KTH Electrical Engineering

Adaptive Real-time Monitoring for Large-scale Networked Systems

ALBERTO GONZALEZ PRIETO

Doctoral Thesis
Stockholm, Sweden, 2008

School of Electrical Engineering
KTH, Stockholm, Sweden

Akademisk avhandling som med tillstånd av Kungl Tekniska högskolan framlägges till offentlig granskning för avläggande av teknologie doktorsexamen den 21 november 2008 i KTH, Stockholm.

Tryck: Universitetsservice US AB

Abstract

Large-scale networked systems, such as the Internet and server clusters, are omnipresent today. They increasingly deliver services that are critical to both businesses and the society at large, and therefore their continuous and correct operation must be guaranteed. Achieving this requires the realization of adaptive management systems, which continuously reconfigure such large-scale dynamic systems, in order to maintain their state near a desired operating point, despite changes in the networking conditions.

The focus of this thesis is continuous real-time monitoring, which is essential for the realization of adaptive management systems in large-scale dynamic environments. Real-time monitoring provides the necessary input to the decision-making process of network management, enabling management systems to perform self-configuration and self-healing tasks.

We have developed, implemented, and evaluated a design for real-time continuous monitoring of global metrics with performance objectives, such as monitoring overhead and estimation accuracy. Global metrics describe the state of the system as a whole, in contrast to local metrics, such as device counters or local protocol states, which capture the state of a local entity. Global metrics are computed from local metrics using aggregation functions, such as SUM, AVERAGE and MAX.

Our approach is based on in-network aggregation, where global metrics are incrementally computed using spanning trees. Performance objectives are achieved through filtering updates to local metrics that are sent along that tree. A key part in the design is a model for the distributed monitoring process that relates performance metrics to parameters that tune the behavior of a monitoring protocol. The model allows us to describe the behavior of individual nodes in the spanning tree in their steady state. The model has been instrumental in designing a monitoring protocol that is controllable and achieves given performance objectives.

We have evaluated our protocol, called A-GAP, experimentally, through simulation and testbed implementation. It has proved to be effective in meeting performance objectives, efficient, adaptive to changes in the networking conditions, controllable along different performance dimensions, and scalable. We have implemented a prototype on a testbed of commercial routers. The testbed measurements are consistent with simulation studies we performed for different topologies and network sizes. This proves the feasibility of the design, and, more generally, the feasibility of effective and efficient real-time monitoring in large network environments.

Acknowledgments

First, I would like to thank my advisor Prof. Rolf Stadler, for his support, for the fruitful discussions, and for introducing me into the world of scientific research. I would also like to thanks all the people at LCN for the stimulating atmosphere, which made it possible for me to focus my efforts on research.

Additionally, I would like to express my gratitude to the organizations and entities that have founded my research: VINNOVA (the Swedish Governmental Agency for Innovation Systems), and the Graduate School in Telecommunications at KTH. A part of this research was performed in the context of "Ambient Networks", a European Commission 6th Framework Integrated Project.

I would like to thank my friends for their continuous help and support. To those in Stockholm for making me feel that I am at home here. To those in Barcelona for making me feel that I never left. I would like to express my special thanks to my family for their understanding, support and patience.

Curriculum Vitae

Alberto Gonzalez Prieto is a PhD candidate at the Royal Institute of Technology, Stockholm, Sweden. His research interests include management of large-scale networks, real-time network monitoring, and distributed algorithms.

He received his M.Sc. in Electrical Engineering from the Universidad Politecnica de Cataluña, Barcelona, Spain in 2002.

He was an intern at NEC Network Laboratories (Heidelberg, Germany) in 2001, working on his Master Thesis. He was also an intern at AT&T Research (Florham Park, NJ, USA) in 2007.

Table of Contents

1. Introduction 11

1.1 Background and Motivation 11
1.1.1 Large-scale Networked Systems 11
1.1.2 Management of Networked Systems 11
1.1.3 Autonomic Management Systems 12
1.1.4 Real-time Monitoring 12

1.2 The Problem 13

1.3 The approach 14
1.3.1 Monitoring through Tree-based Aggregation 14
1.3.2 Constructing a Stochastic Model for the Monitoring Process 15

1.4 Contribution of this Thesis 15
1.4.1 A Protocol for Distributed Monitoring of Large-scale Networked Systems 15
1.4.2 Protocol Evaluation and Testbed Implementation 16
1.4.3 Heuristics for Global Optimization 16
1.4.4 Controllability of Protocol Performance in Real-time 17
1.4.5 Dynamic Adaptation of Protocol Configuration 17

2. Related Research in Real-Time Monitoring with Performance Objectives 19

2.1 Types of Monitoring Queries 19

2.2 Aggregation Graphs 20
2.2.1 Trees 20
2.2.2 Multi-path Graphs 20
2.2.3 Effects of the Tree Topology on the Performance of Aggregation Protocols 21

2.3 The Trade-off between Quality of Estimation and Monitoring Resources 22
2.3.1 Latency vs. Total Traffic 22
2.3.2 Maximum Error vs. Total Traffic 23
2.3.3 Accuracy vs. Storage Requirements for Queries on Data Streams 25
2.3.4 Distributed Streams 26

2.4 Towards Self-configuring Monitoring Protocols 27

3. Summary of Original Work 29

Paper A: A-GAP: An Adaptive Protocol for Continuous Network Monitoring with Accuracy Objectives 29

Paper B: Monitoring Flow Aggregates with Controllable Accuracy *29*

Paper C: Real-time Network Monitoring Supporting Percentile Error Objectives *30*

Paper D: Controlling Performance Trade-offs in Adaptive Network Monitoring *30*

4. Open Research Questions for Future Research **35**

Bibliography **39**

1. Introduction

1.1 Background and Motivation

1.1.1 Large-scale Networked Systems

Large-scale networked systems are omnipresent today, and they deliver services that are critical to both businesses and the society at large. They have experienced an extraordinary growth over the last decades. The Internet, for instance, was a very small network thirty years ago and virtually unknown to the general public. Today it comprises several thousands of Autonomous Systems [82] and close to a billion of daily users [67].

Large-scale networked systems are not confined to the traditional Internet. Server clusters for example, which can offer cluster-based web services or utility computing, form another class of networked systems that is steadily increasing in importance. It is said that Google alone maintains a pool of some one million servers [76]. Further classes of emerging networked systems, for which large deployments are envisioned, include wireless sensor networks and industrial process control systems.

1.1.2 Management of Networked Systems

As networked systems provide key functions to businesses and society, their continuous and correct operation are of the utmost importance. A well-engineered management system helps in achieving the operational objectives of a networked system. Generally, Network Management "refers to the activities, methods, procedures, and tools that pertain to the operation, administration, maintenance and provisioning of networked systems" [63]. One can consider a network management system as executing a closed-loop control cycle, whereby the (distributed) system state is estimated on a continuous basis, and, based on this estimation, a process dynamically determines a set of actions that are executed on the networked system in order to achieve operational objectives.

The contribution of this thesis is in the monitoring part of the control cycle. In the context of the OSI management functional area, monitoring falls into the category of performance management [64]. Furthermore, the focus of this research is on monitoring within a single administrative domain of a networked system, for instance, within an Autonomous System in the context of the Internet.

1.1.3 Autonomic Management Systems

As networked systems have increased in size and complexity, the necessity for simplifying their management and reducing their operational costs has become evident. In response to this need, the vision of *autonomic management*, or more general, autonomic computing has emerged. Following the autonomic computing vision [70], a human manager specifies her goals or high-level objectives for the networked system, which are expressed in the form of behavioral rules, utility functions, and so forth. That is, the manager does not specify what has to be done, but what has to be achieved. These goals are the input to the management system, which takes the necessary actions to achieve them. This means, for instance, that the system continuously reconfigures to maintain its state near a desired operating point, despite disturbances caused by load changes, failures, etc. In order to adapt to such events in a timely manner, *real-time monitoring on a continuous basis is required* [71][72][73].

There is a consensus today in the research community that, in order to achieve the vision of autonomic management for large-scale networked systems, it is necessary to move from traditional centralized approaches to decentralized approaches. The main drivers for such a transition are the need for increased scalability in system size, faster adaptability to changing conditions and increased robustness to different types of failures. In traditional approaches to network management, tasks are performed by dedicated management servers that reside outside the network. The current trend, advocated and pursued by some key vendors, is to *push management intelligence into the networked system*. As a consequence, many management tasks will be performed inside the network by the managed system itself. One such task that we envision will be performed inside the network in the future is continuous real-time monitoring. For this reason, our approach to real-time monitoring is decentralizing the monitoring task and performing it inside the network.

