

Distributed Real-time Monitoring with Accuracy Objectives

Alberto Gonzalez Prieto, Rolf Stadler

School of Electrical Engineering
KTH Royal Institute of Technology, Sweden
{gonzalez,stadler}@ee.kth.se

Abstract. We introduce A-GAP, a protocol for continuous monitoring of network state variables with configurable accuracy. Network state variables are computed from device counters using aggregation functions, such as SUM, AVERAGE and MAX. In A-GAP, the accuracy is expressed in terms of the average error and is controlled by dynamically configuring filters in the management nodes. The protocol follows the push approach to monitoring and uses the concept of incremental aggregation on a self-stabilizing spanning tree. A-GAP is decentralized and asynchronous to achieve robustness and scalability. We provide some results from evaluating the protocol for an ISP topology (Above-net) in several scenarios through simulation. The results show that we can effectively control the fundamental trade-off between accuracy and overhead. The protocol overhead can be reduced significantly by allowing only small error objectives.

1. Introduction

The ability to provide continuous estimates of management variables is vital for management tasks, such as network supervision, quality assurance, and proactive fault management. Generally, management variables that are monitored in these tasks are aggregates that are computed from device variables across the network using functions such as SUM, AVERAGE, MIN, MAX. Sample aggregates are the total number of VoIP flows in a network domain and the maximum link utilization.

For many management tasks, it is crucial to know how accurate such estimates are. However, network management solutions deployed today usually provide qualitative control of the accuracy, but do not support the setting of an accuracy objective.

Engineering continuous monitoring solutions for network management involves addressing the fundamental trade-off between accurate estimation of a variable and the management overhead in terms of traffic and processing load. Obviously, a high accuracy comes at the cost of a high overhead and, similarly, low accuracy estimation can be achieved with a low overhead. We found this trade-off first discussed in [2]. Since then, several authors addressed this issue as we show in [1].

In this paper, we address the problem of continuous monitoring with accuracy objectives in large-scale network environments. Specifically, we want to achieve an efficient solution that allows us to control the accuracy of the estimation.

The paper introduces A-GAP, a generic aggregation protocol with controllable accuracy. A-GAP is based on GAP (Generic Aggregation Protocol), which allows for continuously computing aggregates of local variables by (i) creating and maintaining a self-stabilizing spanning tree and (ii) incrementally aggregating the variables [1] (fig 1). A-GAP is push-based in the sense that changes in monitored variables are sent towards the management station. The protocol controls the management overhead by filtering updates that are sent from monitoring nodes to the management station. The filters periodically adapt to the dynamics of the monitored variables and the network environment. All operations in A-GAP, including computing the aggregation function and filter configuration, are executed in a decentralized and asynchronous fashion to ensure robustness and achieve scalability.

We developed a stochastic model of the monitoring process, which allows us to compute the filter widths as the solution for the optimization problem of minimizing the management overhead for a given estimation error. A heuristic solution to this problem is implemented in A-GAP.

The paper is organized as follows. Section 2 defines the problem of real-time monitoring with accuracy objectives. Section 3 describes our proposal, A-GAP, which is evaluated in section 4. Section 5 concludes the paper.

2. The Problem: Real-time Monitoring with Accuracy

System architecture. This work assumes a distributed management architecture, whereby each network device participates in the computation by running a management process, either internally or on an external, associated device. These management processes communicate via a *management overlay network* for the purpose of monitoring. We also refer to this overlay as the *network graph*. A node in this graph represents a management process together with its associated network device(s). While the topology of this overlay can be chosen independently from the topology of the underlying physical network, we assume in this paper, for simplicity, that both topologies are the same, i.e., that the management overlay has the same topology as the underlying physical network.

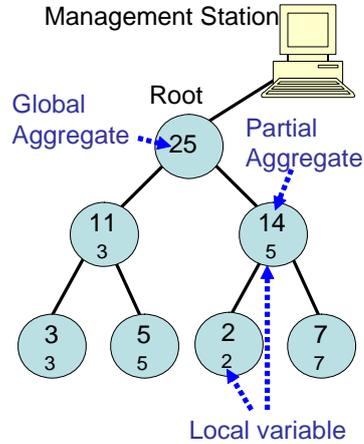


Fig. 1. Example of aggregation tree. Distributed computation of the sum of local variables (w_i)

Problem statement. We consider a dynamically changing network graph $G(t) = (V(t), E(t))$ in which nodes $n \in V(t)$ and edges/links $e \in E(t) \subseteq V(t) \times V(t)$ may appear and disappear over time. Each node n has an associated *local variable* $w_n(t) \geq 0$.

The objective is to engineer a protocol on this network graph that provides a management station with a continuous estimate of $\sum_n w_n(t)$ for a given accuracy. The accuracy is expressed as the *average error* of the estimate over time. The protocol must minimize the (maximum) processing load across all nodes.

