Towards the Improvement of Performance Anomaly Prediction

Marat Zhanikeev*, Yoshiaki Tanaka*†,
* Global Information and Telecommunication Institute, Waseda University,
1-3-10 Nishi-Waseda, Shinjuku-ku, Tokyo, 169-0051 Japan
Email: maratishe@asagi.waseda.jp
† Advanced Research Institute for Science and Engineering, Waseda University
17 Kikuicho, Shinjuku-ku, Tokyo, 162-0044 Japan

Abstract— **Growing demand for pro-active abilities in network management requires performance monitoring agents not only to be able to monitor the anomalies, but also to predict future occurrences. Recent research in this area would usually apply a neural network algorithm on raw SNMP or NetFlow data to obtain the knowledge about the patterns in performance data. The results are not always satisfactory due to highly unpredictable nature of cross-traffic in the network. This paper attempts to improve the prediction quality by using data obtained from end-to-end probing. The results prove higher resilience to cross-traffic interference and better pattern recognition.**

I. Introduction

The issue of realtime management has long been the subject of many research works [1] [2]. This research target brings additional complications to conventional management problems, such as scalability through distribution and data aggregation [3], alternative sources of performance data [4], and lightweight monitoring and management infrastructure [5]. Real-time constraints pose a few rather unyielding requirements, such as (1) fast data collection without loss of important artifacts, (2) online processing with a lightweight calculation algorithm, and (3) prompt decision making.

Conventionally the above constraints are met using a standard set of available monitoring technology, such as widely used SNMP and NetFlow. Both technologies are commonly shipped together with network equipment and work out of the box.

To equip operation center with pro-active abilities, it is common to use neural network algorithms with SNMP or NetFlow data used as input. The output of a neural algorithm is a set of patterns extracted from data, which may help to predict future occurrences of similar anomalies.

However, due to intrinsic features of both SNMP and NetFlow techniques, as well as because of interference with highly unpredictable cross-traffic, predictions are made with high error rate, and, therefore, are not reliable.

We do believe that the main reason of such poor performance is not due to cross-traffic interference, but rather because of limited ability to properly correlate multiple data sources when using SNMP or NetFlow. Apart from correlation error, there are also errors introduced in the process of communication between operating center and each monitored entity. As SNMP, for example, is mostly based on recursive counters, each dynamic metric has to have two separate readings. That means that there should be 4 one-way communication sessions for each metric. As the same network is usually used for such communications, cross-traffic brings additional error in SNMP and NetFlow readings and timings of alarms.

In this paper, we propose to use end-to-end probing as an alternative source of performance data. End-to-end measurements are already widely employed in end-to-end performance discovery [4], but, as far as our knowledge extends, end-to-end data has not been tried with neural networks for anomaly predictions.

Section 2 offers more detailed insight into available performance data sources. Section 3 offers comparative results based on raw performance data, while Section 4 goes even further and introduces a pre-filtering scheme based on simple rulesets. Comparison of the results proves that end-to-end data offers “cleaner” view of network performance as well as much higher validity of predictions.

0-7803-9179-9/05/$20.00 ©2005 IEEE.

II. Performance Data Sources

A. SNMP and NetFlow Data

Even though SNMP and NetFlow are very different technologies, they share one common feature, which is the fact that they both operate locally at each monitored entity. SNMP performs in accordance with Management Information Base (MIB) description, which tells the SNMP agent how and which data about performance to collect. NetFlow operates in accordance with a ruleset, that can define very small particulars of data collection.

As SNMP and NetFlow only offer data pertaining to each particular monitored entity, overall performance picture of a domain requires correlation of data from a number of SNMP or NetFlow sources. Timely and precise aggregation and correlation of data poses a number of research problems addressed in currently ongoing research.

For the tests within this paper we use readings of ifInOctets counter from SNMP RMON MIB, which stands for the number of bytes transmitted by router interface, as indicator of utilization.

B. Data Obtained by Probing

Quite differently from SNMP and NetFlow, end-to-end measurements offer data that are already automatically correlated over a number of entities included in the path. Additional advantage of end-to-end probing is the flexibility of the probe inserted into the network. Fig.1 displays the probing method used by this paper.

