Scalable Monitoring via Threshold Compression in a Large Operational 3G Network

Suk-Bok Lee[1], Dan Pei[2], MohammadTaghi Hajiaghayi[2,3], Ioannis Pefkianakis[1], Songwu Lu[1]
He Yan[4], Zihui Ge[2], Jennifer Yates[2], Mario Kosseifi[5]
[1]UCLA Computer Science [2]ATT Labs–Research [3]University of Maryland
[4]Colorado State University [5]AT&T Network Services

ABSTRACT

Threshold-based performance monitoring in large 3G networks is very challenging for two main factors: *large network scale* and *dynamics in both time and spatial domains*. There exists a fundamental tradeoff between the size of threshold settings and the alarm quality. In this paper, we propose a scalable monitoring solution, called threshold-compression that characterizes the tradeoff via intelligent threshold aggregation. The main insight behind our solution is to identify groups of network elements with similar threshold behaviors across location and time dimensions, thus forming spatial-temporal clusters and generating the associated compressed thresholds within the optimization framework. Our evaluations on a commercial 3G network have demonstrated the effectiveness of our threshold-compression solution, e.g., threshold setting reduction up to 90% within 10% false/miss alarms.

Categories and Subject Descriptors: C.2.3 [Computer-Communication Networks]: Network Operations

General Terms: Measurement, Algorithms

1. INTRODUCTION

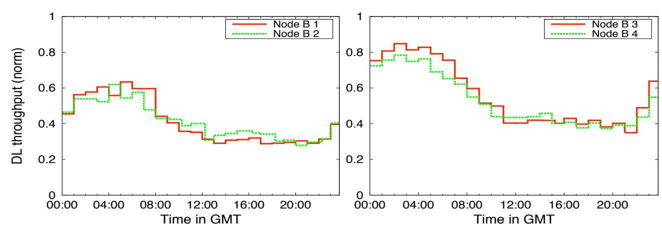
The current practice for monitoring the health of a large-scale network is to use pre-defined thresholds of selected key performance indicator (KPI) metrics. However, direct application of such a pre-computed, threshold-based alarming model does not scale in 3G networks due to the two main factors: (1) massive data volume and large network scale; (2) rich dynamics in both time and spatial domains. A single static threshold per KPI fails to capture such spatial and temporal dynamics, leading to unacceptably poor alarm quality with nearly 70% false positives/negatives. On the other hand, a finer-grained location- and time-dependent threshold setting can capture network dynamics but incurs prohibitively high system management complexity. The number of thresholds to be maintained grows very large with the increasing number of network elements (NEs) and the time granularity. For example, given that one regional area has about 5,000 cells and 30 KPIs, the per-NE hourly threshold scheme has as many as $5K \times 24 \times 30 = 3.6$ million thresholds in a single area. Therefore, it is increasingly difficult to monitor an operational 3G network with naive pre-defined threshold scheme. To this end, we propose a scalable threshold-based solution, called threshold-compression, which has both merits of a small number of used thresholds and accurate capturing of spatial-temporal network dynamics.

SIGMETRICS'11, June 7–11, 2011, San Jose, California, USA.
ACM 978-1-4503-0262-3/11/06.

2. THRESHOLD COMPRESSION

We describe threshold-compression by highlighting the motivation, problem formulation, and compression algorithm suite.

Case for similar threshold behavior. Our threshold compression approach is motivated by two key observations: (1) threshold behavior similarity among a certain group of NEs, and (2) stable/close threshold trends over some period of time. Figure 1 shows example NE-pairs on downlink-throughput KPI using per-NE-hourly thresholds. Such spatial similarity is attributed to the geographic locations of NEs and the user population in the corresponding area. For example, NEs in urban (/rural) areas are likely to have similar high (/low) dynamics over time. Time-domain similarity is also observed, as each NE is likely to have similar high (/low) demand during peak (/sleep) hours. For example in the figure, each NE-group shows very stable threshold behavior during peak hours between 11:00 GMT and 22:00 GMT, which provides us an opportunity to form a temporal-domain cluster.