1.1.4 Real-time Monitoring

The focus of this thesis is decentralized real-time monitoring, a key building block in the realization of decentralized autonomic management. Real-time monitoring provides the necessary input to the decision-making process of network management, enabling management systems to perform self-configuration and self-healing tasks.

This research addresses monitoring of *global metrics* in large-scale networked systems. Global metrics describe the state of the system as a whole, in contrast to local metrics, such as device counters or local protocol states, which capture the state of a local entity. Global metrics are computed from local variables using *aggregation functions*, such as SUM, AVERAGE and MAX. Examples of global metrics in the context of the Internet are the total number of VoIP flows in a domain and the list of the 50 subscribers with the longest end-to-end delay. In the context of server clusters, the workload distribution among the servers is an example of a global metric.

An important aspect of real-time monitoring is the overhead it introduces. Clearly, the more detailed and accurate state information a monitoring system provides, the higher the cost of the monitoring task in terms of overhead. At the same time, the more detailed and accurate the monitoring data, the better the system's ability to

achieve its operational goals. Therefore, it is important to engineer monitoring systems that allow controlling such trade-offs.

In the context of this thesis, we address the problem of efficient monitoring for given performance objectives. We argue that, with the autonomic computing vision in mind, *the monitoring system itself must be autonomic*. As a consequence, given a set of objectives, such as the quality of the estimation, the monitoring system must configure itself in a way that these objectives can be met, and it must dynamically adapt to changing conditions.

1.2 The Problem

In the context of large-scale networked systems, our goal is to engineer an efficient monitoring protocol that provides a management station with a continuous estimate of a global metric for given performance objectives.

We call such a global metric an *aggregate*. It denotes the result of computing a multivariate function, whose variables are local metrics from nodes across the networked system.

In the light of the above discussion, the protocol should meet the following design goals:

- *Effectiveness.* The protocol should achieve a given accuracy objective.
- *Efficiency.* The overhead introduced by the protocol should be minimal.
- *Adaptability*. The protocol should be capable of adjusting its configuration to different networking conditions in order to operate effectively and efficiently. For instance, it should quickly adapt to changes in the network topology or to node failures. It should further adjust to changes in the evolution of the monitored metrics.
- *Scalability.* Performance metrics of the protocol, such as the overhead, should grow moderately with the system size.
- *Controllability.* The performance of the protocol should be controllable. For instance, the protocol should support the setting of objectives for different performance metrics, such as estimation accuracy, overhead, and adaptation time.

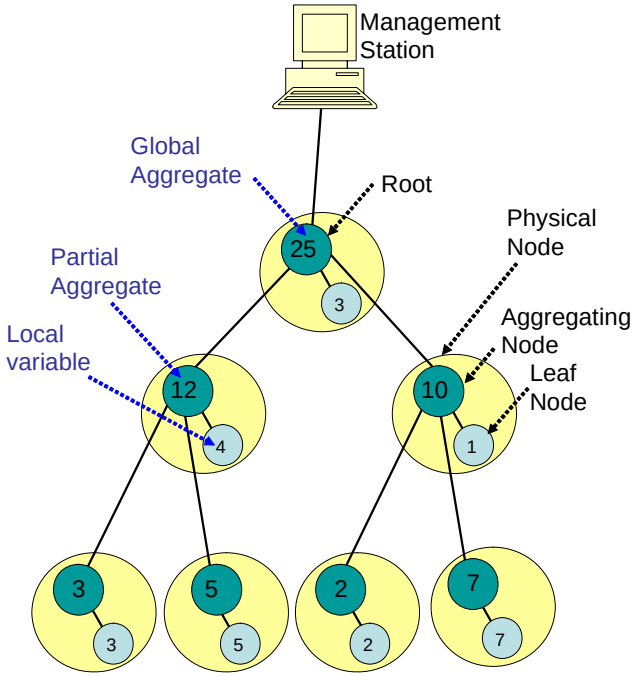

Fig. 1. Example of an aggregation tree with aggregation function SUM.

Despite the key role of continuous real-time monitoring in realizing adaptive networked systems, currently, there are no solutions for continuous monitoring of global metrics that meet the goals above. Specifically, achieving a satisfactory degree of controllability has proved to be a hard goal. For instance, related works in continuous monitoring support accuracy objectives for only the maximum estimation error.

This research is based on the following assumptions. First, it assumes a distributed management architecture, whereby each node in the networked system participates in the monitoring task by running a monitoring process, either internally or on an external, associated device. Second, it assumes that local metrics can be accessed on each node, where they are periodically updated in an asynchronous fashion.

1.3 The approach

1.3.1 Monitoring through Tree-based Aggregation

In the literature, two approaches are described to compute aggregates in a distributed fashion. The first is based on using a spanning tree, where the aggregate is incrementally computed from the leaves towards the root [19], [27], [22], [18], [41],

[37], and [35]. The result of the computation is available at the root node. More recently, results have been reported on computing aggregates using gossiping protocols [77], [78], [79], [80], and [81]. In this case, the result of the computation is available on all nodes, and it converges probabilistically to the true value (when the local variables remain constant).

In this research, we follow the tree-based approach, as quantitatively controlling the accuracy objectives is tractable in tree-based approaches, while it remains unknown (to the best of our knowledge) whether it is feasible for a gossip-based approach.

We use a mechanism we call *in-network aggregation*, whereby aggregates are computed inside the network in a distributed way. Figure 1 shows an example of such an aggregation process. It uses an *aggregation tree*, whereby each node holds information about its children in the tree, in order to compute the partial aggregate, i.e., the aggregate value of the local management variables from all nodes of the subtree where this node is the root. The computation is *push-based* in the sense that updates of monitored variables are sent towards the management station along the aggregation tree.

In order to achieve efficiency, we combine the concepts of in-network aggregation and that of *filtering*. Filtering drops updates that are not significant when computing an aggregate for a given accuracy objective.

1.3.2 Constructing a Stochastic Model for the Monitoring Process

To achieve the protocol design goals of efficiency and controllability, we need *a model for the distributed monitoring process* that relates performance metrics to control parameters. We develop a model based on discrete-time Markov chains, which allows us to describe the behavior of individual nodes in the aggregation tree in their steady state.

To be applicable in practice, we believe it to be important for the model to be general enough to capture a range of performance objectives, for instance, protocol overhead on nodes or different error objectives including percentile-based and average error objectives. In fact, all related works in monitoring consider accuracy objectives for only the maximum estimation error, and the models used in those works do not support the above range of performance objectives (see section 2.3).

1.4 Contribution of this Thesis

1.4.1 A Protocol for Distributed Monitoring of Large-scale Networked Systems

This thesis presents A-GAP, a novel distributed protocol for real-time continuous monitoring of global metrics (i.e., aggregates), which aims at achieving a given monitoring accuracy with minimal overhead.

The protocol has proved to be effective, efficient, adaptive, and controllable. It is effective since it closely meets given performance objectives. It is efficient in the sense that, when allowed a modest error in estimating an aggregate, it reduces the overhead significantly compared to the case where no error is allowed, which means that all changes in local variables are reported to the management station. The protocol effectively adapts its configuration to changes in the networking conditions, in order to continuously meet the performance objectives (see section 1.4.5). It is controllable, for its performance can be controlled along different dimensions, as discussed in section 1.4.4.

Regarding scalability, our protocol achieves a logarithmic increase of the total protocol overhead with the number of aggregating nodes. In addition, for a given accuracy objective, it achieves a linear increase of the maximum protocol overhead on a node with the network size. (At this point, it is unclear for us whether a sub-linear behavior can be achieved, and this aspect merits further investigation.)

1.4.2 Protocol Evaluation and Testbed Implementation

We have evaluated our protocol experimentally, through simulation and testbed implementation.

For the simulation-based evaluation, we have used both real and synthetic traces, and a wide range of network topologies. We have evaluated the protocol for the following criteria. First, the capability to *effectively control performance trade-offs*, including the capability to meet performance objectives. Second, its capability for *accurately predicting its performance*, including the error distribution, and the incurred overhead. Third, the capability to *adapt to changes in the networking conditions*, such as a sudden change in topology. Fourth, its versatility in *supporting different accuracy objectives*, such as, the average error, the maximum error, and percentile errors.

The implementation of a prototype on a testbed of commercial routers has permitted us to evaluate a fifth criterion, the *feasibility of real-time flow monitoring* with controllable accuracy in today's IP networks (i.e., without requiring changes to the routers).

1.4.3 Heuristics for Global Optimization

Our protocol attempts to solve the problem of minimizing the monitoring overhead for a given estimation error using two heuristics. The first attempts to minimize the maximum processing load on all nodes by minimizing the load within each node's neighborhood. In other words, the first heuristic maps the global optimization problem onto a set of local optimization problems, which are solved independently and asynchronously. This way, the computational complexity of solving the global problem is reduced.