Throughout the paper we use SUM as aggregation function. Other functions can be supported as well, as discussed in section 5.

3. A-GAP: a Distributed Solution

A-GAP controls the management overhead and estimation accuracy by modifying filters in the management nodes. A filter reports changes of the local partial aggregate if its new value exceeds the local filter width. The filter widths periodically adapt to the dynamics of the monitored variables and the network environment. The accuracy objective can be dynamically changed from the management station if needed.

We developed a stochastic model of the monitoring process, which includes the dynamics of the local variables, the filter widths, the overhead incurred and the estimation accuracy. Using this model, we express the filter widths as the decision variables for the problem of *minimizing the maximum load across all nodes in the management overlay for a given average error of the estimation of the global aggregate*. A heuristic solution to this optimisation problem is implemented in A-GAP. The model and the heuristics are described in [1].

In the above model, local variables change following independent random walks. This assumption has been made in similar contexts [1], and it facilitates an algorithmic solution. In practice, the parameters of the random walk must be estimated.

Design principles of A-GAP. First, for reasons of scalability and robustness, A-GAP is a decentralized and asynchronous protocol. Although a centralized solution to the above optimisation problem could be achieved using grid search algorithms, such an approach is not feasible, since its computational complexity grows exponentially with the number of nodes.

Second, A-GAP is an extension of the GAP protocol, which provides continuous estimation of global aggregates by creating a spanning tree on the management overlay and incrementally aggregating the local variables on this spanning tree. When running A-GAP, all nodes of the management overlay execute the same code. The root node of the spanning tree holds the current estimate of the global aggregate.

Third, A-GAP realizes a heuristic in which the above global problem is mapped onto a local problem that each node solves. Each node attempts to minimize its processing load for a given accuracy regarding its local aggregate. This is achieved by periodically re-computing local filters based on local information.

Re-computing local filters. Each node periodically executes a control cycle in an asynchronous fashion, as follows. The node starts by polling its children for statistics

related to their partial aggregates. Then, the node re-computes the filters of a subset of its children. The subset is chosen using a round-robin policy, whereby the sets of two consecutive rounds overlap. The new filters are determined by minimizing the local processing load subject to an accuracy objective for the local partial aggregate. This accuracy objective is given by the node's parent. The problem is solved through a grid search, where the search space is limited to small changes of the current filter width. Next, the new accuracy objectives for the children are computed. Finally, the node updates the statistics of its partial aggregate, which will be polled by its parent during the next control cycle.

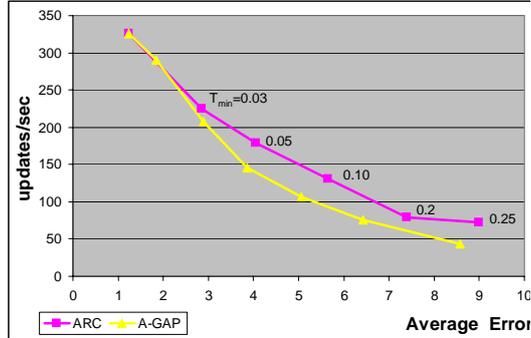


Fig. 2. Management Overhead vs Accuracy for the A-GAP and ARC protocols

4. Evaluation through Simulation

Setup description. We have evaluated A-GAP through extensive simulations using the SIMPSON simulator [4]. The results presented in the paper are based on the topology of Abovenet [5], consisting of 654 nodes and 1332 links. The overlay topology is chosen to follow the physical Abovenet topology. The control cycle of A-GAP is 1 second. The results reported below corresponds to a measurement period of 30 seconds simulation time of the protocol in steady state, which is reached after a warm-up period of approximately 25 seconds.

Accuracy vs Overhead Trade-off. In this simulation scenario, the global aggregate increases at an average rate of about 60 units per second. Figure 2 shows the overhead, i.e., the maximum processing load, as a function of the average error. As can be seen, the overhead decreases monotonically, as the error objective is increased. For small errors, the load decreases faster than for larger errors. As expected, the overhead can be reduced by allowing a larger average estimation error. For example, allowing an error of 3 units reduces the load by 40%, an error of 8 units reduces the load by 85%. (Note that the granularity of a local variable is 1 unit).

Figure 2 also compares the performance of A-GAP against an asynchronous rate-control scheme (ARC). As A-GAP, ARC uses a spanning tree and incremental aggregation to continuously estimate the aggregate of local variables. The control parameter for ARC is the minimum time interval (T_{\min}) for a node to send an update to its parent. In these and other experiments we performed [1], A-GAP incurred a lower overhead than ARC. As expected, both approaches perform similarly for very small estimation errors.

