Advanced Performance Modeling
with Combined Passive and Active Monitoring

Annual Project Report – 2nd year
October 4, 2013

Project period of
October 1, 2013 through September 30, 2013

Principal Investigators
Alex Sim[1], Constantine Dovrolis[2]

Project members: Jeasik Choi[1], Kejia Hu[1], Demetris Antoniades[2]

Project Website: http://sdm.lbl.gov/apm

Table of Contents

[1] Lawrence Berkeley National Laboratory
[2] Georgia Institute of Technology

1 Project Summary

In recent years, network technologies have evolved rapidly. Advanced science networks such as those managed by ESnet and Internet2 in the US operate at speeds of up to 100 Gbps today. Owners of these networks are aggressively researching and deploying network-level QoS in an attempt to isolate large parallel data transfers from the rest of user traffic. Despite improvements in network technologies, optimal selection of shared network resources and efficient scheduling of data transfers in a distributed and collaborative environment are challenging tasks in achieving superior performance in accessing data. Monitoring the state of shared network resources and estimating their future performance have been used for identifying efficient selection and scheduling of network resources. However, existing network tools based on active probing cannot directly model high-speed networks and estimate or predict optimal network performance. They do not provide users with information about ongoing data transfers. Performance estimation models for high-speed networks should be able to consider new variables such as network fluctuating capacity and available bandwidth and higher concurrency of data streams. Moreover, existing estimation models performing active probing put extra load on the network resources, which might interfere with network performance. To maximize the throughput of data access operations, monitoring information about network performance needs to be collected in passive mechanisms without imposing extra load on network resources. This measurement information must be synthesized to drive a predictive estimation model of end-to-end network performance.

To improve the efficiency of resource utilization and scheduling of scientific data transfers on high-speed networks, we started a project on Advanced Performance Modeling with combined passive and active monitoring (APM) that investigates and models a general-purpose, reusable and expandable network performance estimation framework. The predictive estimation model and the framework will be helpful in optimizing the performance and utilization of networks as well as sharing resources with predictable performance for scientific collaborations, especially in data intensive applications. Our prediction model utilizes historical network performance information from various network activity logs as well as live streaming measurements from network peering devices. Historical network performance information is used without putting extra load on the resources by active measurement collection. Performance measurements collected by active probing is used judiciously for improving the accuracy of predictions. For a simple analogy, highway vehicle traffic pattern analysis (Figure 1) would give drivers time estimation for travel planning (e.g. it takes roughly about 1.5 hours from Berkeley to San Francisco Airport on Monday morning 8:30am, or about 40 minutes around 1pm).

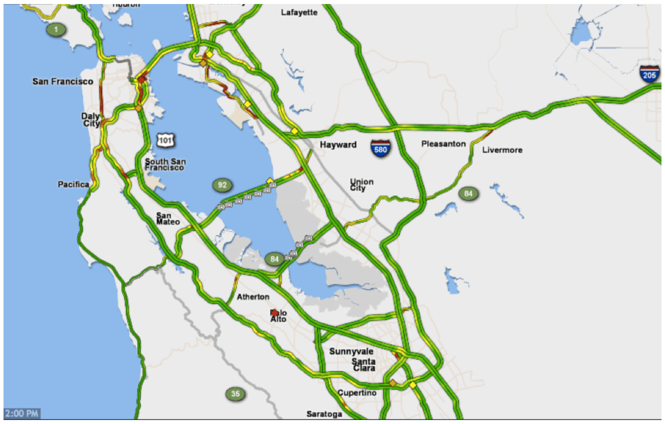

Figure 1: This project's network traffic analysis is analogous to highway traffic estimation and prediction.

This research explores fundamental questions on the relationship between monitoring and estimation of network resource performance.

This document gives a brief overview on the technical progress in the APM project from Lawrence Berkeley National Laboratory and Georgia Institute of Technology, for the project period from October 1, 2012 to September 30, 2013.

2 Project Accomplishments

2.1 Access to relevant network data

We have identified Simple Network Management Protocol (SNMP) link utilization time series and NetFlow logs as the most useful network performance data for this project collected by ESnet.