The second heuristic is used to solve the local optimization problem. The solution is obtained through a grid search, in which the search space is limited to small changes of the filters' widths. This limitation further reduces the computational complexity.

Our results show that *the combination of these two heuristics permits us to meet the design goals*.

1.4.4 Controllability of Protocol Performance in Real-time

Our evaluation results show that *the performance of a distributed monitoring protocol can be controlled* along different performance dimensions, including protocol overhead, accuracy of metric estimation, and adaptation time to changes in the networking conditions. Furthermore, we have shown that it is feasible to control the trade-offs among these performance metrics.

The trade-off between estimation accuracy and protocol overhead in steady state can be controlled effectively and efficiently by filters in the nodes of the aggregation tree. Allowing a modest error in estimating an aggregate permits reducing the overhead significantly (compared to an approach where all changes in local variables are reported).

The trade-off between protocol overhead and adaptation time can be controlled through the aggregation tree topology. Generally speaking, allowing a larger protocol overhead permits reducing the adaptation time by choosing the appropriate aggregation tree topology. Our results suggest that the protocol overhead is strongly influenced by the number of aggregating (internal) nodes, and that the adaptation time primarily depends on the height of the aggregation tree.

Our results show that it is *feasible to support different types of accuracy objectives*, such as, the average error, percentile errors and the maximum error. This versatility permits to fulfill different requirements from different monitoring data consumers. For instance, while the average error is a significant control parameter for many practical scenarios, in the context of probabilistic quality of service assurance, percentile-based accuracy is often relevant. The maximum error is relevant for deterministic quality of service assurance.

To support several forms of controllability, we have shown that it is *feasible to provide accurate real-time estimations for the performance of a distributed monitoring protocol*. Specifically, it is feasible to provide a management station with the distribution of the estimation error for the aggregate, and the expected overhead for each node on the aggregation tree.

1.4.5 Dynamic Adaptation of Protocol Configuration

We have engineered a distributed monitoring protocol that *dynamically adapts to changes* in the network conditions. These changes include modification to the network topology, node and link failures, and changes to the statistical behavior of the monitored metrics. The protocol adapts by re-configuring the filters and the topology of the aggregation tree, in order to meet the performance objectives under the new conditions.

Our protocol adapts very quickly. For instance, for all simulation scenarios we tested, in case of a node failure, the adaptation time is very short: the settling time for

the accuracy is a fraction of a second, and it takes a few seconds for the overhead to settle.

2. Related Research in Real-Time Monitoring with Performance Objectives

In this section, we relate our work to recent and current research with similar objectives in the fields of wireless sensor networks, queries on data streams, and network monitoring.

First, we classify monitoring queries in order to position our work with respect to that in the literature. Section 2.2 discusses different types of aggregation graphs for computing global metrics. In section 2.3, we present recent results on a fundamental trade-off in monitoring: quality of the estimation versus the required monitoring resources. Section 2.4 discusses self-configuration in the context of network monitoring.

2.1 Types of Monitoring Queries

We classify monitoring queries on aggregates into three categories:

- *1-time queries* refer to a snapshot of the system state. They request an estimation of the aggregate at a specific time.
- *N-time queries* request an estimation of the aggregate at N discrete times. N-time queries are usually periodic.
- *Continuous queries* request a continuous estimation of the aggregate.

Most of the existing works on queries on aggregates focus on N-time queries. They are commonly realized as periodic and independent 1-time queries [27][35][19][21][22][37]. In contrast, our work addresses continuous queries like the research reported in [31], [32], and [33].

Note that this query classification is not universally accepted in the literature. When some authors talk about *continuous queries*, they are actually referring to what we here call N-time queries (e.g., [27], and [11]).

2.2 Aggregation Graphs

A key concept in distributed computation of aggregates is the aggregation graph. It determines how partial aggregates are exchanged between the nodes and how they are aggregated in the nodes of this graph.

2.2.1 Trees

The most common aggregation graph is the tree, where children send their partial aggregates to their parents and the aggregate is available at the root node [19][27] [22][18][41][37][35]. Several algorithms have been proposed for creating and maintaining aggregation trees (e.g., [34], [12], and [10]). A key property of trees is that they guarantee that any two vertices are connected by exactly one path. This enables straightforward algorithms for aggregate computation that guarantee that each local variable is only accounted for once. We discuss the importance of this property below.

The drawback of trees as aggregation graphs is that the aggregation protocol is sensitive to packet losses [30]. For instance, consider a 1-time SUM query. If a partial aggregate sent by a node with a large number of children is lost, this can result in a large estimation error at the root node. Therefore, trees are not well suited for lossy environments.

2.2.2 Multi-path Graphs

To overcome the sensibility of trees to packet losses, [30] and [7] propose the use of multipath aggregation graphs instead. These approaches aim at reducing the effects of lossy links by providing path diversity: local variables flow towards the management station(s) following multiple paths.

In contrast to trees, multi-path graphs potentially introduce double accounting: a local variable is considered more than once in the aggregate computation. The effects of double accounting depend on the considered aggregate function. [27] classifies aggregates into two categories. First, duplicate-insensitive aggregates. The computation of these aggregates is not affected by double accounting. In other words, their computation is an idempotent operation. Some examples are MAX and MIN. Second, duplicate sensitive aggregates, like SUM. For this type of aggregates, duplicate accounting distorts the estimation.

The computation of duplicate-sensitive aggregates using multi-path graphs brings the challenge of avoiding double accounting while allowing incremental aggregation in the aggregation graph. To achieve this, [30] and [7] make use of a probabilistic counting algorithm by Flajolet and Martin [15]. This algorithm provides an estimation of the count aggregate for 1-time queries. The work in [15] has been extended to estimate sums in [7]. The key idea of the algorithm in [15] is that it keeps, for each

node, a synopsis, i.e., a sketchy summary, of the local variables that probabilistically avoids double accounting. While double accounting is not fully avoided, the algorithm provides probabilistic guarantees on the bias and standard deviation of the aggregate estimation. The quality of the estimation is controlled by the synopsis size. We will discuss this further in the next section. Note that since this algorithm assumes hash functions that appear random, it is vulnerable to adversarial local variables combinations [7].

Our protocol, A-GAP uses an aggregation tree instead of a multi-path approach for two main reasons. First, trees require nodes in the overlay to process fewer messages than multi-path graphs. A-GAP aims at minimizing the number of messages processed by a node; therefore, trees are preferred. Second, we consider scenarios with limited losses in our evaluation studies. For such scenarios, the algorithms in [30] and [7] perform worse than tree-based approaches in terms of estimation accuracy. We consider scenarios in the context of ISP networks, where packet losses are not common and when they occur, level 2 or level 4 mechanisms can mitigate their impact by retransmitting lost packets. In contrast, in wireless sensor networks, (those targeted by [30] and [7]) lossy links are common and such mechanisms are unaffordable. Therefore, while multi-path approaches are attractive for wireless sensor networks, they are not for scenarios with a reliable network layer.

2.2.3 Effects of the Tree Topology on the Performance of Aggregation Protocols

The topology of the aggregation tree has a significant impact on the performance of an aggregation protocol. Tree characteristics like height and node degree influence the number of messages processed by each node and the latency for computing an aggregate.

Despite the relevance of the tree topology, few works have investigated this issue. One of them is [21], which studies the communication costs of 1-time queries as a function of the relative location of data sources and the management station. [22] analyzes how placing the management station in different locations impacts the protocol performance. [35] investigates how to create efficient trees for certain types of queries. It proposes a tree creation algorithm for 1-time queries with (sql-like) "group by" clauses.

Our protocol A-GAP uses GAP [10], as an underlying protocol, to create and maintain an aggregation tree. Given an overlay, GAP creates a tree that minimizes the distance of each node to the root. Therefore, the choice of the overlay influences the performance of A-GAP. Commonly, for the sake of simplicity, the overlay we choose has the same topology as the underlying physical network.

One of the papers included in this thesis, paper D, discusses how the choice of the topology of the aggregation tree controls the trade-off between protocol overhead and adaptation time. Our results suggest that the protocol overhead is strongly influenced by the number of aggregating (i.e., internal) nodes, and that the adaptation time primarily depends on the height of the aggregation tree.

2.3 The Trade-off between Quality of Estimation and Monitoring Resources

Monitoring involves a fundamental trade-off: quality of the estimation versus required resources for the monitoring task. Achieving high-quality estimations requires significant amounts of resources, and limited resources generally mean low quality estimations. The quality of an estimation has two aspects. First, the accuracy: how close the estimation is to the actual value. Second, the latency: how long it takes to make available an estimation of the current aggregate to the management station(s).

