Peeking Network States with Clustered Patterns

Jinoh Kim[1,2], Alex Sim[2]

[1]Texas A&M University, Commerce, TX, USA
[2]Lawrence Berkeley National Laboratory, Berkeley, CA, USA

Disclaimers

This document was prepared as an account of work sponsored by the United States Government. While this document is believed to contain correct information, neither the United States Government nor any agency thereof, nor the Regents of the University of California, nor any of their employees, makes any warranty, express or implied, or assumes any legal responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by its trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof, or the Regents of the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof or the Regents of the University of California.

Peeking Network States with Clustered Patterns

Jinoh Kim[*][†]
jinoh.kim@tamuc.edu

[*]Department of Computer Science
Texas A&M University-Commerce
Commerce, TX 75428, USA

Alex Sim[†]
asim@lbl.gov

[†]Scientific Data Management Group
Lawrence Berkeley National Laboratory
1 Cyclotron Road, Berkeley, CA, 94720, USA

ABSTRACT

Network traffic monitoring has long been a core element for effective network management and security. However, it is still a challenging task with a high degree of complexity for comprehensive analysis when considering multiple variables and ever-increasing traffic volumes to monitor. For example, one of the widely considered approaches is to scrutinize probabilistic distributions, but it poses a scalability concern and multivariate analysis is not generally supported due to the exponential increase of the complexity. In this work, we propose a novel method for network traffic monitoring based on clustering, one of the powerful deep-learning techniques. We show that the new approach enables us to recognize clustered results as patterns representing the network states, which can then be utilized to evaluate "similarity" of network states over time. In addition, we define a new quantitative measure for the similarity between two compared network states observed in different time windows, as a supportive means for intuitive analysis. Finally, we demonstrate the clustering-based network monitoring with public traffic traces, and show that the proposed approach using the clustering method has a great opportunity for feasible, cost-effective network monitoring.

Categories and Subject Descriptors

C.2 [**Computer-Communication Networks**]: Miscellaneous

General Terms

Management, Measurement

Keywords

Network traffic, monitoring, clustering, degree of changes

1. INTRODUCTION

Network traffic monitoring has long been an essential task in network operations and management for various purposes, such as network performance measurement and forecasting [27, 18, 13], traffic classification [16, 10, 19, 6], analysis and visualization [14, 19], security [26], and so forth. The monitored result and historical information can be utilized for reconfiguring the network to optimize performance or to reinforce security. While hard to stress its importance enough, the basic and undeniable assumption upon the monitoring work is the irregular property of tracked variables, which makes it challenging to overlook the current snapshot or to estimate future states of the network. Moreover, a sophisticated monitoring task involves a complex degree of analysis against a set of monitored variables to understand the state of the network comprehensively. The network traffic data volume is also a concern for scalable monitoring. These additional complexities make it infeasible to perform in-depth examination as well as timely detection and response.

A simple form of network traffic monitoring, keeping track of the volume of incoming/outgoing traffic in the network, would not be comprehensive, and can only provide a partial characteristic of the network state. One of the traditional approaches is to scrutinize probabilistic distributions of essential variables related to traffic statistics, obtained from the data set collected within a predetermined time interval (known as a time window) [5]. For example, the distribution of packet lengths can be referred to the characterization of the network state in the current time window. However, a non-trivial challenge with this approach is an exponential increase of computational and storage complexity with additional variables to be monitored and larger time windows. Since the network administrator may want to include multiple traffic-related attributes in the monitoring process, multivariate analysis is commonly needed for network monitoring, making it less attractive to employ the distribution-based monitoring.

In this paper, we propose a novel approach to the network traffic monitoring using *clustering*, one of the powerful deep-learning techniques. What makes the new proposed approach unique and more efficient than traditional approaches is that the clustering method has the ability to combine multivariate attributes in a straightforward manner without an excessive extra cost, which is a critical challenge when relying on the probabilistic distributions. Moreover, it is possible to construct the deterministic number of clusters ($k > 1$) regardless of the number of variables to be tracked, thus simplifying the monitoring process. In addition, the clustering method is an unsupervised learning technique which does not require a prior knowledge and a complicated training process with labeling that may be even unavailable.

The basic idea of our proposed approach is to recognize clustered results as patterns that represent the network states. The patterns can then be utilized to evaluate "similarity" of network states over time. Since a state consists of a static number of clusters, comparing similarity of given network states can be reduced to a pattern comparison problem. In this paper, we demonstrate the clustering-

.

based network monitoring with public traces data, and show the pattern changes over time with visual representations. In addition to the changing patterns, we establish a new measure called "degree of changes" to quantitatively gauge the similarity between two compared states observed in different time windows, as a supportive means for intuitive analysis. Our preliminary results indicate that the new proposed approach using the clustering method has a great opportunity for feasible, cost-effective network monitoring.