NetFlow data provides detailed information for end-to-end performance. Using NetFlow data, we can have accurate information about large end-to-end flows, the amount of data transferred back and forth, and the actual throughput of these transfers. NetFlow data comes with several privacy concerns however. NetFlow data include IP addresses, which raises significant user-privacy concerns over Personally Identifiable Information (PII) issues and data confidentiality. An important access agreement between our project and ESnet has been signed by the project members. Currently, 0.7 TB of NetFlow data from Jan. 2012 to Dec. 2012 is archived in a secured repository at LBNL and used for this project.

The SNMP link utilization time series provide aggregated traffic information for each ESnet component (a measurement every 30 seconds for every router interface). We have access to publicly available ESnet SNMP data collected from stats.es.net. There are no privacy issues associated with such data.

Additional SNMP time series from Georgia Tech’s main border router are now available for this project. Currently we acquired a 10-day SNMP dataset for all the interfaces of that router. Additional data are available upon request.

We have also discussed with Open Science Grid the possibility of accessing their data transfer log archive. Due to their funding limitations, the OSG project log archive is on hold currently. We plan to resume the discussion once the OSG log archive is available.

2.2 A novel edge-to-edge flow inference method using SNMP link utilization data

The Simple Network Management Protocol (SNMP) is widely used to monitor aggregated link usage from network components (routers, switches, etc.). Such data, even though they do not include a lot of information, provide a valuable source for network administrators, aiding decisions about network routing, provisioning and configuration. SNMP data is simple to collect and maintain, providing a low disk space option for a log of historical network usage.

In this work, we provide evidence that by leveraging SNMP link utilization data we can accurately identify edge-to-edge (e2e) flow information about large network transfers. The motivation for this work comes from the need of a statistically significant set of edge-to-edge throughput samples, allowing us to perform TCP throughput prediction in a monitored network based on historical measurements. The network-wide availability of SNMP data and the limited or highly constrained availability of NetFlow data provide the motivation to explore different approaches for increasing the size of the sample of flow throughput values.

We have developed a methodology for inferring network transfers from SNMP traffic utilization time-series data. Our method is the result of two main observations. First, looking at the time series data of a link’s usage, we observe events where the usage of the link increases (or decreases) to a different level, deviating from the link’s normal behavior up to that point. These events could be considered as starting (or ending) points of high-throughput transfers. Second, these events propagate from the input links of a router to the output links of the same router, and from there to the neighboring routers, allowing us to infer the actual route that the specific flow followed.

Figure 2 illustrates these observations over a realistic network example. Each router connects a DOE site’s internal network to other ESnet sites through several interfaces. Using SNMP link usage data one can form the utilization time-series for each interface, which represents the traffic transferred between two connected routers. Looking at the time-series between R7 and R9 one can observe an increase in the link utilization at some point. This increased utilization lasts for a time period and then drops. Such behavior can be attributed to a transfer initiated from R7′s access network towards some destination. Following this increase from R9 to the next router and so on, we can observe that the corresponding flow continues through R12 and R14. After R14 the flow either continues to another network or is destined to a machine in the access network served by R14. Note that the involved router interfaces do not have the same traffic variations in general. At the point that this transfer starts or ends, however, their traffic level changes in a similar fashion. Other transfers can be identified in different parts of the network at the same time. For example in Figure 2 we can observe a transfer between R1 and R11 and two transfers between R2 and R6.