A rate-control approach that would permit to set update intervals individually for each node would probably perform better than ARC. Note though that, while A-GAP allows to quantitatively control the error objective, rate-control approaches do not support this functionality in a straightforward way.

From this simulation data, we also evaluated the difference between the accuracy objective and the measured error. For all measurement points, we found that the measured error is about 1.5 units above the error objective. We explain this by the fact that updates from different nodes in the network experience different delays for reaching the root of the tree. This difference depends on the network topology and delays.

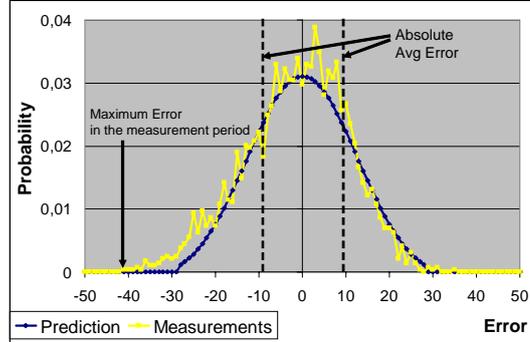


Fig. 3. Distribution of Predicted Errors and Measured Errors at the Root Node.

Distribution of the estimation error. In this simulation scenario, the global aggregate oscillates around a constant value. Figure 3 includes a curve that shows the predicted error based on our stochastic model [1]. The second curve gives the measured error from a simulation run. (The curves correspond to an error objective of 8). A vertical bar indicates the average error.

As we can see, the predicted error distribution is close to the actual distribution. More importantly, the distributions have long tails. While the average error is 9.5, the maximum error in this measurement period is 41 and the maximum possible error (that can occur in an infinite measurement period) is 180. Based on this observation, we argue that an average error objective is more significant for practical scenarios than a maximum error objective, as suggested by other authors (see [1]).

5. Discussion

In this paper, we introduce A-GAP, a protocol for continuous monitoring with accuracy objectives. A-GAP follows the push approach to monitoring and uses the concept of incremental aggregation on a spanning tree. A-GAP is decentralized and asynchronous, two key properties for achieving robustness and scalability.

Although we have used SUM as the aggregation function throughout this paper, other aggregate functions like AVERAGE, MIN and MAX can be supported with straightforward modifications. For instance, AVERAGE can be estimated by maintaining the SUM of the local variables and a node count (obtained using another SUM) at the root.

Our experiments show that we can effectively control the trade-off between accuracy and overhead. A-GAP can reduce the overhead significantly when allowed some error in its estimations.

In A-GAP, accuracy is expressed in terms of the average error, which we argued to be more significant for practical applications than the objective of a maximum error, suggested in the recent literature (see [1]).

To be applicable in practical scenarios, A-GAP requires extensions. Since the model upon which filter re-computation is based does not consider networking and processing delays, the protocol generally misses the error objective by a small margin. In fact, the estimation error of A-GAP exceeds the error objective by a margin that is topology dependent. This means that the protocol needs to be tuned during initialization of the monitoring task.

The model for local filter re-computation uses parameters from the random walk model of the local variables as input. Therefore, these parameters need to be dynamically estimated for each local variable and real-time estimators need to be added to the protocol.

A-GAP enables performance prediction at run-time. Based on our stochastic model, a manager can be provided with an estimation of the expected load on all nodes and the distribution of the estimation error at the root node for a given accuracy objective. This is potentially significant in real scenarios. For instance, a manager could avoid overloading the management system by loosening the error objective.

An implementation of A-GAP with the above mentioned extensions is under way in our laboratory at KTH.

Acknowledgments: The authors would like to thank Mads Dam at KTH for fruitful discussions around the design and evaluation of A-GAP.

This paper describes work undertaken in the context of the Ambient Networks –IST project, which is partially funded by the Commission of the European Union. The views and conclusions contained herein are those of the authors and should not be interpreted as representing the Ambient Networks Project.

References

- [1] A. Gonzalez Prieto and R. Stadler, “Distributed Real-time Monitoring with Accuracy Objectives”, KTH Technical Report, December 2005
- [2] C. Olston, B. T. Loo and J. Widom, “Adaptive Precision Setting for Cached Approximate Values”, ACM SIGMOD 2001, Santa Barbara, USA, May 2001.
- [3] M. Dam, R. Stadler, “A Generic Protocol for Network State Aggregation”, Radiovetenskap och Kommunikation (RVK), Linköping, Sweden, 14-16 June, 2005.
- [4] K. Lim and R. Stadler. SIMPSON — a SIMple Pattern Simulator fOr Networks. <http://www.comet.columbia.edu/adm/software.htm>, 2005.
- [5] N. Spring, R. Mahajan, and D. Wetherall, “Measuring ISP topologies with Rocketfuel”, ACM/SIGCOMM, 2002, Pittsburgh, USA, August 2002.