table 1: Description of Paths.

PID	Source	Destination	# of Links
P1	$NERSC$	ANL	7
P2	ANL	$NERSC$	7
P3	$NERSC$	$ORNL$	7
P4	$ORNL$	$NERSC$	7
P5	ANL	$ORNL$	6
P6	$ORNL$	ANL	6

Fig. 1 shows the SNMP data used as a test dataset that are measured from July 21 to July 27, 2014. The durations for the training dataset are from various weeks prior to July. 21, 2014.[4] RMSE[5] is in bits/s, and is calculated with $RMSE(h) = sqrt(\frac{1}{h} \cdot \sum_{i=1}^{h} (e_n(i))^2)$. After forecasting the first day of the test set $(\hat{x}_1, \cdots, \hat{x}_{2880})$, the forecast error for the first target day $RMSE(h_{day1}) = RMSE(h)$ was computed. The forecast error for the second target day $RMSE(h_{day2}) = RMSE(h + h)$ was computed by adding the observations from the first target day to the previous training set $(x_1, \cdots, x_n, \mathbf{x_{n+1}}, \cdots, \mathbf{x_{n+h}})$. This process was repeated for the next 5 target days to compute the average of RMSE for the one-week test set.

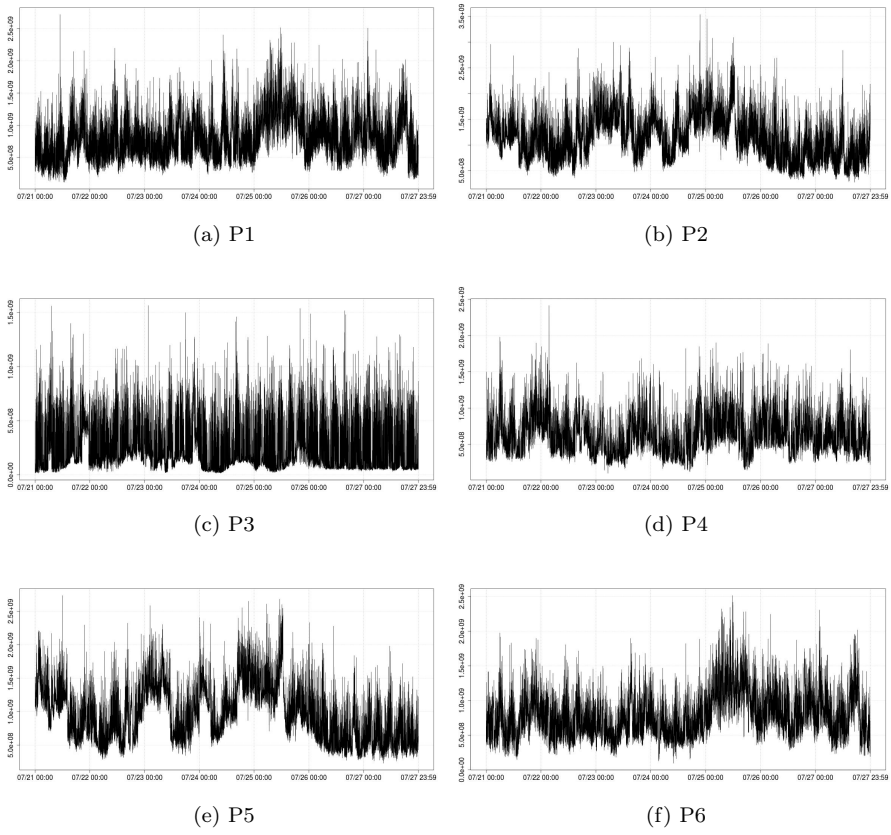
4.2 Seasonality Analysis

Fig. 2 shows the seasonally decomposed SNMP data using the STL. The STL model was derived by using the parameters described in Sec. 3.2. The seasonal cycle was evaluated with possible cyclic periods such as minute, hour, day, and week.

[3] This paper does not include the results from the decreased granularity.

[4] The time and date are in Greenwich Mean Time (GMT) in this paper.

[5] As the SNMP data is collected in every 30 second interval, the size of bandwidth utilization is normalized by dividing by 30. Note that RMSE is in the same order of the bandwidth. MAE and ME are also shown in Table 2.