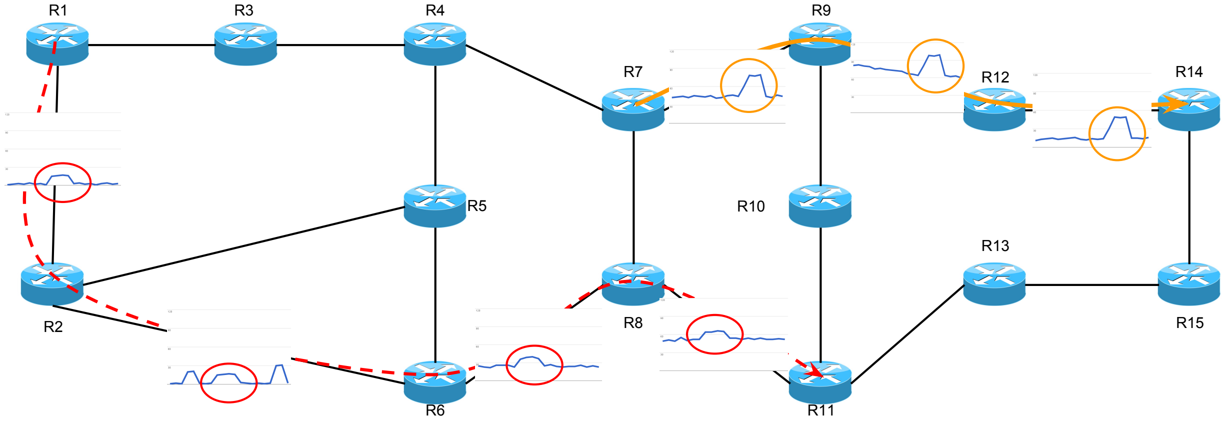

Figure 2: Several traffic utilization increase and decrease events can be identified in each observation period.

Our edge-to-edge flow identification methodology is based on two main observations. First, there exist significant changes in a link's utilization time series when a high-throughput flow starts or ends. Second, these changes appear along the flow's path of successive router interfaces. These two observations allow us to identify the flow start/end times, its duration and its throughput and to track the flow along the network, determining its edge-to-edge path.

Out method consists of the following four steps:

- **Event inference**: In the first step we identify flow-start and flow-end events in the link utilization time-series. This is an online processing step. We use a simple outlier detection method to identify if traffic at the current time period (last 30 seconds) has deviated significantly from the traffic's base behavior.
- **Mapping incoming events to outgoing interfaces**: After we identify an event (either flow-start or flow-end) at an input interface, we proceed by identifying the output interface at the same router that the flow is forwarded to. Our algorithm considers all flow events that appear at any router input interface in that time period, and tries to find the most likely outgoing interface that each of those flow events also appears in.
- **Identify edge-to-edge path**: This step aims to identify the next router that each identified flow is forwarded to. This step is accomplished easily when we have the network topology of the given network (ESnet), including the IP address of every router interface in that network. If this information is not available, it can be inferred using a periodic set of traceroute measurements between all network edge-points.
- **Infer transfer duration and throughput**: After we identify a flow-end event, we can estimate the average throughput during the flow's duration, based on the variations in the link utilization when that flow started and ended.

Note that the first three steps of our algorithm can be executed in real-time as new traffic utilization data become available. The information provided by these steps, though not complete, may also be of interest to understand the edge-to-edge flow of traffic in the given network. Full information about an individual transfer becomes available as soon as the transfer finishes.

To evaluate our method, we created experimental transfers using a client-server tool between two hosts located at Georgia Tech and LBNL. Figure 3 plots the true positive rate (TRP) as a function of the throughput of the transfer event. We can observe that the method can identify events larger than 3Mbits/sec with a TPR that is larger than 95%. The drop in TPR for smaller throughput values is mostly attributed to noise. Figure 4 plots the TPR as a function of the transfer duration. The TPR rate seems to be stabilized for values larger than 90% for transfer durations that are longer than 60 seconds. In the ESnet SNMP dataset we have a new utilization value every 30 seconds. In general, transfers with durations longer than or equal to two SNMP reporting periods can be identified with high accuracy.

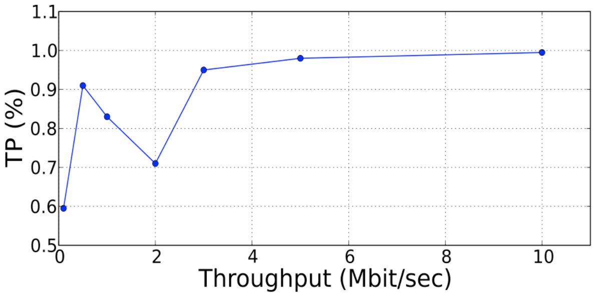

Figure 3: *True positive rate of event inference method as a function of the event throughput.*