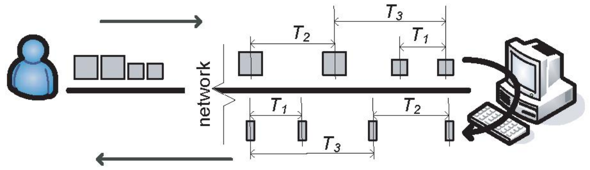


Fig. 1. Probing method with specifically designed probe structure.

Talking about probe flexibility, by using a special probe structure in Fig.1, we can elicit a number of independent metrics:

$$M_1 = \frac{S_1}{T_1}, M_2 = \frac{S_2}{T_2} \tag{1}$$

Since first packet pair uses smaller size S_1 packets than the piggy-backed pair (S_2), pairs traverse the network at different speeds, and metrics M_1 and M_2 can be considered fairly independent. The third metric that we use is $M_3 = T_3$, which is simply the interval between pairs in the probe at the time of arrival.

Probing itself is conducted in round-trip manner, with the opposite end replying to UDP packets by small packets that serve as acknowledgment for probing source. This way no synchronization is required between probing ends.

III. Predictions Using Raw Performance Data

A. Network Model

For testing we use a middle-size network generated using SSFNET simulator [6]. It offers good performance for large networks and contains implementations of NetFlow and SNMP at each node.

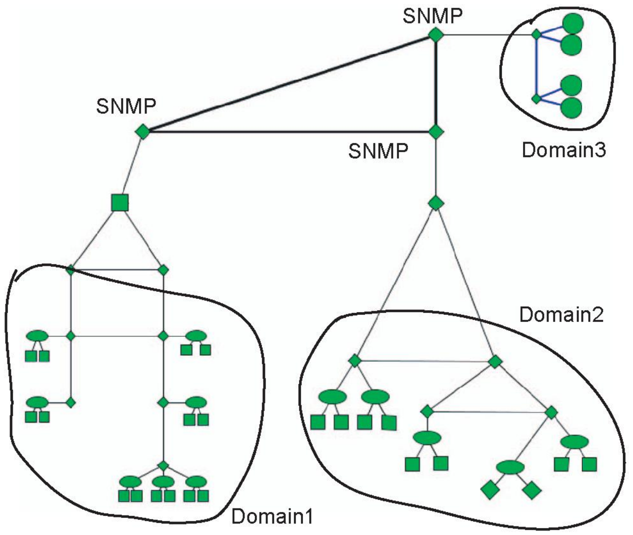


Fig. 2. Network model used for simulation.

Network applied for tests is displayed in Fig.2. We generated three separate domains with three backbone routers among them. Number of hosts in each domain is also random. Pareto traffic is generated among all the domains in such a manner that backbone links are utilized at the average rate of 20-30%.

End-to-end measurements are performed so that the path comprises two of three backbone nodes, and SNMP statistics are polled from a single backbone router selected randomly. For the sake of simplicity, polls are collected directly without communication. That allows us to consider that SNMP statistics only represent the actual trends in the traffic.

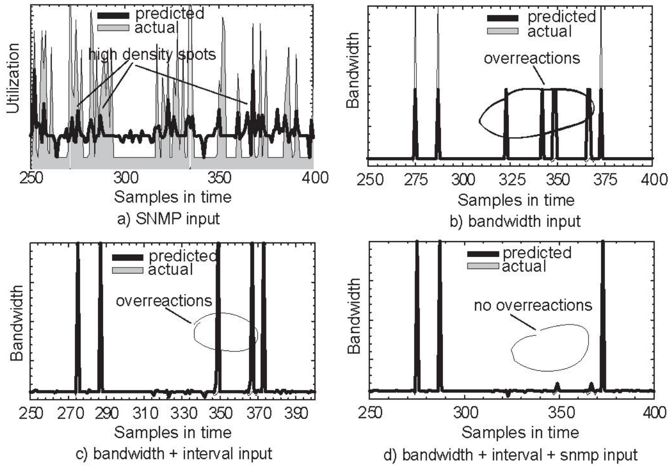
B. Neural Network Setup

Neural networks require training period to extract existing patterns from available data. Similarly, in the present study we use backprop neural algorithm and split the data into training and runtime parts. Training part is used by neural network for retrieving patterns and matching them against the actual outcome, while the runtime part is used for visual matching of pattern-based predictions and actual outcome. For each test we split the data into five equal parts, each consisting of 60 seconds of collected data. Training-to-runtime ratio for SNMP data is 75%, while we use only one third of samples for training in case of end-to-end probing data.