(a) P1 (b) P2

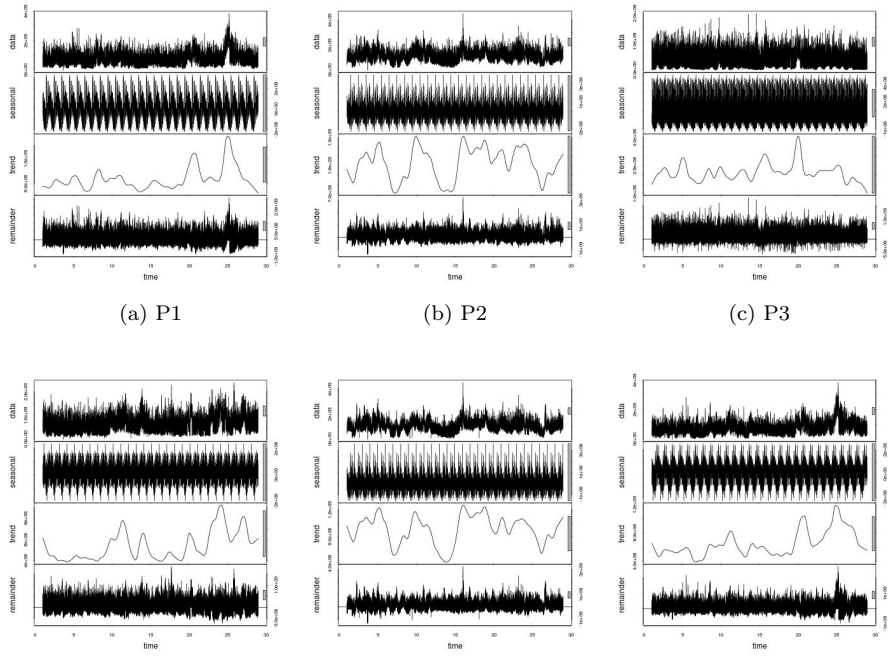
(c) P3 (d) P4

(e) P5 (f) P6

Fig. 1: Bandwidth Utilization Graphs for Experimental Paths: The size of traffic is shown in vertical axis as bit/s. The horizontal axis shows the time from July 21, 00:00:00, 2014 to July 27, 23:59:30, 2014.

The smoothing parameter for the seasonality (n_s) was evaluated with possible values such as the same value with n_p or multiples or inverse multiples of n_p. The smoothing parameter for trend (n_t) was also evaluated with multiple values. With larger n_t, the Interquartile Range (IQR) of the trend component got smaller. This is because smoothing from Loess [10] of the trend component gets smoother with larger n_t, and this result increases the IQR of the remainder component.

Different values of seasonality smoothing window (n_s) showed the similar forecast accuracy. The IQR of the seasonal component did not change with different n_s. In addition, trend smoothing window (n_t) changed the shape of trend, but did not change the forecast accuracy. As a result, we selected n_s and n_t as the same as n_s. While the shape and the IQR were changed with different n_t, the forecast error was still similar. This suggests that the ARIMA is more crucial component than STL in our forecast modeling. However, fitting with STL is essential since it removes seasonal component from the original time series. Using the Seasonal ARIMA or the ARIMA without STL appeared to be another possible



(a) P1 (b) P2 (c) P3

(d) P4 (e) P5 (f) P6

Fig. 2: Seasonally Decomposed Components: The top plot in each graph is from the raw SNMP measurement data. The second plot is for the seasonal component. The third plot is for the trend component. The bottom plot is for the remainder. The horizontal axis shows the time as days, and the duration is 4 weeks from June 24, 00:00:00, GMT 2014 to July. 20, 23:59:30, GMT 2014.

choice, however computation time of the modeling these choices took too long to conduct the experiments. Only after seasonal adjustment, the computation time of the ARIMA modeling was viable.

Fig. 3 shows the forecast errors when using different seasonal cycles. It is well known that Internet traffic has cyclic self-similarity in daily interval [13]. The average forecast error (RMSE) with daily seasonality was 4.9% better than that of weekly seasonality and 2.8% better than that of hourly seasonality. This result shows that the SNMP data of the scientific data traffic has stronger daily self-similarity than hourly or weekly periods, similar to the Internet traffic. The average values of Hurst parameters [12] from `P1` to `P6` were 0.92, 0.94, 0.93, 0.89, 0.94, and 0.87 respectively, which confirm the self-similarity. The remainder of STL decomposition did not pass the Ljung-Box test [24], which means that autocorrelation still exists. Therefore, we used ARIMA to remove existing autocorrelation from the seasonally adjusted time series.

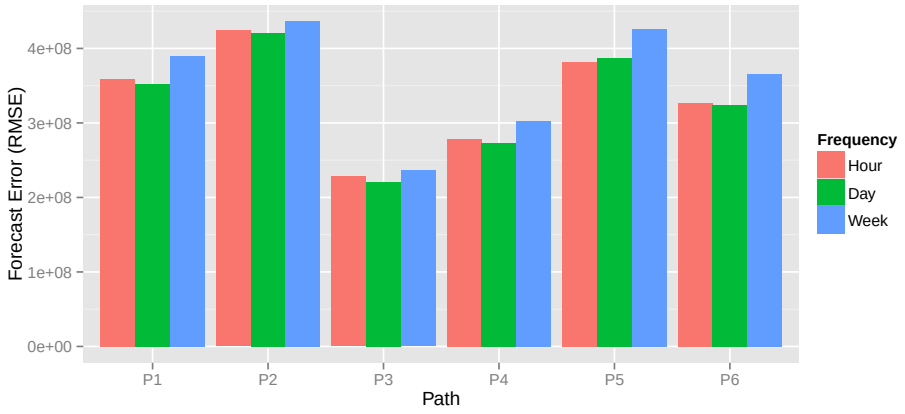


Fig. 3: Forecast Error Comparison with Different Seasonal Cycles: The size of the traffic is shown in vertical axis as bit/s. The training set size is 4 weeks ($n = 80640$). The number of observations per seasonal cycle is one hour, one day, and one week.