In the literature, we find different instantiations of this trade-off investigated. Here, we classify them into three categories: latency vs. total monitoring traffic (we also refer to total monitoring traffic as overall monitoring traffic), maximum error vs. total monitoring traffic, and accuracy vs. storage requirements.

2.3.1 Latency vs. Total Traffic

Several papers on monitoring wireless sensor networks (WSN) (e.g., [19], [22], [21], and [27]) discuss the trade-off between latency for computing the aggregate and either total traffic of the monitoring protocol or total dissipated energy, for the case of N-time queries.

In WSN, the total traffic and the total dissipated energy are closely related, since radio communications dominate energy consumption in wireless sensors [27]. There is an almost linear dependency between total traffic and total dissipated energy. Some authors consider the total energy as the parameter to minimize, while others consider the total number of transmitted bits or messages. We group their works under the same category, namely latency vs. total traffic.

The importance of these two metrics in WSN scenarios comes from the fact that the lifetime of a WSN, a key aspect of these networks, depends on them. A WSN lifetime is limited by the batteries of its sensors. Note that battery replacement is not an option in most realistic scenarios. Therefore, the maximization of the network lifetime is achieved by minimizing the energy consumption, i.e., the number of messages transmitted per query.

In WSN, it is common to use a locally synchronous algorithm, whereby each node reports its partial aggregate to its parent in the aggregation tree once it has received the reports from all its children with their partial aggregates. Each query creates a wave of updates towards the management station. For these algorithms, each node only sends a single message to its parent for a 1-time query [27][21][22] (n messages for a N-times query). The drawback of such locally synchronous algorithms is that they introduce latencies. Since a node can only send its partial aggregate once it has received the partial aggregates from all of its children, data may be held for some time at some intermediate node in the aggregation tree. The incurred latency is proportional to the tree height [21][22].

Our protocol A-GAP, in contrast, does not delay the sending of partial aggregates from children to parents. Updates may be filtered out though, based on their value. As a consequence, significant changes in the monitored variables are delivered promptly

all the way to the management station. In contrast, locally synchronous algorithms do not immediately forward an update even if it contains a significant change of the partial aggregate. Another difference between our work and the above class of algorithms is that A-GAP has different performance objectives. It aims at minimizing the maximum load across all nodes rather than the total traffic. The reason for this comes from the fact that we consider different scenarios. As discussed above, in WSN, the metric of interest is the total traffic (or total dissipated energy) because it is associated with the network lifetime. In contrast, for other networked scenarios, such as fixed networks or WLANs, the challenge is to avoid nodes from becoming overloaded, especially in large-scale and dynamic networks.

The trade-off between latency for computing the aggregate and total traffic of the monitoring protocol has also been discussed in the context of fixed network. [80] proposes an algorithm based on aggregation trees and gossiping. The algorithm provides all nodes in the network system with an estimate of the global metric. The algorithm ensures eventual consistency, which guarantees that if updates to local variables cease, the estimate of the global metric will eventually be the same in all nodes.

In the algorithm in [80] the control parameter is the gossip rate, i.e., how often nodes exchange state information. The higher this rate, the slower updates to the local variables propagate across nodes.

Note, that none of the related works discussed in this section aims at controlling the accuracy of the estimation.

2.3.2 Maximum Error vs. Total Traffic

Another instantiation of the quality of the estimation vs. resources trade-off studied in the context of WSN is the maximum error versus the total traffic (e.g., [18], [35], [11], [3], and [8]). In these works, the focus is on providing guarantees on the accuracy of the estimation, ensuring that the difference between the estimation and the actual aggregate is always within a configurable range.

In [3] the authors address continuous queries. They give an algorithm in which all sensors hold an estimate of the global aggregate. Changes of this estimate that are larger than a configurable maximum are propagated using epidemic protocols. In the evaluation, the authors show that there is a large difference (one order of magnitude) between the actual error and the maximum error objective (the control parameter of the algorithm). This solution considers only MAX queries and it is unclear how it could be efficiently extended to other aggregation functions.

In [35] the authors focus on periodic N-times queries on MAX, MIN, SUM and AVG aggregates. The proposed algorithm takes into consideration the dynamics of the local variables. In each period, nodes report their local variables if and only if the current value differs from the last report in more than a configurable percentage.

As in [3], the authors of [35] find that there is a difference of around one order of magnitude between the average error and the maximum error. The evaluation results are presented in terms of the relative average error (over time) rather than the maximum error (the control parameter of their algorithm). A drawback of the proposal in

[35] is that it does not consider the impact of the local changes on the aggregate, but only its relative change. For instance, consider a SUM aggregate with a maximum error of 20%. A local variable (v1) change from 1 to 2 would be reported, while a change from another variable (v2) from 100 to 118 would not be reported. In such a case, for the same use of resources (1 message), the error would be smaller if the reported change was that of v2. Therefore the algorithm in [35] can not be optimal, since the objective is to minimize the total traffic for a given maximum error.

The work in [11] addresses N-time queries, but has similarities with A-GAP, namely the aggregate is computed incrementally along a tree, and filters are used in both cases. It differs from our work in two aspects. First, it considers accuracy objectives for only the maximum estimation error. Second, the algorithm in [11] estimates statistics from all the partial aggregates, which depend on the filters. Therefore, after modifying a filter, new estimations need to be made. In contrast, A-GAP only estimates the local variables, which are not affected by filters. All other variables are continuously computed by A-GAP based on these estimates. This gives A-GAP more flexibility in selecting the duration of the control cycle. This comes at the cost of A-GAP assuming independence among the local variables. In some scenarios, independence is a common assumption, e.g.: the number of voice over IP flows entering the domain. It is an open issue how the performance of A-GAP would be affected in case local variables independence would not hold.

The work in [55] is similar to that in [11]. In contrast to [11] however, filter reconfiguration only takes place, if the expected performance gain is larger than the cost for reconfiguration.

The work in [8] investigates continuous queries on frequency distributions. Examples of such distributions are the number of flows per application or the number of flows per destination in a network. The algorithm presented in [8] aims at guaranteeing that for any element in the domain (e.g., the domain of applications), the estimation error is always within a configurable range.

From frequency distributions different types of queries can be answered [83]. Take the example of the frequency of flows per application. It permits answering queries on the number of http flows, the total number of flows in the network or the list of most popular applications. This flexibility comes at the cost of a higher overhead compared to that of an aggregation protocol that uses scalar-valued aggregation functions, such as SUM, etc. For instance, if the management application is only interested in the number of VoIP and http flows, protocols like [11], which use scalar-valued aggregation, would likely provide such estimates with a lower overhead.

As in [35], the solution in [8] only considers the relative changes of local variables, not their impact on the aggregate. Therefore, it suffers from the same drawbacks.

While all related work on aggregation protocols considers a single error objective, namely, the maximum error, our protocol A-GAP supports a range of controllable error objectives, including average error, percentile error and maximum error. Supporting the maximum error objective is relatively straightforward, compared to the other objectives mentioned above. The works in [18] and [35] show that the maximum error is a loose upper bound on the accuracy achieved. This is in line with the results from our simulation scenarios, where the error distribution at the root node of the aggregation tree resembles a normal distribution and has long tails. This observation shows

that the maximum error is of limited practical relevance and suggests that other error objectives, such as the average error, should be used instead.

The performance curves that relate overhead to quality of estimation achieved for the protocols described in [18], [35], and [11] are similar to a negative exponential function for the errors considered. Our evaluation gives the same behavior for our A-GAP protocol.

2.3.3 Accuracy vs. Storage Requirements for Queries on Data Streams

Another instantiation of the trade-off between quality of estimation and monitoring resources is accuracy vs. storage requirements. This trade-off is well-known in the context of data streaming. A data stream is a sequence of data items generated over time by a source. Examples include stock tickers, sequences of clicks in web navigation sessions, periodic readings of sensors, etc.

Queries on data streams commonly refer to the entire stream of data over its lifetime. Providing exact answers requires storing the whole data stream in the general case, which is infeasible in many scenarios. The challenge is accurately answering queries on data streams while minimizing the amount of memory required.

Several researches (see [2] for a survey) have investigated how storage (i.e., memory) requirements can be traded for the accuracy of the answer. The literature offers a variety of algorithms for controlling this trade-off by creating summaries of the data stream, and the accuracy of the information that can be extracted is a function of the summary size. Examples of such algorithms are:

- *Random sampling.* This technique consists in storing only a subset of the data stream. Such algorithms implicitly assume that a small subset of the stream contains enough information to answer the query. These algorithms can be combined with other summarization techniques.