The input data is also different for SNMP and end-to-end probing. For the lack of other data, in case of SNMP we apply commonly used ifInOctets counter readings. On the other hand, special structure of the probe allows us to capture more metrics in parallel, and here we use combinations of metrics, as specified in figures.

C. Simulation Results

Comparative results of predictions based on SNMP only data and combinations including end-to-end probing are displayed in Fig.3. It could be visually confirmed that SNMP data is, by definition, very noisy. Short-term drastic changes in its statistics do not necessarily stand for a change in performance, although some weak relation exists nevertheless. However, this intrinsic feature of SNMP counters makes pattern recognition very difficult, which can be seen from the prediction line in the figure. Although major changes in the variance of prediction amplitude match higher density areas in actual data, the amplitude is not the same, which indicates high prediction error. Even if a threshold were applied to predictions, even filtered outcome would not offer any reliable data.



a) SNMP input

b) bandwidth input

c) bandwidth + interval input

d) bandwidth + interval + snmp input

Fig. 3. Predictions using various data source combinations, with 1000 and 1400 byte packets in active probes.

On the other hand, end-to-end probing-based predictions in Fig.3 are much more accurate. End-to-end probing is a natural way to smoothen performance data, which is the reason for much smaller amount of spikes in the actual data. Predictions, too, are much more accurate, with only a few "overreactions" in Fig.3(b). We call it overreaction, when the amplitude of prediction does not match the amplitude in the actual data at the same point in time.

Using combination of M_1, M_2, and M_3 as input in Fig.3(c) results in even more improved performance, with less overreactions. The best case, however, was found in Fig.3(d), where we created a combination of end-to-end probing and SNMP raw data. Obviously, the patterns from SNMP became clearer with the help of end-to-end metrics, as predictions in this case have displayed the lowest best error rate.

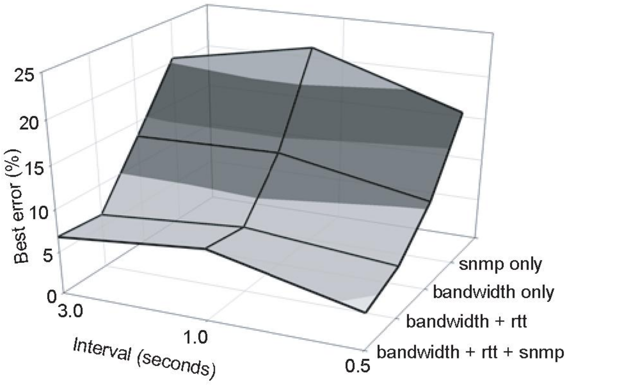


Fig. 4. More thorough best error performance comparison at various polling and probing intervals.

This trend, however, did not apply to all test cases in our study. The results in Fig.3 are obtained by applying the data from measurements and polls made at 0.5 second intervals. This case of only a subset in Fig.4, which presents prediction performance over a wider set of polling and probing frequencies. While the previously mentioned case of 0.5 second interval exhibits high level of dependence on the combination of input data, some grouping is possible with more sparse data polls.

For example, the considerable improvement of predictions with SNMP+probing mix is not found with higher values of interval. To generalize, we can state that with higher values of interval SNMP data becomes less valuable for prediction, while end-to-end probing data remains usable.

IV. Predictions After Ruleset-Based Correlation

A. Rulesets as Binary Filters of Input Data

To deal with the noisy data problem experienced in Fig.3(a), we considered using threshold-based rulesets to filter unwanted data. The outcome of the ruleset is binary, and, therefore, the neural network operates on a sequence of mixed 0 and 1 entries.

We tried two rulesets of increasing complexity in Fig.5 and Fig.6 for SNMP data, and only one simple ruleset in Fig.7 for end-to-end probing.