4.3 Bandwidth Utilization Forecast

We compared possible modeling choices including parameter selections. The model was developed based on the Box-Jenkins methodology [7], using ARIMA on seasonally adjusted SNMP data using the STL. The orders of the ARIMA model (p,d,q) were selected in the automated mechanism in Sec. 3.3. After fitting the forecast model with the selected parameters, Ljung-Box test was conducted to check whether the overall residuals are similar to the white noise, and whether the residuals of the forecast model passed the test.

We tested the possible forecast methods on seasonally adjusted time series data. Fig. 4 illustrates the comparison of forecast errors for different forecast models: the ARIMA, the Exponential smoothing state space model (ETS) [19], and the Random Walk (RW) [7]. The forecast error of the ARIMA is the lowest, which led us to use the ARIMA in the forecast model.

Furthermore, we applied Hampel filter [29] to evaluate whether removing outliers helps the forecast accuracy. Hampel filter is a moving window nonlinear data cleaning filter that can remove outliers based on Hampel identifier [15]. Outliers were removed with t-value above 3 or -3, based on 3-sigma rule [29] and moving window length of 6 hours. We observed that these parameters were sufficient to remove the most of outliers from the SNMP data measured in 2013 and 2014. The forecast error is slightly improved, but it is very marginal. Therefore, we decided not to use Hampel filter in our forecast model.

4.4 Training Set Size

Intuitively, there is a tradeoff between less computational requirement with smaller training set size and better accuracy with larger training set size. Smaller training

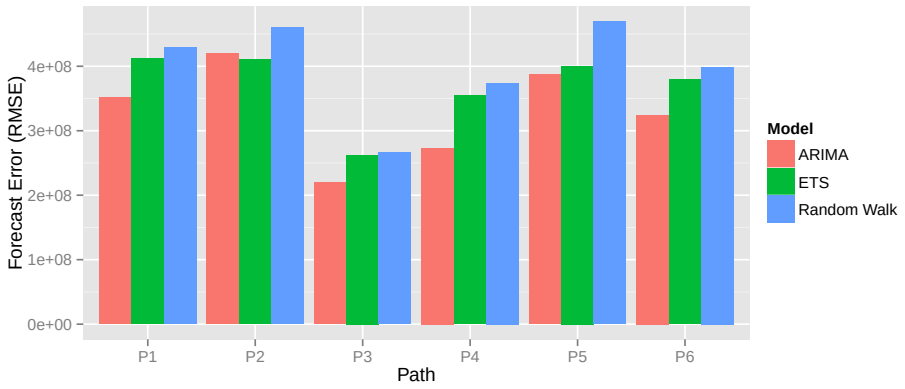


Fig. 4: Forecast Error Comparison for Different Forecast Models on Seasonally Adjusted Data: The training set duration is 4 weeks. The seasonal cycle is one day.

set size certainly requires less CPU time and less storage including memory and disk. Larger training set size does not always guarantee better forecast accuracy because adding more historical data to the forecast modeling may not improve forecast accuracy due to possible deviances on the older history from the recent history. In addition, the past history has lesser degree of impact than the recent history on the forecast models such as the ARIMA. To find the best possible training set size, we have studied the effect of different sizes of data on forecast accuracy. Fig. 5 shows the forecast errors for different sizes of training sets. We also tested training set sizes from 4 to 52 weeks. Although the forecast accuracy was the best with 20 weeks, this was marginally better than other training set sizes. Since smaller training set requires less computational resources, we used 4 weeks of training set size in the following experiments. We also observed that increasing training set size more than 20 weeks makes the forecast accuracy consistently worse. This shows that increasing the training set size does not guarantee better forecast accuracy. This training set size was also effective in the delayed model update, shown in the next Section (Sec. 4.5).

4.5 Delayed Model Update

We observed that even when KPSS test [22] did not confirm the stationarity, the time series did not drift significantly. Thus, we evaluated whether the stationary assumption of SNMP data was appropriate even when the KPSS test result suggested non-stationary. We observed that the variances in the training sets and the sudden bandwidth utilization changes made the test results inaccurate in some cases.

As illustrated in Fig. 6, the stationary assumption results the forecast error (RMSE) 1.6% less than that of forecast without the assumption. The forecast

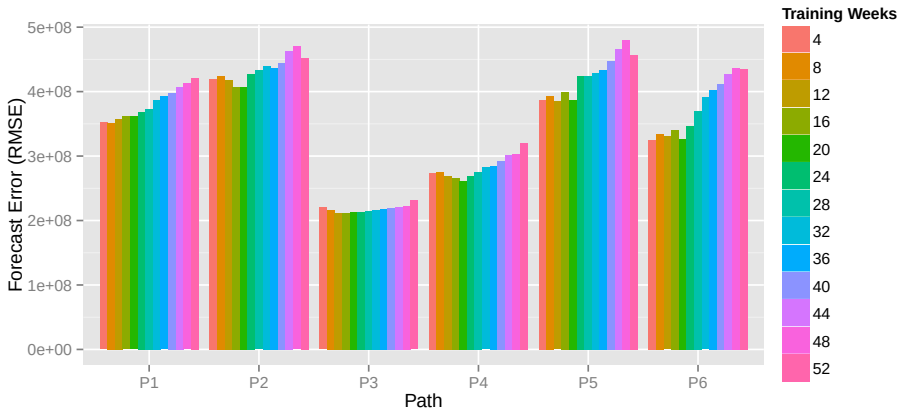


Fig. 5: Forecast Error Comparison for Different Training Set Sizes: The training set sizes are from 4 to 52 weeks. The number of observations per seasonal cycle is one day.