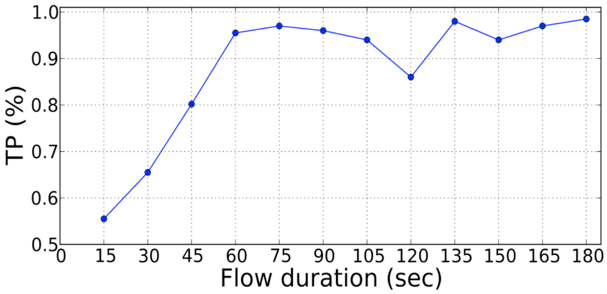

Figure 4: True positive rate of event inference method as a function of the event duration.

2.3 Study of statistical prediction model with NetFlow data

NetFlow records are composed of multiple time series with uneven collecting time stamps, and the traditional single time series model is infeasible to model the data. Also, NetFlow records show mixed effects of variables in the data, and simple consideration of fixed effects will cause information loss in the model. Additionally, the large volume and high dimension of NetFlow data require a fast algorithm. Generalized Linear Mixed Model (GLMM) has the capability of flexible structure of the model in the link function, variance sourcing and incorporating mixed effect without restriction on the size of data, and fits our needs of analyzing the NetFlow data. However, conventional methods such as Estimation Lasso and the Backward-Forward Selection are very computationally costly, and may not handle large dataset with n observations in p dimensions. In this project, we developed an efficient statistical method, the Predictive Lasso, which calculates the prediction without multiple iterations in O(np). We improved the model with the approach obtaining the estimates of fixed and random effects by Lasso with minimum mean squared prediction error (MSPE) in Linear Mixed Model (LMM). Then, we extended the results to GLMM with the log link and Poisson assumption. A major computational advantage is that our procedure does not require the Expectation-Maximization (EM), unlike other previous methods, which require utilizing the EM algorithm to handle the unobserved random effects. We developed the new approach based on bootstrapping to select the optimal penalty parameter λ in Lasso.

The GLMM is defined with a vector of random effects v and the responses $y_1,...,y_m$ of m groups that are conditionally independent such that the probability density function (pdf) of each response $f_i(y_i|v)$ follows the exponential family with $E(y_i|v) = \mu_i$, $g(\mu_i) = x'_i\beta + z'_iv$, $g^{-1} = h$ where $v \sim N(0, \Phi)$, x_i is the observed fixed effect, and z_i is the index that indicates the group of random effect. The g(.) is the link function, and takes various forms such as Gaussian, Poisson and Logit with different assumptions of the model. In the NetFlow data, there are two types of variables; (1) continuous variables such as the size of the data transfer and the transfer duration, and (2) count variables such as the number of congestions or the number of extreme large data transfers within a certain time window length. In order to predict these two types of response variable, the GLMM is constructed in the following two types.

- y_i is the continuous variable, and assumed $g(x) = x$, $y_i|v$ follows Gaussian distribution.
- y_i is the count variable, assumed $g(x) = \log(x)$, $y_i|v$ follows Poisson distribution.

The final prediction model following Gaussian distribution is built with equation for fixed effects $\check{\beta} = \mathit{argmin}_{\beta}(y - X\beta)'\Gamma'\Gamma(y - X\beta) + \lambda\ \Sigma|\beta_i|$ and for random effects $\check{d} = \mathit{argmin}_{d}(y - X\beta)'M'M(y - X\beta) + \mathrm{tr}((2HBZ - HF'Z - Z'FR')G) - \mathrm{tr}(FB\Sigma) + \lambda\Sigma|di|$ and has two advantages, immunity to model misspecification and fast computational algorithm. The details of our model and the validation of the model can be found in the paper submitted to the Annals of Applied Statistics (Publication, 5).

The model is applied to the 100 million NetFlow records to predict the data transfer duration for a certain volume of data and congestion frequency. The model ($y = \beta_{start}s(x_{start}) + \beta_{pkt}x_{pkt} + Z_{ip-path}v_{ip-path} + e$) predicts the data transfer duration, assuming influences from the fixed effects including transfer start time and transfer size and the random effects including network transfer condition such as protocol, source and destination port numbers and transfer path such as source and destination IP addresses. The model also suggests the importance of variation in random effects such as IP path and start time selection in the prediction of the transfer duration.