- *Histogram.* Several algorithms have been proposed for computing histograms with memory and accuracy bounds (for a given data set size). [2] classifies these algorithms into three types: (i) piece-wise constant functions, (ii) set of buckets with increasing quantiles by a constant value, and (iii) those aimed at answering iceberg queries [14]. Iceberg queries identify data values whose frequency is above a given value. An example of an iceberg query is: which applications generate more than 5% of the total traffic?

- *Wavelets.* These are mathematical functions that project a data stream onto a set of vectors. In order to reconstruct the stream perfectly from the set of vectors, all wavelets coefficients are needed. The stream can be approximated however using only the most significant coefficients. It has been shown that wavelets can provide higher accuracy than histograms for the same memory usage [2].

- *Query-specific sketching*. Most of the above algorithms are generic in the sense that they can be used for answering a variety of queries. The literature also provides a number of sketches tailored to specific queries. Within this group, sketches for queries on frequency moments have attracted a lot of interest. The information provided by the frequency moments includes the number of distinct elements in the data stream, the Gini coefficient, and the multiplicity of the most frequent item. The Flajolet and Martin algorithm [15] discussed in section 2.2 falls into this category.

All the algorithms above were originally designed for N-time queries over a single data stream. The extension of these algorithms to distributed systems could reduce the size of the messages exchanged by including in them a summary of the stream rather than the whole stream. For instance, the Flajolet and Martin algorithm was applied in wireless sensor networks for N-time COUNT [30] and N-time SUM [7] queries. In these works, the stream was composed of the union of the current values of the local variables.

2.3.4 Distributed Streams

As mentioned above, most of the research in queries on data streams considers only a single data stream. The multi-stream case has attracted limited attention so far. The most relevant work in this area is that of Gibbons and Tirthapura on querying *distributed streams* [16, 17]. In this case, there are a number of parties, each of them observing a single different data stream. At a differentiated node, an aggregate function on those data streams is computed.

The work in [16, 17] addresses a very similar problem to the one investigated in this thesis in that it considers a set of local variables that change over time (called data streams in this context) in a networked environment, and it estimates a global metric observing performance objectives. However, there are a number of differences in the problem statement. First, this work addresses N-time queries on the entire data stream up to the time of the query [16][17], which does not match with the semantics of real-time monitoring, as historical data is usually considered in the queries. Second, [16] and [17] compute the global metric using a centralized approach, which leads to a solution that is not scalable, while our approach is distributed, using an aggregation tree.

A second piece of work in querying distributed data streams is [54]. In contrast to the above discussed works [16][17], this work considers 1-time queries on the current state of the network. That is, it does not consider queries on the entire data streams, but only on the current values of the sources. The authors of [54] identify the optimal location in the network for the query operators (i.e., the aggregating nodes in the aggregation tree) for minimal protocol overhead, but they do not take estimation error objectives into account.

2.4 Towards Self-configuring Monitoring Protocols

Over the last years, efforts have been made towards engineering self-configuring management systems, with the goal of simplifying the tasks of human managers. Self-configuration is of particular importance for distributed management systems, as they are more complex than centralized ones [59]. Their configuration has several more dimensions, and it is more difficult to foresee the effect of configuration changes.

The problem of optimal configuration in the context of network monitoring has been studied in [59], [60], [61], [56], and [57]. These works contain performance models for monitoring systems and comparative studies of centralized vs. decentralized approaches. They show that, in general and for key metrics, distributed approaches can achieve better performance than centralized ones.

Efficiency for hierarchical monitoring approaches is achieved using two key mechanisms, namely in-network filtering and aggregation. Both mechanisms trade overhead for loss of information. Both mechanisms control the ratio between incoming and outgoing monitoring traffic for a node. This metric is called selectivity in [59], and it is a common input parameter to the performance models used for the monitoring approaches. However, the models in [59] do not relate the filter configuration to its selectivity. As a consequence, they do not permit to predict the system performance based on the choice of the filters. Note that in the work presented in this thesis, the stochastic model for the monitoring process relates the selectivity of each aggregating node to the filters, therefore, predicting the performance metrics based on them is possible.

The work reported in [58] addresses self-configuration of tree topologies for the purpose of event monitoring with minimal overhead. The algorithm proposed in [58] creates the aggregation tree, bottom up, in an iterative fashion. In each step, a new level of the tree is created until the root is determined. This work is related to a problem addressed in this thesis, namely, identifying a suitable topology for the A-GAP aggregation tree. However, it is not clear how the approach in [58] can be applied to determine a topology that meets the performance objectives of adaptation time and estimation accuracy.

Note that, similarly to the work in [59], the model of the monitoring process in [58] does not permit predicting performance metrics in a manner presented in this thesis.

Some conclusions reached by the above discussed research ([59], [60], [61], [56], and [57]) are consistent with the results from this thesis work. First, the overhead of hierarchical solutions tends to decrease with the number of internal nodes in the tree. Second, the marginal improvement in performance obtained by adding an internal node, decreases with the number of internal nodes.

3. Summary of Original Work

Paper A: A-GAP: An Adaptive Protocol for Continuous Network Monitoring with Accuracy Objectives

We present A-GAP, a novel protocol for continuous monitoring of network state variables, which aims at achieving a given monitoring accuracy with minimal overhead. Network state variables are computed from device counters using aggregation functions, such as SUM, AVERAGE and MAX. The accuracy objective is expressed as the average estimation error. A-GAP is decentralized and asynchronous to achieve robustness and scalability. It executes on an overlay that interconnects management processes on the devices. On this overlay, the protocol maintains a spanning tree and updates the network state variables through incremental aggregation. Based on a stochastic model, it dynamically configures local filters that control whether an update is sent towards the root of the tree. We evaluate A-GAP through simulation using real traces and two different types of topologies of up to 650 nodes. The results show that we can effectively control the trade-off between accuracy and protocol overhead, and that the overhead can be reduced by almost two orders of magnitude when allowing for small errors. The protocol quickly adapts to a node failure and exhibits short spikes in the estimation error. Lastly, it can provide an accurate estimate of the error distribution in real-time.

This paper has been published in IEEE Transactions on Network and Service Management (TNSM), Vol. 4, No. 1, June 2007.

Paper B: Monitoring Flow Aggregates with Controllable Accuracy

We show the feasibility of real-time flow monitoring with controllable accuracy in today's IP networks. Our approach is based on Netflow and A-GAP. A-GAP is a protocol for continuous monitoring of network state variables, which are computed from device metrics using aggregation functions, such as SUM, AVERAGE and MAX. A-GAP is designed to achieve a given monitoring accuracy with minimal overhead. A-GAP is decentralized and asynchronous to achieve robustness and scalability. The protocol incrementally computes aggregation functions inside the network and, based on a stochastic model, it dynamically configures local filters that control the overhead and accuracy. We evaluate a prototype in a testbed of 16 commercial routers and provide measurements from a scenario where the protocol continuously estimates the total number of FTP flows in the network. Local flow metrics are read out from Netflow buffers and aggregated in real-time. We evaluate the prototype for the following criteria. First, the ability to effectively control the trade-off between monitoring accu-

racy and processing overhead; second, the ability to accurately predict the distribution of the estimation error; third, the impact of a sudden change in topology on the performance of the protocol. The testbed measurements are consistent with simulation studies we performed for different topologies and network sizes, which proves the feasibility of the protocol design, and, more generally, the feasibility of effective and efficient real-time flow monitoring in large network environments.

This paper has been published in the 10th IFIP/IEEE International Conference on Management of Multimedia and Mobile Networks and Services (MMNS 2007), San José, California, USA, October 2007.

Paper C: Real-time Network Monitoring Supporting Percentile Error Objectives

We report on the versatility of A-GAP for supporting different types of accuracy objectives. Previously, we considered accuracy objectives expressed in terms of the average error for this protocol. In this paper, we focus on percentile error objectives. A-GAP is a protocol for continuous monitoring of network state variables. Network state variables are computed from device counters using aggregation functions, such as SUM, AVERAGE and MAX. A-GAP is designed to achieve a given monitoring accuracy with minimal overhead. A-GAP is decentralized and asynchronous to achieve robustness and scalability. It executes on an overlay that interconnects management processes on the devices. On this overlay, the protocol maintains a spanning tree and updates the network state variables through incremental aggregation. Based on a stochastic model, it dynamically configures local filters that control whether an update is sent towards the root of the tree. We evaluate A-GAP through simulation using real traces and an ISP topology (Abovenet). The results prove the versatility of A-GAP for supporting different types of accuracy objectives. The results also show that we can effectively control the trade-off between accuracy and protocol overhead, and that the overhead can be reduced significantly by allowing small errors.

This paper has been published in the 14th HP Software University Association (HP-SUA) Workshop, 8-11 July 2007, Munich, Germany.