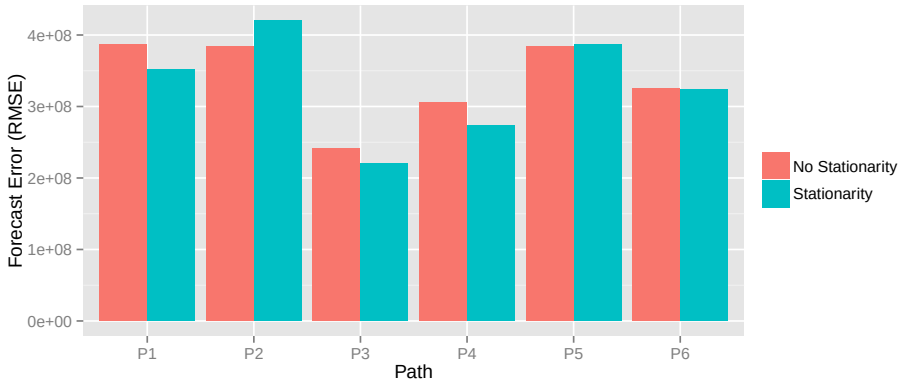


Fig. 6: Forecast Error Comparison for Stationary Assumption: The training set size is 4 weeks. The seasonal cycle is one day.

error with the stationary assumption in our previous work [37] was 10.9% less than the forecast error without the assumption. Higher forecast errors in this experiment were resulted from more frequent abrupt bandwidth usage changes during the target dates. Nevertheless, the stationary assumption still held in both cases, and resulted in lower forecast errors.

We re-evaluated the forecast errors for different training set sizes with the delayed model update. Fig. 7 shows that training set size with 4 weeks resulted

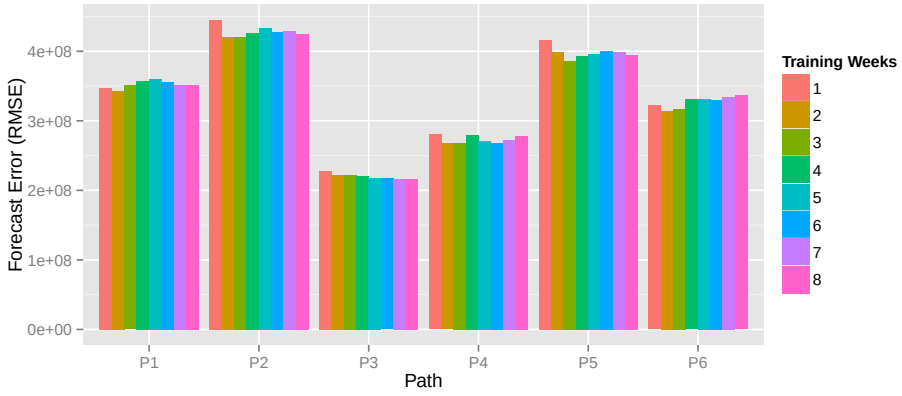


Fig. 7: Forecast Error Comparison of Delayed Model Update for Different Training Set Sizes: The training set size is from 1 to 8 weeks. The seasonal cycle is one day.

in better forecast accuracy.[6] We observed stationarity assumption of SNMP data holds up to 8 weeks in the training dataset. This observation led to a hypothesis that delaying model updates at least one week would not degrade the forecast error, instead of updating and re-fitting the model whenever new measurement data is available.

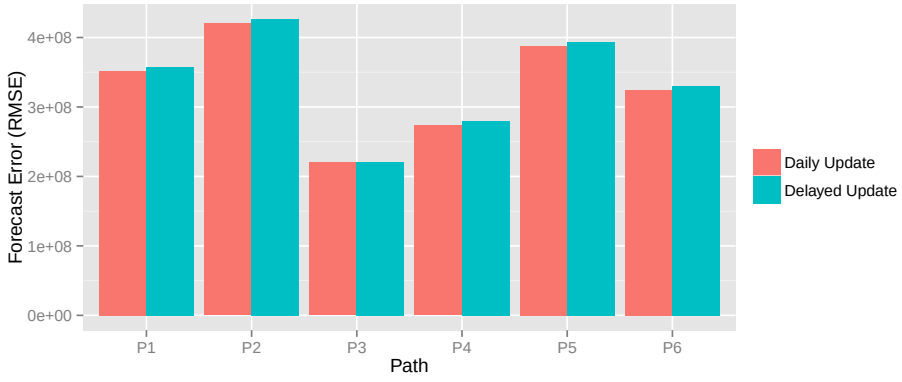


Fig. 8: Forecast Error Comparison for Delayed Model Update: The training set size is 4 weeks. The seasonal cycle is one day.