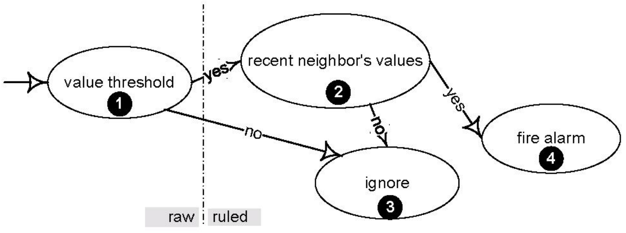


Fig. 5. Simple ruleset used for SNMP data.

Logic of SNMP ruleset in Fig.5 is a simple neighbor correlation algorithm. When the threshold is surpassed in (1), the algorithm checks in (2) whether the current counter reading for the same variable on a neighboring node experiences the same condition. The alarm, which is the binary 1, is fired only in case the test in (2) yields positive result, or, otherwise, the data is discarded.

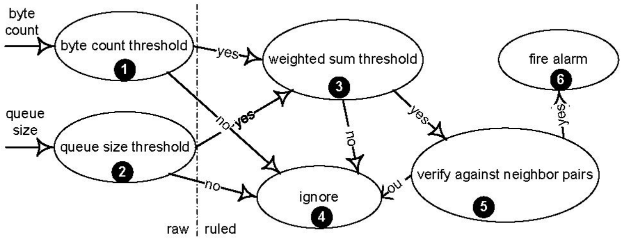


Fig. 6. More complex ruleset applied to SNMP data.

The ruleset in Fig.6 is similar to Fig.5 in logic, but has one additional input metric. Therefore, within the ruleset, the behavior of utilization and queue size are correlated, and alarm is fired only if changes in both metrics are simultaneous.

Ruleset for probing in Fig.7 has a different design, and imposes deviation threshold at the entrance in (1) and (2) for measurements received from both pairs in the probe. We remember the last 10 samples for each metric to calculate the variance. After the deviation threshold is successfully passed by both metrics their weighted sum is exposed to another threshold, the positive outcome of which results in firing the alarm. For the present study we give equal weights to both inputs.

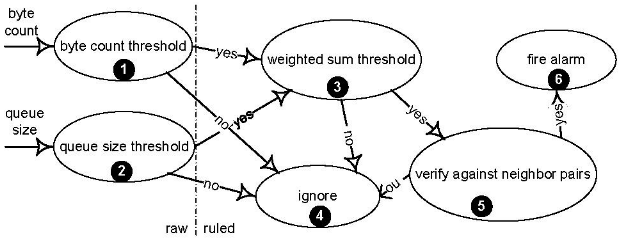


Fig. 7. Ruleset applied by probes for filtering alarms.

B. Simulation Results

Fig.8 displays the three prediction cases, one raw and two with the above SNMP-based rulesets. The effectiveness of rulesets is obvious, as less than 10% of alarms pass the rulesets. However, the value of the outcome of rulesets is still questionable, as, after being matched to prediction results, averagely only half of alarms are correctly predicted. It is doubtful that predictions with 50% error rate can be considered by any pro-active management system.

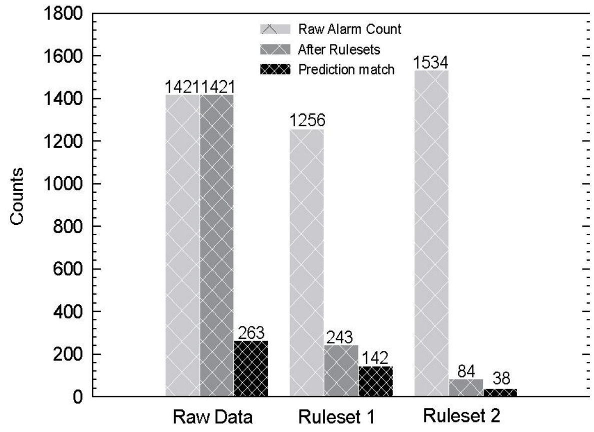


Fig. 8. Comparison of performance with and without rulesets applied to SNMP data.