Paper D: Controlling Performance Trade-offs in Adaptive Network Monitoring

A key requirement for autonomic (i.e., self-*) management systems is a short adaptation time to changes in the networking conditions. In this paper, we show that the adaptation time of a distributed monitoring protocol can be controlled. We show this for A-GAP, a protocol for continuous monitoring of global metrics with controllable accuracy. We demonstrate through simulations that, for the case of A-GAP, the choice of the topology of the aggregation tree controls the trade-off between adaptation time and protocol overhead in steady-state. Generally, allowing a larger adaptation time permits reducing the protocol overhead. Our results suggest that the adaptation time primarily depends on the height of the aggregation tree and that the protocol

overhead is strongly influenced by the number of internal nodes. We outline how A-GAP can be extended to dynamically self-configure and to continuously adapt its configuration to changing conditions, in order to meet a set of performance objectives, including adaptation time, protocol overhead, and estimation accuracy.

This paper has been accepted for publication in the 11th IFIP/IEEE International Symposium on Integrated Network Management (IM 2009), New York, New York, USA, June 1-5, 2009.

List of Publications in the Context of this Thesis

1. A. Gonzalez Prieto and R.Stadler "A-GAP: An Adaptive Protocol for Continuous Network Monitoring with Accuracy Objectives", IEEE Transactions on Network and Service Management (TNSM), Vol. 4, No. 1, June 2007
2. A. Gonzalez Prieto and R. Stadler "Controlling Performance Trade-offs in Adaptive Network Monitoring", accepted for publication in the 11th IFIP/IEEE International Symposium on Integrated Network Management (IM 2009), New York, New York, USA, June 1-5, 2009.
3. A. Gonzalez Prieto and R. Stadler, "Monitoring Flow Aggregates with Controllable Accuracy", 10th IFIP/IEEE International Conference on Management of Multimedia and Mobile Networks and Services (MMNS 2007), San José, California, USA, October 2007.
4. A. Gonzalez Prieto and R. Stadler “Real-time Network Monitoring Supporting Percentile Error Objectives", 14th HP Software University Association (HP-SUA) Workshop, 8-11 July 2007, Munich, Germany.
5. A. Gonzalez Prieto, R. Stadler, P. Kersch, R. Szabo, G. Nunzi, M. Brunner and S. Schuetz, "Distributed Network Management for AN" , 16th IST Mobile and Wireless Communication Summit, Budapest, Hungary, July 1-5, 2007.
6. A. Gonzalez Prieto and R. Stadler "Adaptive Distributed Monitoring with Accuracy Objectives", in the Proceedings of ACM SIGCOMM workshop on Internet Network Management (INM 06), Pisa, Italy, September 11, 2006.
7. R. Ocampo, L. Cheng, K. Jean, A. Gonzalez Prieto, A. Galis, Z. Lai, "Towards a Context Monitoring System for Ambient Networks", Chinacom, Beijing, China, October 25-27, 2006.
8. J. Nielssen, Z. Lajos Kis, A. Gonzalez Prieto, R.Stadler, M. Brunner, "Pattern-based Network Management for Ambient Networks", 15th IST Mobile and Wireless Communication Summit, Myconos, Greece, June 4-6, 2006.
9. A. Gonzalez Prieto, R. Stadler "Distributed Real-Time Monitoring with Accuracy Objectives," in the Proceedings of IFIP Networking, Coimbra, Portugal, May 15-19, 2006.
10. A. Gonzalez Prieto, R. Stadler "Design and Implementation of Performance Policies for SMS Systems," in the Proceedings of the 16th IFIP/IEEE Distributed Systems: Operations and Management (DSOM 2005), Barcelona, Spain, October 24-26, 2005.
11. M. Brunner, A. Galis, L. Cheng, J. Andres Colas, B. Ahlgren, A. Gunnar, H. Abrahamsson, R. Szabo, S. Csaba, J. Nielssen, A. Gonzalez Prieto, R. Stadler, G. Molnar, "Towards Ambient Networks Management," Second International Workshop on Mobility Aware Technologies and Applications (MATA 2005), Montreal, Canada, October 17-19, 2005.
12. A. Gonzalez Prieto, R. Stadler, "Scalable Policy Distribution for Ambient Networks," 14th IST Mobile and Wireless Communication Summit, Dresden, Germany, June 19-23, 2005.
13. A. Gonzalez Prieto, R. Stadler "Policy-based Management for SMS Systems," RVK 2005, Linkoping, Sweden, June 14-16, 2005.

14. A. Gonzalez Prieto, R. Stadler "Evaluating a Congestion Management Architecture for SMS Gateways," 9th IFIP/IEEE International Symposium on Integrated Network Management (IM 2005), Nice, France, May 15-19, 2005 (poster).
15. M. Brunner, A. Galis, L. Cheng, J. Andres Colas, B. Ahlgren, A. Gunnar, H. Abrahamsson, R. Szabo, S. Csaba, J. Nielssen, A. Gonzalez Prieto, R. Stadler, G. Molnar, "Ambient Networks Management Challenges and Approaches," First International Workshop on Mobility Aware Technologies and Applications (MATA 2004), Florianpolis, Brazil, October 20-22, 2004.
16. A. Gonzalez Prieto, R. Cosenza and R. Stadler "Policy-based Congestion Management for an SMS Gateway," in the Proceedings of the IEEE 5th International Workshop on Policies for Distributed Systems and Networks (POLICY 2004), Yorktown Heights, New York, June 7-9, 2004.

4. Open Research Questions for Future Research

Based on the work in this thesis, we have identified a set of open research questions regarding future work on decentralized real-time monitoring of global metrics under performance objectives.

- What is a suitable engineering framework for decentralizing monitoring tasks? Our work shows that a key step in designing a solution for real-time monitoring under performance objectives is formulating a global optimization problem and solving that problem in a distributed way. This includes the mapping of the global problem onto a set of local problems, which can be solved independently and asynchronously. Currently, such a mapping is custom-made for each case and remains more an art rather than a craft. In the literature several examples of such mappings can be found, but there is no fundamental understanding of the engineering principles behind this task. To exemplify the difficulties an algorithm designer faces when performing this mapping, consider the well-known case called “tragedy of the commons” [66], where individuals, by trying to maximize a local utility function, jeopardize the achievement of a global objective. While this phenomenon is known, there is no fundamental understanding on how to define the local problems in such a way that their solutions provide good approximations to the solution of the global problem.

- What are the fundamental performance limits of decentralized and centralized monitoring schemes, and how do they compare to each other? Commonly, decentralized schemes achieve better scalability and higher robustness at the cost of providing suboptimal performance. While this qualitative assessment is well-known, it remains a challenge to quantify it. For instance, in order to improve the adaptation time by x%, what performance degradation (in percentage) must be allowed?

- Which aggregation functions can be efficiently supported by distributed monitoring protocols? It is straightforward to see that some aggregates can be computed efficiently using in-network aggregation and aggregation trees. An example is the MAX across a set of scalars. Independently from the number of local variables to aggregate, each node only needs to report to its parent the partial aggregate for its subtree (in this case, this is a scalar). However, it is not straightforward how to compute efficiently other aggregates. For instance, the TOP-10 applications regarding traffic in a domain. The total traffic of an application can be obtained at the root node using the SUM with all the local variables accounting for traffic for that application. Using this approach, nodes in the aggregation tree need to report to their parents all applications they have information about.

- For which aggregation functions can we develop valid performance models for the monitoring process? Our work shows that such models are key to design monitoring systems that are controllable and achieve performance objectives. Aggregation functions of interest include histograms.

- Which type of monitoring protocols operate efficiently in wireless and mobile environments? These environments pose additional challenges to decentralized real-time monitoring. For instance, mobility can cause frequent aggregation tree reconstructions, affecting the performance of the monitoring task. Wireless links with high loss rates can also impact this performance.

The monitoring protocol developed as part of this thesis, A-GAP, can be improved in several ways. In the following, we give some examples.

- A key mechanism in the solution presented in this thesis is filtering. An alternative for trading estimation accuracy for monitoring overhead is to use a rate-control mechanism. It would be an interesting M.Sc. thesis project to develop a rate-based protocol with performance objectives and compare it to a filter-based one such as A-GAP. Our stochastic model and the design of A-GAP can be taken as a starting point for this task.

- A-GAP attempts to solve a network-wide optimization problem in a distributed fashion. Finding a centralized algorithm for optimally solving the network-wide problem would provide a benchmark for A-GAP and help in improving the design of A-GAP. The first obstacle for finding such centralized algorithm is to find a closed-form formulation for the network-wide optimization problem, a challenging task per se.

- In A-GAP, each node solves a local optimization problem. Currently, the solution is determined based on a grid search in a limited search space. Are there alternative and faster algorithms for solving the local problem? Similarly to the previous problem statement, a first step towards finding a faster algorithm could be to find a closed-form formulation for the local problem.