This paper is organized as follows. In Section 2, we provide a brief summary of relevant studies on network traffic monitoring. Section 3 presents our initial experiment and its result to examine the distribution-based monitoring with a public traffic data. In Section 4, we introduce our proposed approach based on clustering, and show how feasible it is for monitoring with our observations from a set of experiments. Finally, we conclude our presentation in Section 5 with a summary and future exploration plans.

2. RELATED WORK

Network traffic classification has boosted its importance over the decade as an effective means of network management and security, using payload inspection and machine learning (ML) techniques [6, 17]. With the increasing use of encryption and the emphasis on privacy, the latter approach using flow statistics with no need for inspecting payloads has been widely studied, with supervised learning with well-known classifiers [12] and semi-supervised learning based on clustering [28, 10, 4]. While supervised learning is known to yield greater accuracy, supplying labelled data is not a trivial requirement, which made clustering to be spot-lighted. However, the semi-supervised approach also requires such training-purpose data despite the use of clustering. Although numerous proposals have been made, network traffic classification has still many challenges, and a critical one is classification accuracy with limited availability of traffic traces for testing and the daily emergence of newly introduced applications [6]. In this work, we use clustering but we do not attempt to classify network traffic; rather we develop a novel method for network traffic monitoring using clustering.

Probabilistic models and samplings have been considered for network traffic monitoring, especially for high volume traffic, and changing data patterns has been studied in streaming network traffic measurements [5]. Network monitoring could use streaming data mining techniques, and sampling methods and data reduction techniques were studied by frequency counting [7], histogram [11], sliding windows [8], random sampling [22], wavelets [20], and dimensionality reduction [24]. Many of these sampling methods provide a quick understanding of the monitored data stream, but characterizing accurate data distribution from the streaming data is still a challenge, especially with the recent hardware advances, which produces data records at a much higher rate. In addition, the critical hurdle is how to combine multiple attributes for comprehensive analysis rather than single dimensional streaming data analysis, as we will discuss in detail in the next preliminary study section.

Visualization has also been widely accepted for network management with the power of intuitive analysis. CAIDA provides a tool for visualizing the Internet topology using the Autonomous Systems (ASes) information [1], which is helpful to understand the interconnectivity of routing systems over the global Internet. Another tool provides a cyber-security map visualizing global cyber attacks with the source, target, and attack information in real time [3]. Additionally, the NeTraMark project [19] implemented tools for BLINK [16] and Traffic Dispersion Graphs [14], mainly for traffic classification. Through the visualization, we will present that our new approach based on clustering opens a great opportunity for intuitive analysis.

3. PRELIMINARY STUDY USING DISTRIBUTIONS

We define monitoring as an activity that constantly watches the monitored target to identify its state, for detecting unexpected patterns or sudden changes over time. The target of network monitoring would be a local or ISP network, and the administrators analyze the network states using computing tools and collected network traffic statistics.

We assume that the time domain is partitioned by a predetermined fixed interval for the monitored target, and the data collection for monitoring takes place within a time window W_i to determine the associated network state S_i. We also define *degree of changes* as the dissimilarity of two states, and use the notation $\Delta_{i,j}$ for comparing two states S_i and S_j.

Our approach to examine a state is with the distributions of the given network traffic attributes. As an initial study, we conducted experiments to see the effectiveness of probabilistic distributions for network monitoring, with a public trace UNIBS[1] collected between 10AM on September 30 and 2AM on October 1, 2009 [9]. Note that we selected the above time frames without any preferences out of the 3-day trace data set. The average number of flows is 789 flows/hour with a high degree of variance (min=20, max=7052).

Figure 1 shows cumulative distributions for two variables (flow duration and average number of packets in flows) over 16 monitoring time windows, each of which has one-hour in length. From the figure, we can see that some time windows have somewhat similar patterns, while some others show clearly different patterns. To better understand the characteristics of the given traffic data set, we analyzed the composition of applications in each window. The breakdown of applications was performed using the associated groundtruth data provided together with the traces [25]. We agree that it is hard to explain the network state only through the compositions of applications. However, we believe that it is a good reference to infer network states. Figure 2 shows the compositions of applications over the time windows. It shows that some time frames are highly related, such as 11AM–5PM and 8PM–10PM, with respect to application compositions. In contrast, some show strongly unrelated breakdowns with other windows, such as in 10AM, 11PM, 12PM, and 1AM.

From the distributions shown in Figure 1, those related time windows in the application breakdowns also show similar patterns (e.g., 11AM–5PM) in the plots. However, some plots do not seem to agree with the breakdown information. For example, two windows for 12AM and 1AM show a very close pattern although the compositions of applications for those windows are highly distinctive each other. From the result, network monitoring using distributions would be useful but only to some extent.