(a) NodeB1 and NodeB2 (b) NodeB3 and NodeB4

Figure 1: Downlink-throughput KPI: similar threshold (per-NE-hourly) behavior among different Node Bs.

Desirable properties of threshold compression. To ensure scalable monitoring performance as well as practical threshold management, threshold-compression should have the following properties: (1) High compression gain: The resulting threshold setting should remain small even with a large number of NEs; (2) Low false alarm rate: The compressed thresholds must result in good alarm quality, i.e, low false positive rate (FPR) and false negative rate (FNR), and thus, we use a concept of *threshold closeness* of lower ($T^{i,j}_{lower}$) and upper ($T^{i,j}_{upper}$) bound to approximate the per-NE-hourly thresholds of (NE i and hour j); (3) Management-oriented grouping policy: The spatial-temporal clusters must be easy to manage and update in the monitoring system. To this end, we employ a consistent NE grouping policy where each NE can belong to only one NE group (but there can be multiple hour groups within an NE group), hence a two-level hierarchical clustering structure.

Problem formulation. We formulate the threshold compression problem taking the alarm quality as well as the required clustering policy into account. The objective is to find the minimum number of spatial-temporal clusters (or equivalently the minimum threshold setting) from a given fine-grained threshold setting with the following constraints: (1) Each compressed threshold must be within the *permissible* threshold interval of $T^{i,j}_{lower}$ and $T^{i,j}_{upper}$; (2) NE grouping must be consistent in time; (3) Each cluster must consist of continuous time steps (optional rule).

It turns out that this problem is not only NP-hard (regardless of the optional rule) but indeed it is very hard to approximate as well. The proof is given in the full version of the paper [1].

Threshold compression algorithm suite. Our threshold-compression takes a two-staged approach. We first decouple the spatial NE grouping from the original two-dimensional clustering problem, then further proceed with temporal-domain clustering within each identified NE group. Our key strategy for clustering is to combine spatial-temporal blocks if they (i) have common intersection in their permissible intervals, and (ii) meet the consistent NE grouping rule. Note that having common intersection among the cluster members ensures the satisfying alarm quality.

1. NE grouping: greedy coloring approach. The first stage identifies NE groups each showing similar threshold behavior each hour among its members. As the first-level of clustering hierarchy, each NE group, in fact, consists of 24 hour-groups, which will be compressed further in the next stage via time-domain clustering. Then, the NE grouping problem naturally reduces to the graph coloring that asks the minimum number of colors (NE groups) assignable to each vertex (NE) such that no edge (common intersection) connects two identically colored vertices (group members). This graph coloring instance is NP-hard, and we employ a greedy coloring heuristic, which works quite well in practice. Specifically we apply the Welsh-Powell algorithm [2] that uses at most $\max_i \min\{d(v_i)+1, i\}$ colors, that is at most one more than the maximum degree of the graph. We convert our problem instance to a graph $G(V,E)$, where each NE corresponds to a vertex in G. For each vertex pair v_i and $v_{i'}$, we put an edge between them if their counterpart NEs i and i' have disjoint threshold intervals in any hour. Then the vertices colored γ (by the greedy coloring algorithm) can be readily transformed to NE-group γ in our problem.

Once identified, each NE group γ defines its own permissible threshold interval to reflect each member's interval. Setting the group threshold interval to the common intersection among the members makes the next-stage clustering procedure to keep control on the resulting alarm quality.