- The stochastic model used in A-GAP assumes independence among local variables. Such an assumption does not hold in many networking scenarios. Therefore, the question arises: what is the impact of correlated local variables on the performance of A-GAP? This is a challenging task, since the complexity of the problem grows with the square of the system size.

- Currently, A-GAP re-configures the node filters periodically. Alternatively, this re-configuration could be event-triggered. This approach could significantly reduce the overhead or adaptation time. Different solutions could be studied in the context of a M.Sc. thesis.

- Security. Distributed protocols involve nodes that cooperate to achieve a common goal. Faulty or malicious nodes may jeopardize the achievement of this goal. One could make A-GAP robust against such situations by adding mechanisms that identify and isolate faulty or malicious nodes.

Bibliography

[1] ANSI/IEEE Std 802.1D, 1998.

[2] B. Babcock, S. Babu, M. Datar, R. Motwani, and J. Widom, “Models and issues in data stream systems”, In 21st ACM Symposium on Principles of Database Systems, Madison, USA, June 2002.

[3] A. Boulis, S. Ganeriwal, and M. B. Srivastava, “Aggregation in sensor networks: an energy - accuracy tradeoff”, Elsevier Ad-hoc Networks Journal (special issue on sensor network protocols and applications), pages 317–331, 2003.

[4] D. Breitgand, D. Dolev, and D. Raz, “Accounting mechanism for membership size-dependent pricing of multicast traffic”, In Internet Charging and QoS Technology (ICQT), Munich, Germany, September 2003.

[5] M. Brunner, A. Galis, L. Cheng, J. Andres Colas, B. Ahlgren, A. Gunnar, H. Abrahamsson, R. Szabo, S. Csaba, J. Nielssen, A. Gonzalez Prieto, R. Stadler, and G. Molnar, “Ambient networks management challenges and approaches”, In First International Workshop on Mobility Aware Technologies and Applications (MATA 2004), Florianpolis, Brazil, August 2004.

[6] M. Brunner, A. Galis, L. Cheng, J. Andres Colas, B. Ahlgren, A. Gunnar, H. Abrahamsson, R. Szabo, S. Csaba, J. Nielssen, A. Gonzalez Prieto, R. Stadler, and G. Molnar, “Towards ambient networks management”, In Second International Workshop on Mobility Aware Technologies and Applications (MATA 2005), Montreal, Canada, October 2005.

[7] J. Considine, F. Li, G. Kollios, and J. Byers, “Approximate aggregation techniques for sensor databases”, In 20th International Conference on Data Engineering, Boston, USA, March 2004.

[8] G. Cormode , M. Garofalakis, S. Muthukrishnan, and R. Rastogi, “Holistic aggregates in a networked world: distributed tracking of approximate quantiles”, In ACM SIGMOD International Conference on Management of Data, Baltimore, USA, June 2005.

[9] S. Coulombe and G. Grassel, “Multimedia adaptation for the multimedia messaging service”, IEEE Communications, 42(7), July 2004.

[10] M. Dam and R. Stadler, “A generic protocol for network state aggregation”, In Radiovetenskap och Kommunication (RVK), Linkoping, Sweden, June 2005.

[11] A. Deligiannakis, Y. Kotidis, and N. Roussopoulos, “Hierarchical in-network data aggregation with quality guarantees”, In 9th International Conference on Extending Database Technology (EDBT), Crete, Greece, March 2004.

[12] S. Dolev, A. Israeli, and S. Moran, “Self-stabilization of dynamic systems assuming only read/write atomicity”, Distributed Computing, 7:3–16, 1993.

[13] N. Niebert, W. Mohr, L. Hiebinger, J. Von Häfen, A. Aftelak, D. Bourse, K. El-Khazen, M. Klemettinen, J. T. Salo, F. Carrez, F. Bataille, P. Hölttä, C. Prehofer, and N. Jefferies, “Ambient networks - research for communication networks beyond 3G”, In 13th IST Mobile and Wireless Communications-Summit 2004, Lyon, France, June 2004.

[14] M. Fang, N. Shivakumar, H. Garcia-Molina, R. Motwani, and J. D. Ullman, “Computing iceberg queries efficiently”, In 24th Int. Conf. Very Large Data Bases (VLDB), New York City, USA, August 1998.

[15] P. Flajolet and G. N. Martin, “Probabilistic counting algorithms for data base applications”, Journal of Computer and System Sciences, 31(2):182–209, October 1985.

[16] P. B. Gibbons and S. Tirthapura, “Estimating simple functions on the union of data streams”, In ACM Symposium on Parallel Algorithms and Architectures, Crete Island, Greece, June 2001.

[17] P. B. Gibbons and S. Tirthapura, “Distributed streams algorithms for sliding windows”, In ACM Symposium on Parallel Algorithms and Architectures, Winnipeg, Canada, August 2002.

[18] C. Intanagonwiwat, D. Estrin, R. Govindan, and J. Heidemann, “Impact of network density on data aggregation in wireless sensor networks”, In 22nd International Conference on Distributed Computing Systems, Vienna, Austria, July 2002.

[19] C. Intanagonwiwat, R. Govindan, and D. Estrin, “Directed diffusion: A scalable and robust communication paradigm for sensor networks”, In Sixth Annual International Conference on Mobile Computing and Networking (Mobicom), Boston, USA, August 2000.

[20] V. Klee and G.J. Minty, “How good is the simplex algorithm?”, In Inequalities III, pages 159–175. Academic Press, 1972.

[21] B. Krishnamachari, D. Estrin, and S. Wicker, “The impact of data aggregation in wireless sensor networks”, In International Workshop of Distributed Event-Based Systems, Vienna, Austria, July 2002.

[22] B. Krishnamachari, D. Estrin, and S. Wicker, “Modelling data-centric routing in wireless sensor networks”, Technical Report CENG 02-14, USC Computer Engineering, 2002.

[23] K. S. Lim and R. Stadler, “A navigation pattern for scalable internet management”, In IEEE/IFIP IM 2001, Seattle, USA, May 2001.

[24] K. S. Lim and R. Stadler, “Weaver - realizing a scalable management paradigm on commodity routers”, In 8th IFIP/IEEE International Symposium on Integrated Management (IM 2003), Colorado Springs, USA, March, 2003.

[25] K. S. Lim and R. Stadler, “Simpson - a simple pattern simulator for networks”, http://www.s3.kth.se/lcn/software/simpson.shtml, accessed on July 2008.

[26] A. Liotta, G. Pavlou, and G. Knight, “Exploiting agent mobility for large scale network monitoring”, IEEE Network, special issue on Applicability of Mobile Agents to Telecommunications, 16(3), May/June 2002.

[27] S. R. Madden, M.J. Franklin, J.M. Hellerstein, and W. Hong, “TAG: a tiny aggregation service for ad-hoc sensor networks”, In Fifth Symposium on Operating Systems Design and Implementation, Boston, USA, December 2002.

[28] M. J. Maullo and S.B. Calo, “Policy management: an architecture and approach”, In First IEEE International Workshop on Systems Management, Los Angeles, USA, April 1993.

[29] J. D. Moffett and M. S. Sloman, “Policy hierarchies for distributed systems management”, IEEE Journal on Selected Areas in Communications (J-SAC), 11(3), 1993.

[30] S. Nath, P. B. Gibbons, S. Seshan, and Z. Anderson, “Synopsis diffusion for robust aggregation in sensor networks”, In Second ACM Conference on Embedded Networked Sensor Systems, Baltimore, USA, November 2004.

[31] C. Olston, J. Jiang, and J. Widom, “Adaptive filters for continuous queries over distributed data streams”, In ACM SIGMOD 2003, San Diego, USA, June 2003.

[32] C. Olston, B. T. Loo, and J. Widom, “Adaptive precision setting for cached approximate values”, In ACM SIGMOD 2001, Santa Barbara, USA, May 2001.

[33] C. Olston and J. Widom, “Efficient monitoring and querying of distributed, dynamic data via approximate replication”, IEEE Data Engineering Bulletin, March 2005.

[34] R. Perlman. Interconnections. Addison Wesley Longman, 2000.

[35] M. A. Sharaf, J. Beaver, A. Labrinidis, and P. K. Chrysanthis, “Balancing energy efficiency and quality of aggregate data in sensor networks”, ACM International Journal on Very Large Data Bases, 13(4):384–403, December 2004.

[36] M. Sloman, “Policy driven management for distributed system”, Journal of Networks and Systems Management, 2(4), 1994.

[37] I. Solis and K. Obraczka, “The impact of timing in data aggregation for sensor networks”, In International Conference on Communications (ICC), Paris, France, June 2004.