[6] In this particular target dates, 3 weeks resulted in better accuracy than 4 weeks. However, 4 weeks were generally resulted in better accuracy in other target dates.

Fig. 8 shows the forecast errors with delayed model update compared to those with daily update. The average forecast error with delayed model update was only 1.5% different with that of daily update. Instead of updating and re-fitting the model for the daily forecast with cross-validation, we updated the minimal parts in the model such as auto-correlation and moving averages from the initially fitted model. The result shows that the accuracy was not degraded, and the computation time was improved by 78.1% compared to traditional approach such as the Box-Jenkins methodology [7][8] with updating the models in daily period. The average CPU usertime from the delayed model update took 119.4s to forecast 7 days duration ahead per path compared to 545.6s from the model updated once per day (78.1% computation time reduction).

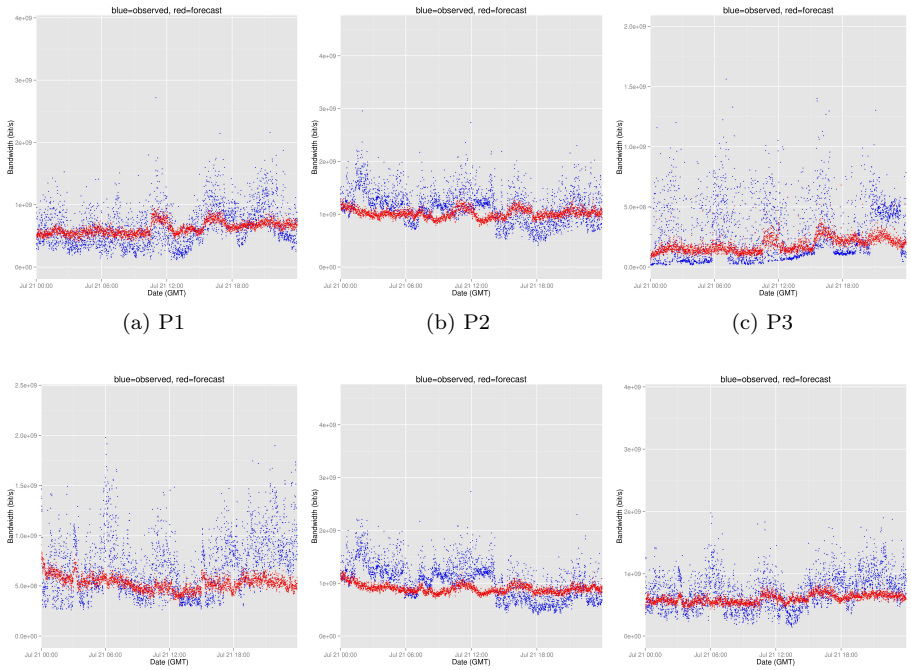
Fig. 9 shows the forecast results of the delayed model update for one day test set on July 21, 2014 in `P1`. It shows that our red-colored forecast values are close to the blue-colored observed data. Table 2 shows the standard deviations of the training set of 4 weeks and the test set from July 21 to July 27, 2014. This result shows the accuracy of our forecast model. When sudden spikes in the bandwidth utilization were observed from the training sets or the test sets, our forecast model was resilient to those sudden changes and accurate with RMSE within the standard deviations of the training sets or the test sets.

Since Mean Error (ME) is much closer to 0 than Mean Absolute Error (MAE) in Tab. 2, the forecast would be more accurate for the large data transfers. ME is denoted as $ME(h) = \frac{1}{h} \cdot \sum_{i=1}^{h} e_n(i)$, and MAE is denoted as $MAE(h) = \frac{1}{h} \cdot \sum_{i=1}^{h} |e_n(i)|$. This is because the forecast errors are mixed with positive and negative values. When the transfer time is longer than 30 seconds (10 TB transfer takes 800 seconds at theoretical maximum throughput on a 100Gbps network), the aggregated forecast errors from the large data transfer would decrease. With the same reason, increasing time-scale by smoothing would decrease the forecast errors.

Table 2: Forecast Error Metrics. The values are expressed in Gbps (10^8 bit/s). SD_{Train} is the standard deviation of the training set. SD_{Test} is the standard deviation of the test set. RMSE, MAE, and ME are the different types of forecast errors of the cross-validation.