In addition to visual monitoring, the degree of changes may be considered as a supportive information between two time windows, which can be measured with KS test [2]. However, this approach would be complicated when considering additional variables [15, 23], hampering comprehensive recognition of patterns.

These challenges motivate us to explore possible alternative approaches that may have greater potential in scalability and feasibility. As will be introduced in the next section, clustering, with its strong benefit of aggregation, enables to yield a predetermined number of clusters regardless of the number of variables considered in monitoring. Therefore, network monitoring using clustering would be highly scalable, and provide a consistent way to keep track of patterns for network states.

[1] http://www.ing.unibs.it/ntw/tools/traces/

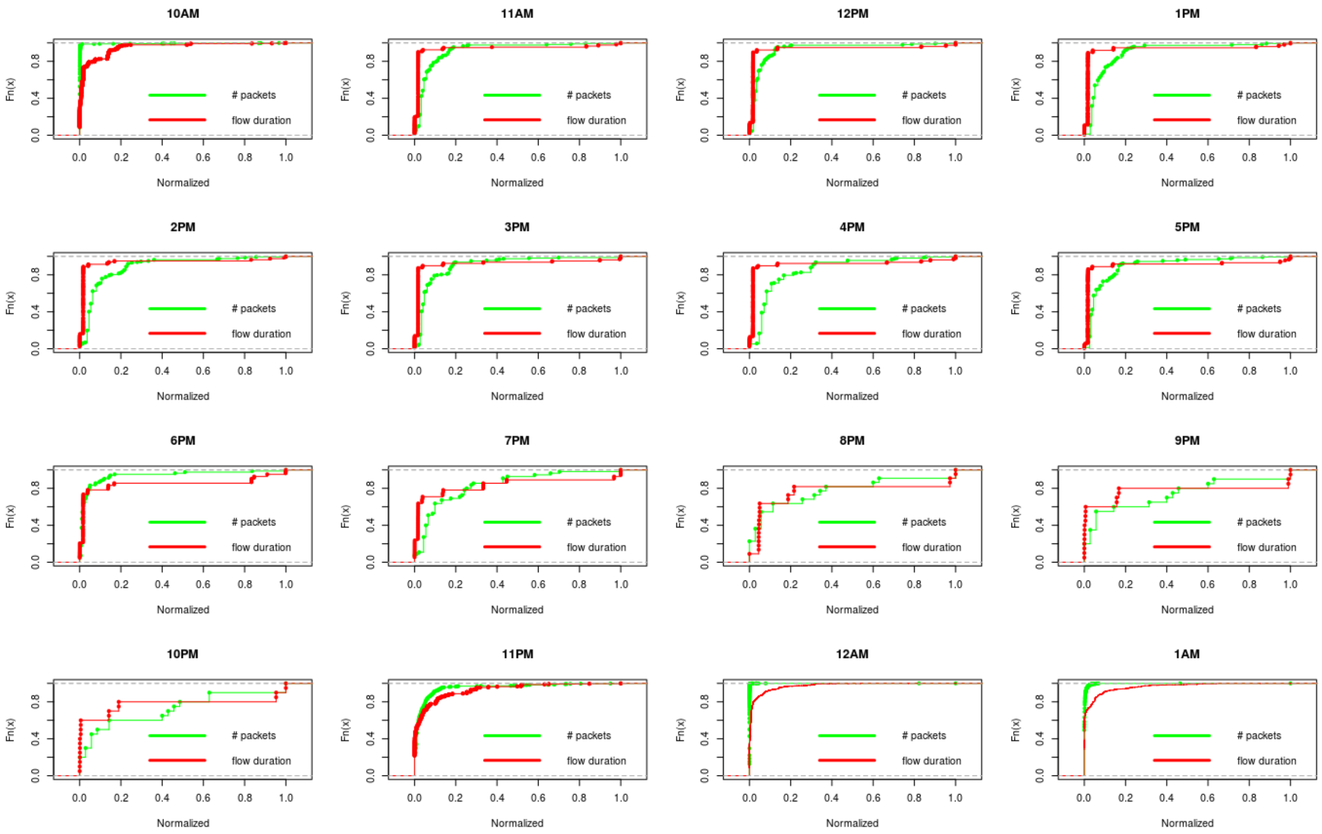

Figure 1: Cumulative distributions of two network traffic attributes (flow duration and average number of packets in flows) over 16 consecutive time windows, on UNIBS trace between 10AM on Sep. 30, 2009 and 2AM on Oct. 1, 2009.