On the other hand, rulesets, when used to probing data, result in improvement of prediction hits in Fig.9. Two combinations of raw data (bandwidth from one pair and from both pairs) still exhibit almost 20 wrong binary predictions. When the ruleset is applied, the number of alarms is three times smaller and only 2 out of 21 predictions are wrong. The physical meaning of this result is that ruleset only leaves the most important and correlated performance anomalies, while preserving the pattern of their distribution in time.

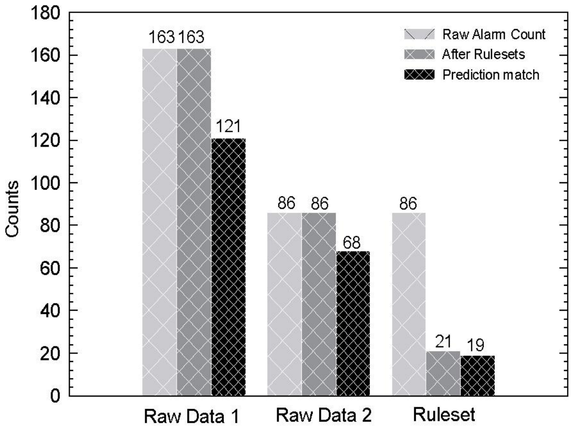


Fig. 9. Performance comparison with and without rulesets applied to probing data.

To compare the performance of SNMP and probing-based predictions, we should note that changes in SNMP-based results from application of rulesets are mostly quantitative, while the ratio of total alarms to predicted alarms remains averagely the same. In the case of probing, the change is in the quality, as filtered data is continuously improving with the decrease of total number of alarms, i.e. the most important alarms are last to be filtered out of the data.

V. Conclusion

In this paper we attempted to deal with the issue of low quality to performance predictions used in pro-active management. Such predictions are conventionally based on raw readings from either SNMP agents, or NetFlow messages. SNMP and NetFlow have one feature is common, which is the fact that both have to be dispatched to the actual monitored entity.

As performance, by definition, is attributed not to a single network element, but rather to a path or network domain, we assumed that end-to-end metrics obtained through active probing would be much more able to represent the performance of the network.

To test our assumption, we replaced the SNMP input into neural network algorithm by the data obtained from end-to-end probing. Comparison of the results proved that our assumption is correct in that probing-based data contains more evident performance patterns than raw SNMP data.

Knowing well the common noisiness attributed to SNMP polls, we conducted another test to compare performance of SNMP and probing-based predictions using only binary input. To obtain binary data we applied rulesets to the input, the outcome of which was a time sequence of 1 and 0 entries.

The rulesets helped to rectify the problem of noise and reduce the number of false SNMP alarms, but we discovered that patterns in filtered data were still not strong, and only half of predictions were correct. On the other hand, ruleset-based filtering of measurement data resulted in improved quality of probing-based alarms, and further decreased the number of false alarms.

As in this paper we performed simulation test with near ideal condition for SNMP polls, practical verification of the findings is required. As all SNMP polls have to be transmitted over the network and, therefore, interfere with the cross-traffic, we can assume that the real network results for SNMP will be even worse than those presented in this paper. Probing in this simulation study, however, was very close to authentic, and we do not expect any major discoveries during real network tests.

References

[1] S. Gupta, “Deriving network management methods,” Institute of System Research, Tech. Rep. CSHCN TR 1997-23, 1997.

[2] J. L. Alberi, T. Chen, S. Khurana, A. Mcintosh, M. Pucci, and R. Vaidyanathan, “Using real-time measurements in support of real-time network management,” in *Passive and Active Measurement Workshop*, April 2001, pp. 234–242.

[3] Y. Lin and M. Chan, “A scalable monitoring approach based on aggregation and refinement,” in *IEEE Journal on Selected Areas in Communications*, vol. 20 no. 4, May 2002, pp. 677–690.

[4] A. W. Moore, “Measurement-based management of network resources,” Universtity of Cambridge, Tech. Rep. UCAM-CL-TR-528, April 2002.

[5] K. G. Anagnostakis, S. Ioannidis, S. Miltchev, and J. M. Smith, “Practical network applications on a lightweight active management environment,” *Lecture Notes in Computer Science*, vol. 2207, pp. 101–105, 2001.

[6] “SSFNET : Scalable simulation framework,” available at : http://www.ssfnet.org/, January 2004.