PID	SD_{Train}	SD_{Test}	RMSE	MAE	ME
`P1`	3.86	3.32	3.57	2.18	0.25
`P2`	3.83	4.26	4.26	2.58	-0.57
`P3`	2.24	2.20	2.20	1.76	-0.38
`P4`	2.95	2.67	2.80	2.17	0.45
`P5`	3.99	4.07	3.93	2.74	-1.04
`P6`	3.70	3.08	3.31	1.79	0.30

Fig. 10 shows the forecast errors for the logit transformed data as in Eq. 5, compared to the forecast errors for the non-logit-transformed data (seasonally adjusted data). The forecast errors were derived from the forecast models using STL and ARIMA described in Sec. 3.2 and Sec. 3.3. The forecast error (RMSE) after the logit transformation was marginally better with the training set of 4



(a) P1 (b) P2 (c) P3

(d) P4 (e) P5 (f) P6

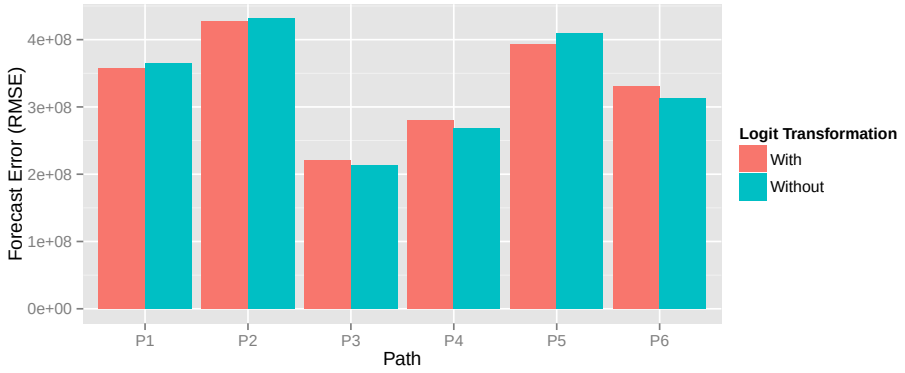
Fig. 9: Bandwidth Utilization Forecast: The forecast is for one target date, July 21, GMT 2014. The x axis is the date. The y axis is the bandwidth (bit/s). Blue colors are for the observed data. Red colors are for the forecasts.

weeks. It was 17.6% better with 8 weeks training set compared to those of non-logit-transformed data. This is because the logit transformation sets the lower and upper bounds in the modeling and fitting procedures, which helps reduce the potential under-estimation and over-estimation from the forecast.

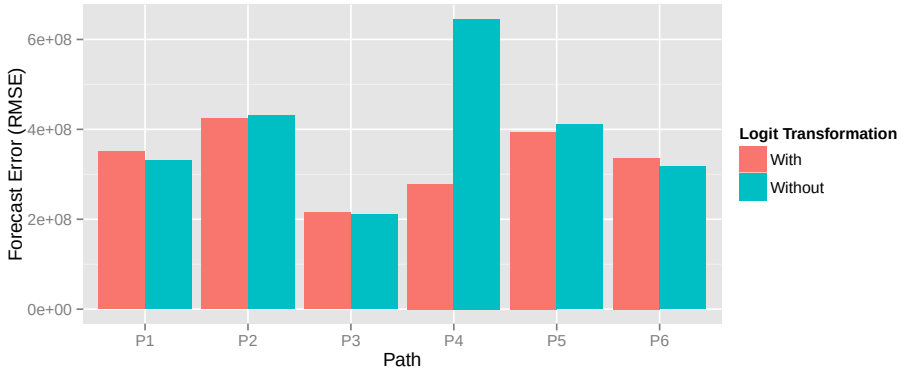
Fig. 11 shows the forecast errors for the logit transformed data without the stationary assumption. The forecast error was 58.9% better than the forecast errors for the non-logit-transformed data with the training set of 4 weeks, and it was 57% better with the 8 weeks training set. This result means that reducing the bounds by logit transformation is more effective when more training set size is required, or the fitted forecast model is non-stationary. Furthermore, this result confirms that both the logit transformation and the stationarity assumption are effective for the forecast model for the network bandwidth utilization with the SNMP measurement data.

5 Conclusions

We present a network bandwidth utilization forecast model. It can support efficient network resource utilization and scheduling, alternate path finding of