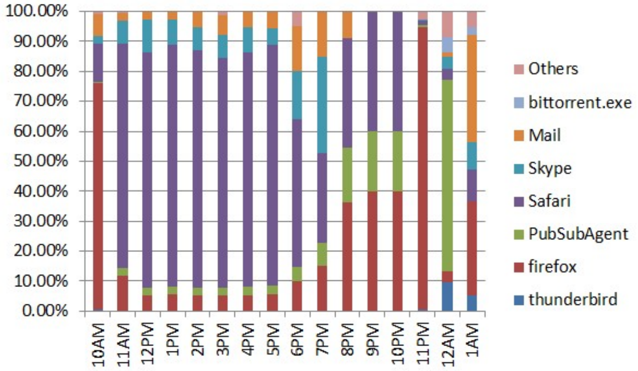

Figure 2: Breakdown of applications for time windows (10AM–1AM), compiled from the associated groundtruth data

4. THE PROPOSED APPROACH

In this section, we present our approach to network monitoring based on clustering. For clustering, we selected the simple K-means algorithm with its manageable complexity. We first demonstrate clustering results against the UNIBS data used in the previous section. The degree of changes measure is introduced next with the quantitative results.

Figure 3 shows the clustering results for the time windows with two attributes: flow duration (x-axis) and the number of packets per flow (y-axis). For the number of clusters, we chose four clusters since it approximates to the minimal sum of squares within groups with a relatively small number of clusters. Note that the cluster IDs and colors in the plots were randomly selected by the tool (R). As shown in the figure, the clustered results are highly intuitive to interpret and also to figure out correlated patterns. The clustering result for 10AM is quite different from the one for 11AM, while the clustering results for 11AM–5PM look closely similar. The cluster patterns from 8PM to 10PM also show highly similar patterns. Unlike these, the time frames between 11PM–1AM show clearly different and unrelated patterns. Interestingly, we can see that the clustering results strongly agree with the compositions of applications, shown in Figure 2.

From Figure 3, we see that the clustering-based monitoring is helpful to intuitively identify similarity of patterns among time windows. However, some patterns may not be clear enough to determine similarity of them only through the visual monitoring. Any quantitative measure would be helpful to perceive the degree of similarity with greater confidence if provided. We next introduce the degree of changes (Δ) that we defined for this purpose, which measures the gaps between compared clustered results based on centroid positions of clusters in the Euclidean space.

Suppose two time windows W_i and W_j, and the associated cluster sets $C_i = \{c_i^0, c_i^1, ..., c_i^k\}$ and $C_j = \{c_j^0, c_j^1, ..., c_j^k\}$, respectively, where k is the number of clusters. Each cluster c_x^y has its centroid p_x^y. Without knowing which cluster in W_i is mapped with one in W_j, we find a set of pairs showing the minimal move. Suppose a distance function $D : C_i \times C_j \rightarrow \mathbb{R}$. Then the problem is re-