2. Hour grouping: minimum cover selection. As the next level of the clustering hierarchy, the time-domain clustering takes the NE grouping result as input to perform the hour grouping for each identified NE-group. Within NE group γ, there are initially 24 hour-groups, each of which we simply refer an hour. Then each hour j is represented by its threshold interval $\Phi^{\gamma,j}_{lower}$ and $\Phi^{\gamma,j}_{upper}$ (i.e., the common intersection among all members at hour j) as a result of NE grouping. Given the set of intervals, the hour grouping problem is to find the minimum number of interval groups such that (i) each interval belongs to one of the interval groups, and (ii) there is common intersection in each interval group. We use a simple greedy algorithm that leads to an optimal solution to this problem. The algorithm is as follows. We first sort all the interval endpoints ($\forall j \in H : \Phi^{\gamma,j}_{lower}, \Phi^{\gamma,j}_{upper}$) in ascending order of their values. We scan the list (in ascending order) until first encountering an upper-bound point $\Phi^{\gamma,j'}_{upper}$. We then put all intervals containing this point (i.e., all hours $j : \Phi^{\gamma,j}_{lower} \leq \Phi^{\gamma,j'}_{upper}$) into a new interval group C'_h, and delete them from the list. We repeat this process until there is no interval in the list. This simple greedy rule indeed finds the minimum number of interval groups, hence the minimum hour groups. The proof is given in the full version of the paper [1].

Now, all hours in each identified interval group C'_h of NE group γ can form a spatial-temporal cluster C''_δ. In order to preserve the threshold-closeness property for all members, we set the compressed thresholds $T_{comp}(\delta)$ within the common intersection across all NEs $i \in C_\gamma$ and hours $j \in C'_h$ in the spatial-temporal cluster, and we use the median point in this study. We again note that this compressed thresholds $T_{comp}(\delta)$ is shared by all NEs and hours in C''_δ, thus reducing the threshold setting while still preserving the location and time specific thresholds.

3. EVALUATION

We evaluate the performance of threshold-compression on the data recorded from June 2010 to August 2010 in one regional 3G network that covers several thousands of NEs. Figure 2 shows the threshold compression gain on different KPIs. The compression gain is defined as the threshold-setting reduction relative to the fine-grained per-NE-hourly setting. Each compression gain in the figure represents the highest threshold-compression gain observed when the resulting false/miss alarm rates FPR and FNR (based on the per-NE-hourly alarm statistics) are both within 10% (and 20%) range. We observe that, within 10% false/miss alarm condition, most KPIs show very high compression gain nearly 80–90%. Tables 1 compare the threshold-setting sizes and false/miss alarm rates produced by different thresholding schemes. Our approach balances very well the problematic tradeoff between the threshold setting and the alarm quality, while other schemes are unable to achieve both.

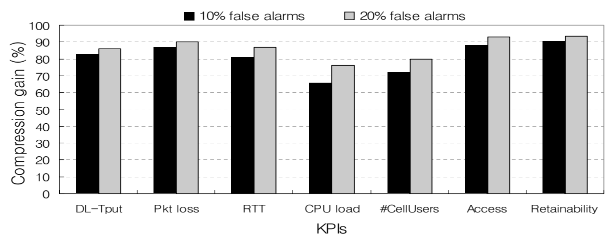


Figure 2: Compression gain on different KPIs.

Threshold scheme	#thresholds	FPR	FNR
per-NE-hourly	**25320**	-	-
threshold-compression	3763	8.4%	2.7%
per-NE-static	1055	**31.1%**	**51.8%**
per-NEtype-hourly	24	**51.2%**	**47.5%**
per-NEtype-static	1	**53.2%**	**58.0%**

Table 1: Thresholding on downlink-throughput KPI.

4. CONCLUSION

Motivated by key observations of spatial-temporal threshold similarity, we have proposed a scalable monitoring solution, called threshold-compression that can characterize the location- and time-specific threshold trend of each individual NE with minimal threshold setting. Our experience with applying our threshold-compression solution in the operational 3G network monitoring has been very positive, and demonstrated the effectiveness of the proposed approach, e.g., threshold setting reduction up to 90% within 10% false/miss alarms.

5. REFERENCES

[1] S.-B. Lee, D. Pei, M. Hajiaghayi, I. Pefkianakis, S. Lu, H. Yan, Z. Ge, J. Yates, M. Kosseifi. Scalable monitoring via threshold compression in a large operational 3G network. *AT&T Technical Report*, 2011.

[2] D. Welsh and M. Powell. An upper bound for the chromatic number of a graph and its application to timetabling problems. *Computer Journal*, 85–86, 1967.