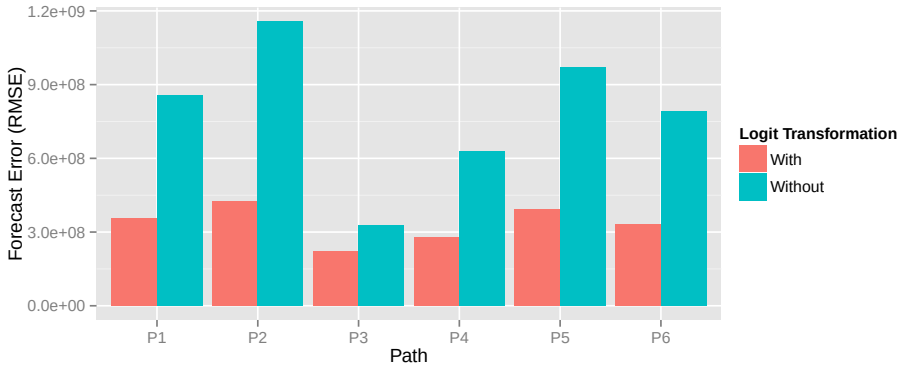
(a) 4 Weeks



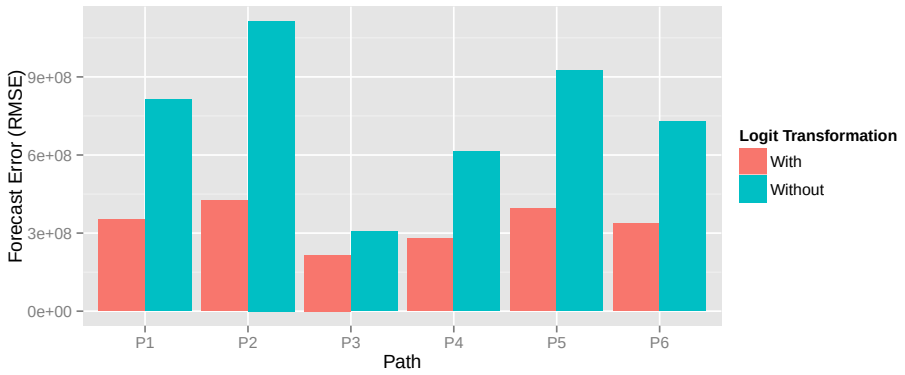
(b) 8 Weeks

Fig. 10: Forecast Error Comparison for Logit Transformation: The forecast is computed with stationary assumption. The seasonal cycle is one day.

data transfers, and planning on network link/bandwidth provision for high-bandwidth networks. Since data sharing opportunities over the wide-area network increase for large scientific collaborations that applications generate large volume of data, it is challenging to efficiently coordinate network resources on a shared network. In addition, sudden bandwidth utilization changes make the forecast more challenging. We observe that the network traffic behavior for the scientific networks shows stationarity and self-similarity in daily periodicity. Logit transformation and stationary assumption show effectiveness in reducing the forecast error. Our experimental results show that the delayed model update reduces the computation time by 78.1% compared to the traditional Box-Jenkins approach. It does not show the degradation of the forecast error when reducing the frequency of the model updates, and it shows the resiliency when there are sudden network bandwidth



(a) 4 Weeks



(b) 8 Weeks

Fig. 11: Forecast Error Comparison for Logit Transformation: The forecast is computed without stationary assumption. The seasonal cycle is one day.

utilization changes. Our forecast model is accurate having RMSE within the standard deviations of the observed measurements. It can be applicable to forecast other time series data with daily seasonality such as vehicular traffic data. The future work includes the adaptive forecast model based on the long-term trend changes in bandwidth utilization and the application of the forecast model to the network provisioning.

Acknowledgments

This work was supported by the Office of Advanced Scientific Computing Research, Office of Science, of the U.S. Department of Energy under Contract No. DE-

AC02-05CH11231. The authors would like to thank Chris Tracy, Jon Dugan, Brian Tierney, Inder Monga, and Gregory Bell at ESnet; Arie Shoshani, K. John Wu, Joy Bonaguro, and Jay Krous at LBNL; Richard Carlson at Dept. of Energy.

References

1. Energy Sciences Network (ESnet). `http://www.es.net/` (2014)
2. Network Simulator (ns2). `http://www.isi.edu/nsnam/ns/` (2014)
3. Aceto, G., Botta, A., Pescapé, A., D'Arienzo, M.: Unified architecture for network measurement: The case of available bandwidth. Journal of Network and Computer Applications 35(5), 1402–1414 (Sep 2012)
4. Akaike, H.: A new look at the statistical model identification. IEEE Transactions on Automatic Control 19(6), 716–723 (1974)
5. Balman, M., Chaniotakisy, E., Shoshani, A., Sim, A.: A Flexible Reservation Algorithm for Advance Network Provisioning. In: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis. ACM/IEEE (Nov 2010)
6. Benson, T., Akella, A., Maltz, D.A.: Network traffic characteristics of data centers in the wild. In: Proceedings of the Conference on Internet Measurement - IMC '10. pp. 267–280. ACM, New York, New York, USA (Nov 2010)
7. Box, G.E.P., Jenkins, G.M., Reinsel, G.C.: Time Series Analysis: Forecasting and Control. John Wiley & Sons, 4th edn. (2013)
8. Brockwell, P., Davis, R.: Time series: theory and methods. Springer-Verlag (2009)
9. Cleveland, R.B., Cleveland, W.S., McRae, J.E., Terpenning, I.: STL: A Seasonal-Trend Decomposition Procedure Based on Loess. Journal of Official Statistics 6(1), 3–73 (1990)
10. Cleveland, W., Devlin, S.: Locally weighted regression: an approach to regression analysis by local fitting. Journal of the American Statistical Association 83(403), 596–610 (1988)
11. Croce, D., Melliay, M., Leonardiy, E.: The quest for bandwidth estimation techniques for large-scale distributed systems. ACM SIGMETRICS Performance Evaluation Review 37(3), 20–25 (Jan 2010)
12. Crovella, M., Bestavros, A.: Self-similarity in World Wide Web traffic: evidence and possible causes. IEEE/ACM Transactions on Networking 5(6), 835–846 (1997)
13. Estan, C., Savage, S., Varghese, G.: Automatically inferring patterns of resource consumption in network traffic. In: SIGCOMM '03. pp. 137–148. ACM
14. Feamster, N., Rexford, J., Zegura, E.: The road to SDN: an intellectual history of programmable networks. ACM SIGCOMM Computer Communication Review 44(2), 87–98 (Apr 2014)
15. Hampel, F.R.: The Influence Curve and its Role in Robust Estimation. Journal of the American Statistical Association 69(346), 383–393 (Jun 1974)
16. He, Q., Dovrolis, C., Ammar, M.: On the predictability of large transfer TCP throughput. In: Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications. vol. 35. ACM, New York, New York, USA (Aug 2005)
17. Hjorth, J.: Computer intensive statistical methods: Validation, model selection, and bootstrap. CRC Press (1993)
18. Hu, N., Steenkiste, P.: Evaluation and characterization of available bandwidth probing techniques. IEEE Journal on Selected Areas in Communications 21(6), 879–894 (Aug 2003)
19. Hyndman, R.J., Koehler, A.B., Snyder, R.D., Grose, S.: A state space framework for automatic forecasting using exponential smoothing methods. International Journal of Forecasting 18(3), 439–454 (Jul 2002)
20. Jain, M., Dovrolis, C.: End-to-end available bandwidth. In: Proceedings of the 2002 conference on Applications, technologies, architectures, and protocols for computer communications - SIGCOMM '02. vol. 32, p. 295. ACM Press, New York, New York, USA (Aug 2002)
21. Krithikaivasan, B., Zeng, Y., Deka, K., Medhi, D.: ARCH-Based Traffic Forecasting and Dynamic Bandwidth Provisioning for Periodically Measured Nonstationary Traffic. IEEE/ACM Transactions on Networking 15(3), 683–696 (Jun 2007)