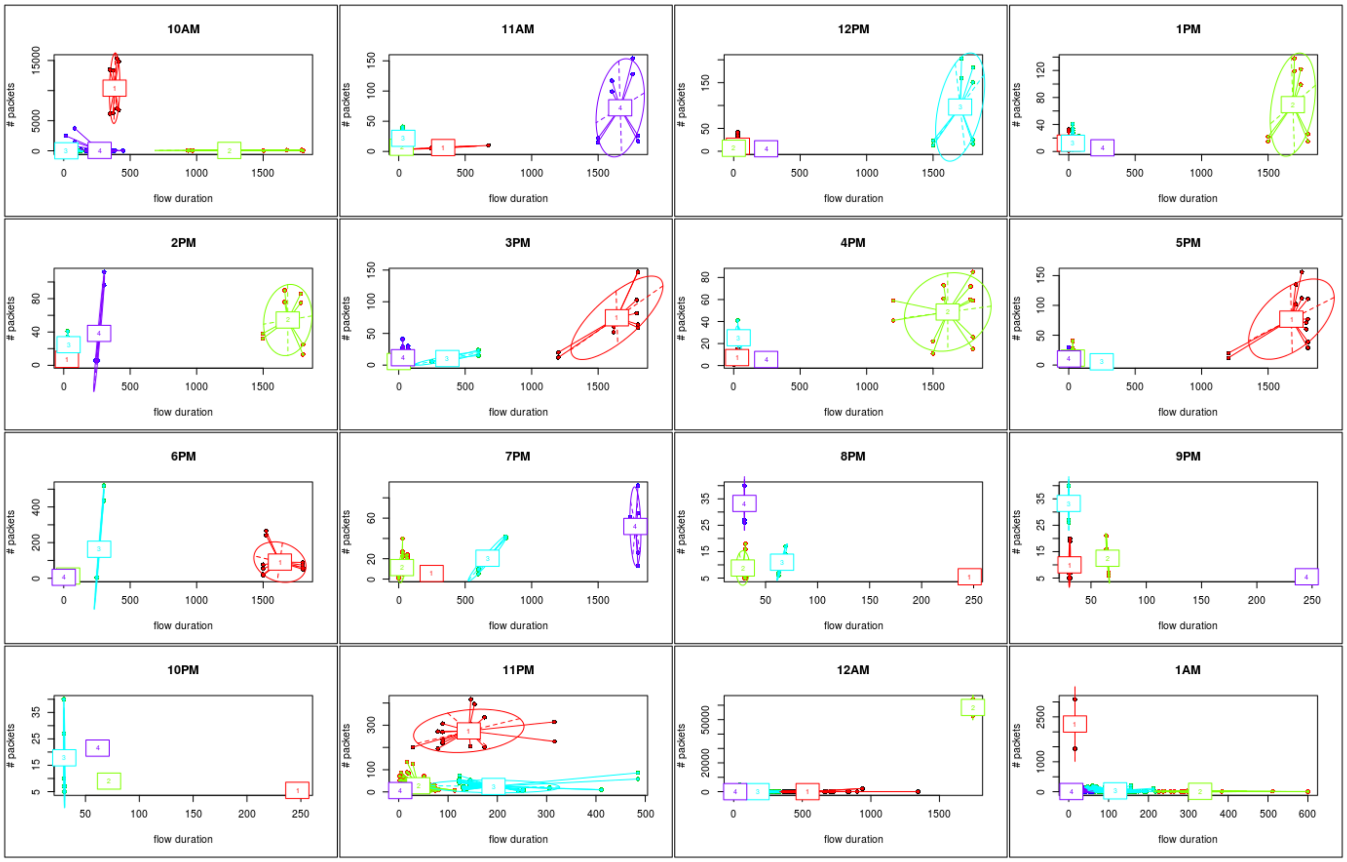

Figure 3: Clustering results (light traffic) over 16 consecutive hourly time windows on UNIBS data trace for flow duration on x axis and average number of packets in flow on y axis, between 10AM on Sep. 30, 2009 and 2AM on Oct. 1, 2009. The number of clusters is set to 4, based on the minimal sum of squares. Note that cluster ID and colors were selected randomly.

duced to the assignment problem that finds a bijection $f : C_i \rightarrow C_j$ with the minimal distance function:

$$\Delta_{i,j} = \sum_{l \in C_i} D(l, f(l))$$

Hungarian algorithm is a well-known method for this type of problem with $O(k^3)$ of the computational complexity where k is the number of clusters [21].

Table 1 shows the degree of changes with statistical significance obtained from five identical runs on the clustering results in Figure 3. Note that we used the Euclidean distance to measure centroid moves after normalized. From the table, we observed relatively small degree of changes for time windows in 11AM–6PM and 8PM–10PM ($\Delta \leq 0.22$). In contrast, we observed a high degree of changes in the adjacent windows of (10AM, 11AM), (7PM, 8PM), (11PM, 12AM) and (12AM, 1AM) ($\Delta > 1.0$), which almost agreeing with the observations on the clustered patterns in Figure 3. The variations are overall insignificant with $\sigma \leq 0.1$ except for the first two Δ's. From the calculated results, this quantitative metric is effective for providing supportive information regarding network traffic pattern changes over time.

To further investigate the effectiveness of identifying patterns and the correlation between the clustered patterns and the metric of the degree of changes, we experiment on a different data set. In this experiment, we did not consider application breakdowns, but compared the plotted patterns with the associated quantitative information. The new data set is a part of the UNIBS data trace, collected in October 2, 2009. Since the data set includes a relatively large number of flows, we performed clustering with every five minute traces from 8:00AM to 9:20AM. The average number of flows is 973 flows/hour with a low variance (min=599, max=1350). The configuration for clustering is the same as the previous experiment with four clusters, based on the sum of squares within groups.

Table 1: Degree of changes (based on centroid position move)

Windows	Average (μ)	Std. Dev. (σ)
(10AM,11AM)	1.04	0.40
(11AM,12PM)	0.22	0.32
(12PM,1PM)	0.02	0.01
(1PM,2PM)	0.04	0.06
(2PM,3PM)	0.11	0.06
(3PM,4PM)	0.12	0.06
(4PM,5PM)	0.07	0.06
(5PM,6PM)	0.08	0.06
(6PM,7PM)	0.41	0.07
(7PM,8PM)	1.32	0.00
(8PM,9PM)	0.05	0.10
(9PM,10PM)	0.07	0.10
(10PM,11PM)	0.10	0.01
(11PM,12AM)	1.57	0.04
(12AM,1AM)	1.52	0.00