Table 1: Comparison of Mean Squared Prediction Error (MSPE) and modeling performance

	Pred. Lasso (our model)	Est. Lasso	B-F Selection
MSPE	127.3	2306	42230
Modeling time (in sec)	142	6.26e+7	5.43e+10

Compared to two other approaches shown in Table 1, our approach Predictive Lasso shows that the prediction accuracy is 18 times better than the Estimation Lasso and 330 times better than the Backward-Forward Selection, and the computation time is the least with 4e+5 times less than the Estimation Lasso and 3.8e+8 times less than the Backward-Forward Selection. Our model greatly improves the prediction accuracy for data transfer duration, which fits the interests of NetFlow data modeling, and provides efficient algorithm.

2.4 Study of data reduction method for large streaming data

Network measurement data is a large streaming data. It is in general intractable to store, compute, search and retrieve large streaming data, but handling large streaming data is essential for various applications such as social network media analysis, energy usage trends, environmental modeling, large scientific simulations, or this project, network traffic performance analysis. Currently, network measurements such as NetFlow collect a sample, one out of 1000 network packets. This is one of the possible random sampling solutions reducing the storage of vast amounts of measured data. Such static sampling methods (linear sampling) has drawbacks: (1) it is not scalable for high-rate streaming data, and (2) there is no guarantee of reflecting the underlying data distribution. In this project, we studied a fundamental issue, which is to reduce the size of large streaming data and still obtain accurate statistical analysis. Our dynamic sampling algorithm reduces the data records in exponential scale, and still provides accurate analysis of large streaming data. We also built an efficient Gaussian Process with a fewer sample measurements, and applied the new algorithm to large data transfers in high-speed networks showing that the new algorithm significantly improves the efficiency of network traffic prediction for large data transfers.

As shown in Figure 5 (a), the network traffic information may be concentrated on a specific time and throughput range (darker area includes 90% of traffic and thus storage size). However, it may not be necessary to store and process all network traffic information. For the purpose of analyzing traffic throughput patterns, we can avoid handling redundant (or noisy) data. As shown in Figure 3 (b), when each network traffic unit is in binary size (0 or 1) and independent and identically distributed (*iid*), the sizes of packets can be modeled by a single Bernoulli parameter θ. In reality, such network traffic data is not *iid*. Instead, the order of network traffic can be (locally) exchangeable. We defined exchangeable random variables and Gaussian processes, and they are used to explain our new model, which we called Bayesian Online-Locally Exchangeable Measures (BO-LEMs). Our algorithm can reduce the number of required network measurement samples by 66% while achieving accurate data analysis for network throughput prediction. The details of our algorithm and the model can be found in the paper LBNL-6341E (Publication, 4).