22. Kwiatkowski, D., Phillips, P.C., Schmidt, P., Shin, Y.: Testing the null hypothesis of stationarity against the alternative of a unit root. Journal of Econometrics 54(1-3), 159–178 (Oct 1992)
23. Leland, W., Taqqu, M., Willinger, W., Wilson, D.: On the self-similar nature of Ethernet traffic (extended version). IEEE/ACM Transactions on Networking 2(1) (1994)
24. Ljung, G., Box, G.: On a measure of lack of fit in time series models. Biometrika 65(2), 297–303 (1978)
25. Lu, D., Qiao, Y., Dinda, P., Bustamante, F.: Characterizing and Predicting TCP Throughput on the Wide Area Network. In: 25th IEEE International Conference on Distributed Computing Systems (ICDCS'05). pp. 414–424. IEEE (2005)
26. Mirza, M., Sommers, J., Barford, P.: A Machine Learning Approach to TCP Throughput Prediction. IEEE/ACM Transactions on Networking 18(4), 1026–1039 (Aug 2010)
27. Papagiannaki, K., Taft, N., Zhang, Z.L., Diot, C.: Long-term forecasting of internet backbone traffic. IEEE Transactions on Neural Networks 16(5), 1110–1124 (Sep 2005)
28. Paxson, V., Floyd, S.: Wide area traffic: the failure of Poisson modeling. IEEE/ACM Transactions on Networking 3(3), 226–244 (Jun 1995)
29. Pearson, R.: Data cleaning for dynamic modeling and control. Proceedings of European Control Conference (1999)
30. Qiao, Y., Skicewicz, J., Dinda, P.: An empirical study of the multiscale predictability of network traffic. In: Proceedings of the International Symposium on High performance Distributed Computing. pp. 66–76. IEEE (2004)
31. Ribeiro, V.J., Riedi, R.H., Baraniuk, R.G., Navratil, J., Cottrell, L.: pathChirp: Efficient Available Bandwidth Estimation for Network Paths. In: Proceedings of the Passive and Active Measurements (PAM) Workshop (2003)
32. Sang, A., Li, S.q.: A predictability analysis of network traffic. Computer Networks 39(4), 329–345 (Jul 2002)
33. Shao, J.: An asymptotic theory for linear model selection. Statistica Sinica 7, 221–264 (1997)
34. Shriram, A., Kaur, J.: Empirical Evaluation of Techniques for Measuring Available Bandwidth. In: Proceedings of the International Conference on Computer Communications. pp. 2162–2170. IEEE (2007)
35. Strauss, J., Katabi, D., Kaashoek, F.: A measurement study of available bandwidth estimation tools. In: Proceedings of the Conference on Internet Measurement - IMC '03. pp. 39–44. ACM, New York, New York, USA (Oct 2003)
36. Yin, D., Yildirim, E., Kulasekaran, S., Ross, B., Kosar, T.: A Data Throughput Prediction and Optimization Service for Widely Distributed Many-Task Computing. IEEE Transactions on Parallel and Distributed Systems 22(6), 899–909 (Jun 2011)
37. Yoo, W., Sim, A.: Network bandwidth utilization forecast model on high bandwidth networks. In: Proceedings of the IEEE International Conference on Computing, Networking and Communications (2015)