Figure 4 demonstrates the clustering results on the 5-minute time

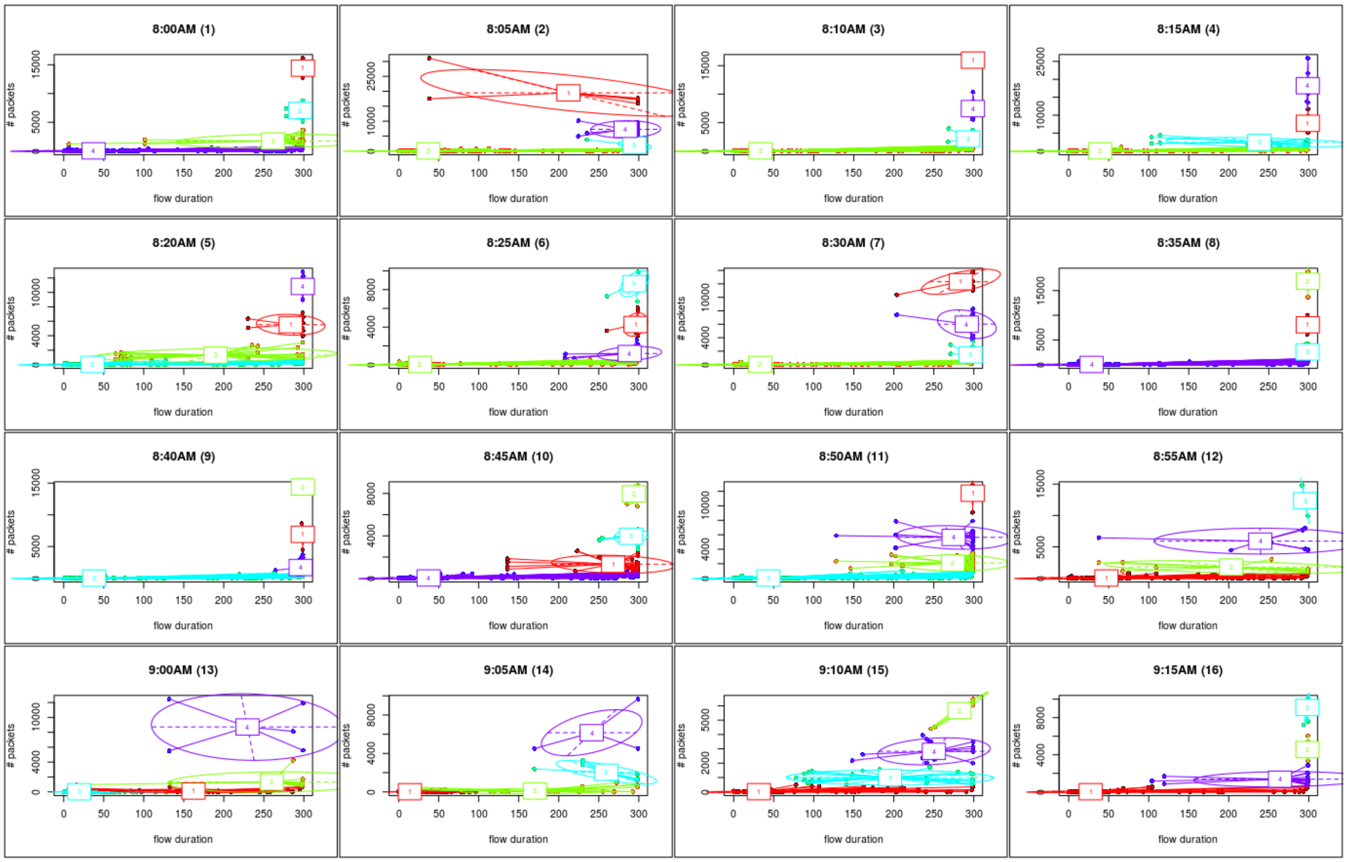

Figure 4: Clustering results (heavy traffic) on UNIBS data trace, for flow duration on x-axis and average number of packets in flow on y-axis, between 8AM and 9:20AM on Oct. 2, 2009.

window data set. For ease of presentation, we identify individual time windows with the unique ID in addition to their beginning time. As in Figure 3, the figure shows clustered pattern changes over time with discernible similarity/dissimilarity between windows. We can see that several time windows such as (3,4), (6,7), (8,9) and (10,11) look pretty similar. On the other hand, some others such as (1,2), (2,3), (12,13) and (15,16) show highly different patterns.