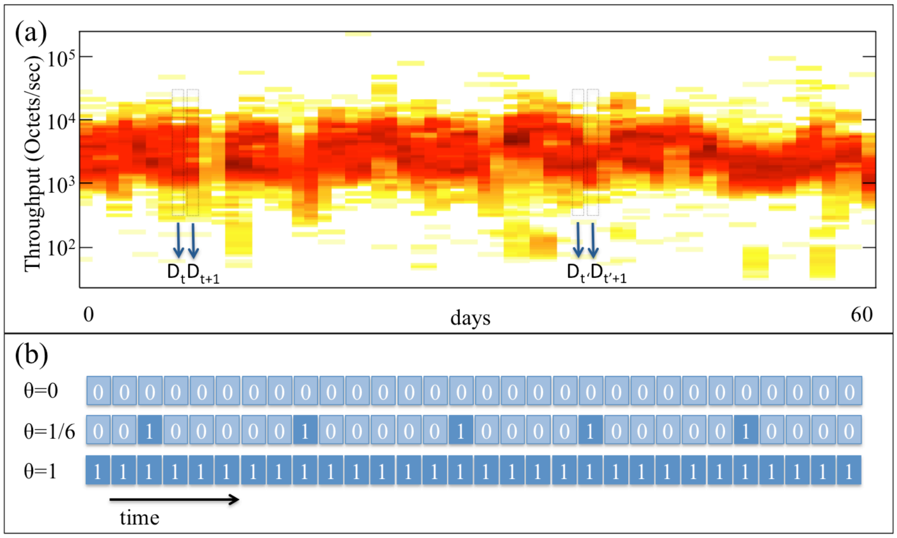

Figure 5: (a): Network traffic patterns on a high-speed network router in ESnet. The regions with darker color represent higher network traffic concentrations. This figure represents network transfers as a density at a particular time (x-axis) and a throughput (y-axis). The yellow, red and black colored regions represent small, moderate and large numbers of transfers, respectively. At a specific time step, the figure represents the network transfers. As an example, two regions at D_t and D_{t+1} are network transfer profiles next to each other. As can be seen, the two regions are similar. Meanwhile, two regions at $D_{t'}$ and $D_{t'+1}$ are similar. (b): An illustration of generating *iid* samples from Bernoulli distribution.

3 Interaction with other groups and projects

- We have collaborated with ESnet for data access, specifically with Brian Tierney, Chris Tracy, Jon Dugan, Chin Guok, Inder Monga and Greg Bell.
- We have collaborated with Shawn McKee at Univ. of Michigan for our access to the OSG data transfer log archive.
- We have collaborated with Warren Mathews and John Merritt at Georgia Tech for our access to the SNMP log data.

4 Plans for the third year

a) Finalize the evaluation of our edge-to-edge transfer inference method using NetFlow data from Georgia Tech and ESnet, and publish the method.

b) Develop a prediction method that uses the previous edge-to-edge transfer throughput estimates to predict the expected throughput at a given edge-to-edge path.
c) Combine the developed methods for end-to-end throughput inference using NetFlow and SNMP link usage data in an integrated prediction framework.
d) Build a prototype of our network performance inference and prediction system.
e) Collaborate with ESnet for the retrieval of recent NetFlow log data from R&E sites.
f) Present the results of this project at related research groups and at the relevant network operations community, including an ESCC/Internet2 Joint Techs meeting, a NANOG meeting, and other network performance-focused workshops.

5 Publications, presentations and other activities

Papers and talks presented during this time period:

5.1 Presentations

1) "*Advanced Performance Modeling with Combined Passive and Active Monitoring*", A. Sim, C. Dovrolis, PI Meeting, 3/2013.
2) "*Statistical prediction models for network traffic performance*", K. Hu, A. Sim, D. Antoniades, C. Dovrolis, The Winter 2013 APAN/ESnet /Internet2 technical meeting (Joint Techs, TIP2013), 2013.

5.2 Publications

1) "*What SNMP data can tell us about Edge-to-Edge network performance*", D. Antoniades, K. Hu, A. Sim, C. Dovrolis, poster in the Passive and Active Measurements Conference (PAM2013), 2013.
2) "*Estimating and Forecasting Network Traffic Performance based on Statistical Patterns Observed in SNMP data*", K. Hu, A. Sim, D. Antoniades, C. Dovrolis, The 9th International Conference on Machine Learning and Data Mining (MLDM2013), 2013.
3) "*Best Predictive GLMM using LASSO with Application on High-Speed Network*", Kejia Hu, Jaesik Choi, Jiming Jiang, Alex Sim, Tech Report LBNL-6327E, 2013.
4) "*Relational Dynamic Bayesian Networks with Locally Exchangeable Measures*", Jaesik Choi, Kejia Hu, Alex Sim, Tech Report LBNL-6341E (submitted to ICDM2013), 2013.
5) "*Analyzing High-Speed Network Data*", Kejia Hu, Jaesik Choi, Alex Sim, Jiming Jiang, submitted to the Annals of Applied Statistics, 2013.
6) "*A novel edge-to-edge flow inference method using link utilization data*", Demetris Antoniades, Constantine Dovrolis, under preparation for submission to ACM SIGCOMM 2014.