Figure 5 shows the corresponding degree of changes. In the figure, x-axis shows a pair of adjacent windows compared to one another, and y-axis shows the calculated Δ for the associated pair of windows (normalized). As shown in the figure, pairs of windows of (1,2), (4,5), (5,6), (9,10), (12,13) and (15,16) show relatively high Δ's, whereas windows of (3,4), (6,7), (8,9), (10,11) and (13,14) reported minor changes, agreeing in overall with the results of clustered patterns. Although not adjacent, two windows of 1 and 4 look alike in Figure 4, and its calculated degree of changes is fairly low with $\Delta_{(1,4)} = 0.34$ (not shown in Figure 5). Also, the windows of (5,11) has a moderate difference of $\Delta_{(5,11)} = 0.42$, supporting the soundness of the cluster-based pattern representation for monitoring.

The figure also includes a plot (“3 attributes”) with an additional variable of the number bytes in flows, showing concurring trends with the other (“2 attributes”). We observed very negligible variations ($\sigma \approx 0$) in this experiment with a relatively large number of flows in each window, indicating that the impact of randomness introduced by the K-means algorithm would not be significant with a sufficiently large number of samples.

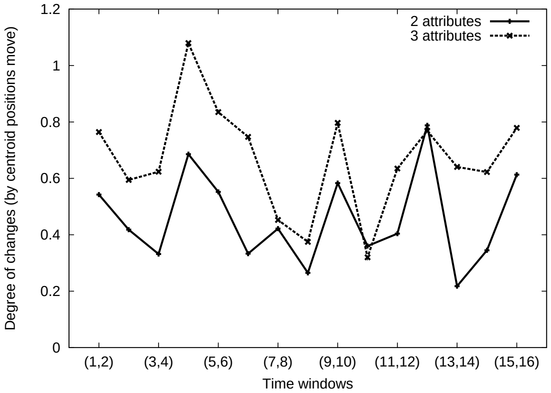

Figure 5: Degree of changes (Δ) based on the centroid positions move with the trace from 8AM to 9:20AM on Oct. 2, 2009. All the observed variances are negligible and almost zeros ($\sigma \approx 0$).

5. CONCLUSION

Network traffic monitoring has taken an increasing attention for effective network management and security with the ever increasing reliance on networked systems and applications. In this paper, we proposed a novel method for effective network traffic monitoring using the clustering technique with the powerful capability for combining multivariate attributes in a straightforward manner. We examined the feasibility of the clustering-based traffic monitoring with visualization for intuitive analysis, and presented our analysis on the clustering results with the compositions of applications. In addition, we defined a new measure of degree of changes to quantitatively evaluate the similarity between two compared network states, and showed that the new measure yields overall agreeable results with the visual patterns, implying that the new measure can be a helpful supportive means for better understanding network states.

This work is in-progress, and there exist many interesting challenges needed further exploration. In this study, we employed centroid positions for estimating degree of changes between two time windows, but additive information could be considered for this measure, such as radius-related variables for taking the size of clusters into account. The public traces we have used in this work were collected in a local area network, and the traffic volume and data frequency can be much greater in a large-scale setting. In such an environment, real-time monitoring with clustering could be a challenging task, and we would like to develop a method for the scalable monitoring service. Another piece of future exploration is about the impact of monitored variables. In this work, we have used flow-level statistics due to the availability of the corresponding groundtruth data. However, other types of attributes such as packet-level information could also be helpful for defining network states, and investigating extensive sets of variables would be necessary to see their impacts.

6. ACKNOWLEDGMENTS

This work was supported by the Office of Advanced Scientific Computing Research, Office of Science, the U.S. Dept. of Energy, under Contract No. DE-AC02-05CH11231.

7. REFERENCES

[1] IPv4 and IPv6 AS Core: Visualizing IPv4 and IPv6 Internet Topology at a Macroscopic Scale in 2014. `http://www.caida.org/research/topology/as_core_network/2014/`.

[2] Kolmogorov-Smirnov Goodness-of-Fit Test. `http://www.itl.nist.gov/div898/handbook/eda/section3/eda35g.htm`.

[3] NORSE Live Attack Map. `http://map.norsecorp.com/v1/`.

[4] L. Bernaille, R. Teixeira, I. Akodkenou, A. Soule, and K. Salamatian. Traffic classification on the fly. *SIGCOMM Comput. Commun. Rev.*, 36(2):23–26, Apr. 2006.

[5] J. Choi, K. Hu, and A. Sim. Relational dynamic bayesian networks with locally exchangeable measures. *LBNL Technical Report, LBNL-6341E*, 2013.

[6] A. Dainotti, A. PescapÃĺ, and K. Claffy. Issues and future directions in traffic classification. *IEEE Network*, 26(1):35–40, Jan 2012.

[7] S. Das, S. Antony, D. Agrawal, , and A. E. Abbadi. Cots: A scalable framework for parallelizing frequency counting over data streams. In *IEEE International Conference on Data Engineering (ICDE)*, pages 1323–1326, 2009.

[8] M. Datar, A. Gionis, P. Indyk, and R. Motwani. Maintaining stream statistics over sliding windows. In *ACM-SIAM symposium on discrete algorithms*, pages 635–644, 2002.

[9] M. Dusi, A. Este, F. Gringoli, and L. Salgarelli. Using GMM and svm-based techniques for the classification of ssh-encrypted traffic. In *Proceedings of IEEE International Conference on Communications, ICC*, pages 1–6, 2009.

[10] J. Erman, M. Arlitt, and A. Mahanti. Traffic classification using clustering algorithms. In *Proceedings of the 2006 SIGCOMM Workshop on Mining Network Data*, MineNet '06, pages 281–286, New York, NY, USA, 2006. ACM.

[11] S. Guha, N. Koudas, and K. Shim. Data-streams and histograms. In *ACM symposium on Theory of computing*, pages 471–475, 2001.

[12] P. Haffner, S. Sen, O. Spatscheck, and D. Wang. Acas: automated construction of application signatures. In *Proceedings of the 2005 ACM SIGCOMM workshop on Mining network data*, MineNet '05, pages 197–202, 2005.

[13] K. Hu, A. Sim, D. Antoniades, and C. Dovrolis. Estimating and forecasting network traffic performance based on statistical patterns observed in SNMP data. In *Machine Learning and Data Mining in Pattern Recognition*, pages 601–615, 2013.

[14] M. Iliofotou, P. Pappu, M. Faloutsos, M. Mitzenmacher, S. Singh, and G. Varghese. Network monitoring using traffic dispersion graphs (tdgs). IMC '07, pages 315–320, 2007.

[15] A. Justel, D. Pena, and R. Zamar. A multivariate kolmogorov-smirnov test of goodness of fit. *Statistics & Probability Letters*, 35:251–259, 1997.

[16] T. Karagiannis, K. Papagiannaki, and M. Faloutsos. Blinc: Multilevel traffic classification in the dark. *SIGCOMM Comput. Commun. Rev.*, 35(4):229–240, Aug. 2005.

[17] H. Kim, K. Claffy, M. Fomenkov, D. Barman, M. Faloutsos, and K. Lee. Internet traffic classification demystified: Myths, caveats, and the best practices. In *Proceedings of the 2008 ACM CoNEXT Conference*, CoNEXT '08, pages 11:1–11:12, New York, NY, USA, 2008. ACM.

[18] J. Kim, A. Chandra, and J. B. Weissman. Using data accessibility for resource selection in large-scale distributed systems. *IEEE TPDS*, 20(6):788–801, 2009.

[19] S. Lee, H. Kim, D. Barman, S. Lee, C.-k. Kim, T. Kwon, and Y. Choi. Netramark: A network traffic classification benchmark. *SIGCOMM Comput. Commun. Rev.*, 41(1):22–30, Jan. 2011.

[20] G. S. Manku and R. Motwani. Approximate frequency counts over data streams. In *VLDB*, pages 346–357, 2002.

[21] G. A. Mills-Tettey, A. Stentz, and S. B. Dias. The dynamic hungarian algorithm for the assignment problem with changing costs. Technical report, Carnegie Mellon University, East Lansing, Michigan, July 2007.

[22] R. Motwani and P. Raghavan. Randomized algorithms. In *Cambridge University Press*, 1995.

[23] T. J. O'Neilla and S. E. Sterna. Finite population corrections for the kolmogorovâĂŞsmirnov tests. *Journal of Nonparametric Statistics*, 24(2):497–504, 2012.

[24] S. Papadimitriou, J. Sun, and C. Faloutsos. Dimensionality reduction and forecasting on streams. *Data Streams, Models and Algorithms*, 31:261–288, 2007.

[25] F. Rgringoli, L. Salgarelli, M. Dusa, N. Cascarano, F. Risso, and k claffy. Gt: picking up the truth from the ground for internet traffic. *ACM SIGCOMM Computer Communication Review*, 39(5), October 2009.

[26] S. Shin and G. Gu. Cloudwatcher: Network security monitoring using openflow in dynamic cloud networks. In *Proceedings of the 2012 20th IEEE International Conference on Network Protocols (ICNP)*, ICNP '12, pages 1–6, Washington, DC, USA, 2012. IEEE Computer Society.

[27] R. Wolski, N. T. Spring, and J. Hayes. The network weather service: A distributed resource performance forecasting service for metacomputing. *Future Gener. Comput. Syst.*, 15(5-6):757–768, Oct. 1999.

[28] G. Xie, M. Iliofotou, R. Keralapura, M. Faloutsos, and A. Nucci. Subflow: Towards practical flow-level traffic classification. In *Proceedings of the IEEE INFOCOM*, pages 2541–2545, 2012.