[38] D.A. Spielman and S. Teng, “Smoothed analysis of algorithms: Why the simplex algorithm usually takes polynomial time”, In ACM Symposium on Theory of Computing, Crete, Greece, July 2001.

[39] N. Spring, R. Mahajan, and D. Wetherall, “Measuring ISP topologies with Rocketfuel”, In ACM/SIGCOMM, Pittsburgh, USA, August 2002.

[40] D. Verma, “Simplifying network administration using policy-based management”, IEEE Network, special issue on Policy-Based Networks, 16(2):20–26, 2002.

[41] J. Zhao, R. Govindan, and D. Estrin, “Computing aggregates for monitoring wireless sensor networks”, In First IEEE International Workshop on Sensor Network Protocols and Applications, Anchorage, USA, May 2003.

[42] T. W. Anderson, and L. A. Goodman, “Statistical Inference about Markov Chains”,The Annuals of Mathematical Statistics, Vol. 28, No.1, Mar 1957, pp 89-110.

[43] A. Liotta, G. Pavlou, and G.Knight, “A Self-Adaptable Agent System for Efficient Information Gatheing”, IEEE/ACM International Workshop on Mobile Agents for Telecommunication Applications (MATA'01), Montreal, Canada, August 2001.

[44] R. van de Meent, and A. Pras, Traffic Measurement Data Repository, University of Twente, http://traces.simpleweb.org/, accessed on December 2007.

[45] JSci - A science API for Java, http://jsci.sourceforge.net/, accessed on July 2008.

[46] JFreeChart, http://www.jfree.org/jfreechart/, accessed on July 2008.

[47] Cisco Netflow, http://www.cisco.com/en/US/products/ps6601/products_ios_protocol_group_home.html, accessed on July 2008.

[48] IETF IP Flow Information Export working group, http://www.ietf.org/html.charters/ipfix-charter.html, accessed on July 2008.

[49] K. Keys, D. Moore, and C. Estan, "A robust system for accurate realtime summaries of internet traffic", SIGMETRICS Perform. Eval. Rev., vol. 33, no. 1, pp. 85--96, 2005.

[50] M. Molina, A. Chiosi, S. D'Antonio and G. Ventre, "Design principles and algorithms for effective high-speed IP flow monitoring ",Computer Communications Volume 29, Issue 10, 19 June 2006, Pages 1653-1664.

[51] L.Yang, G. Michailidis, "Sampled based estimation of network traffic flow characteristics", IEEE Infocom 2007, Anchorage, USA, May 2007.

[52] K. Suh, D. R. Figueiredo, J. Kurose and D. Towsley, "Characterizing and detecting skype-relayed traffic", IEEE Infocom 2006, Barcelona, Spain, April 2006.

[53] T. Mori, M. Uchida, R. Kawahara, J. Pan, and S. Goto, "Identifying elephant flows through periodically sampled packets", 4th ACM SIGCOMM conference on Internet measurement, Taormina, Italy, October 2004.

[54] L. Ying, L. Zhen, D. Towsley and C. H. Xia, "Distributed Operator Placement and Data Caching in Large-Scale Sensor Networks", 27th IEEE INFOCOM 2008, Phoenix, USA, April 2008.

[55] N. Jain, D. Kit, P. Mahajan, P. Yalagandula, M. Dahlin and Y. Zhang, "STAR: Self-Tuning Aggregation for Scalable Monitoring", 33rd International Conference on Very Large Data Bases (VLDB 2007), Vienna, Austria, September 2007.

[56] T. M. Chen and S. S. Liu, "A Model and Evaluation of Distributed Network Management Approaches", IEEE Journal on Selected Areas in Communications , Vol. 20 No. 4 May 2002.

[57] Y. Zhu, T. Chen, and S. Lu, "Models and Analysis of trade-offs in distributed network management approaches", 7th IEEE/IFIP International Symposium on Integrated Network Management, Seattle, WA, USA, 2001.

[58] B. Zhang, and E. Al-Shaer, “Self-organizing Monitoring Agents for Hierarchical Event Correlation”. 18th IFIP/IEEE International Workshop on Distributed Systems: Operations and Management (DSOM 2007), San José, California, USA, October 2007.

[59] A. Liotta, G. Knight, G. Pavlou, “On the Performance and Scalability of Decentralised Monitoring Using Mobile Agents”, 10th IFIP/IEEE International Workshop on Distributed Systems: Operations and Management (DSOM '99), Zurich, Switzerland, October 1999.

[60] A. Liotta, G. Pavlou, G. Knight, “Active Distributed Monitoring for Dynamic Large-scale Networks”, IEEE International Conference on Communications (ICC'01), Helsinki, Finland, June 2001.

[61] A. Liotta, G. Pavlou, G. Knight, “A Self-Adaptable Agent System for Efficient Information Gathering”, IEEE/ACM International Workshop on Mobile Agents for Telecommunication Applications (MATA'01), Montreal, Canada, August 2001.

[62] K. G. Coffman and Andrew Odlyzko, “The Size and Growth Rate of the Internet”, DIMACS Technical Report 99-11, 1999, available at ftp://dimacs.rutgers.edu/pub/dimacs/TechnicalReports/TechReports/1999/99-11.ps.gz.

[63] A. Clemm, “Network Management Fundamentals”, Cisco Press, 2006.

[64] U. Warrier, and L. Besaw, “RFC1095: The Common Management Information Services and Protocol over TCP/IP (CMOT)”, April 1989.

[65] W. Stallings, “SNMP, SNMPv2, SNMPv3, and RMON 1 and “2, Addison-Wesley Pub Co; ISBN: 0201485346; 3rd edition (1998).

[66] G. Hardin, “The Tragedy of the Commons”, Science, 162, pp. 1243-1248, 1968.

[67] The World Factbook, CIA, available at https://www.cia.gov/library/publications/the-world-factbook/rankorder/2153rank.html, accessed on July 2008.

[68] Cable News Network, “Internet failure hits two continents”, January 31st, 2008, available at http://edition.cnn.com/2008/WORLD/meast/01/31/dubai.outage/index.html, accessed on July 2008.

[69] The New York Times, “Error in Skype’s Software Shuts Down Phone Service”, August 17th, 2007, available at

http://www.nytimes.com/2007/08/17/business/17ebay.html, accessed on July 2008.

[70] J. O. Kephart, D. M. Chess. "The Vision of Autonomic Computing," IEEE Computer, vol. 36, no. 1, pp. 41-50, January, 2003

[71] F. Chiang, R. Braun, J. I. Agbinya, ” Self-Configuration of Network Services with Biologically Inspired Learning and Adaptation”, Journal of Network and Systems Management, Vol. 15, No. 1, March 2007.

[72] K. Zimmermann, S. Felis, S. Schmid, L. Eggert, M. Brunner, ” Autonomic Wireless Network Management”, 2nd IFIP TC6 International Workshop on Autonomic Communication (WAC 2005), Vouliagmeni, Greece, October 2005.

[73] T.Suzuki, T. Suda, ” A middleware platform for a biologically inspired network architecture supporting autonomous and adaptive applications”, IEEE Journal on Selected Areas in Communications, Vol. 23, Issue 2, February 2005.

[74] D. Verma, “Policy-Based Networking: Architecture and Algorithms”, New Riders Publishing, 2000.

[75] J. Moffett, M. Sloman, ” Policy Hierarchies for Distributed Systems Management”, IEEE Journal on Selected Areas in Communications, Vol. 11, Issue 9, December, 1993.

[76] ”Google: one million servers and counting”, July 2nd 2007, available at http://www.pandia.com/sew/481-gartner.html, accessed on August 2008.

[77] M. Jelasity, A. Montresor and O. Babaoglu, “Gossip-based aggregation in large dynamic networks,” ACM Transactions on Computer Systems, Vol. 23, Issue 3, pp. 219-252, August 2005.

[78] D. Kempe, A. Dobra and J. Gehrke, “Gossip-Based Computation of Aggregate Information,” In 44th Annual IEEE Symposium on Foundations of Computer Science (FOCS’03), Cambridge, MA, USA, October, 2003.

[79] A. Ghodsi, S. El-Ansary, S. Krishnamurthy, and S. Haridi, “A Selfstabilizing Network Size Estimation Gossip Algorithm for Peer-to-Peer Systems,” SICS Technical Report T2005:16, 2005.

[80] R. van Renesse, K. Birman, and W. Vogels, “Astrolabe: A Robust and Scalable Technology for Distributed System Monitoring,” ACM Transactions on Computer Systems, Vol. 21, issue 2, pp.164-206, May 2003.

[81] F. Wuhib, M. Dam, R. Stadler, A. Clemm: “Robust Monitoring of Network-wide Aggregates through Gossiping,” 10th IFIP/IEEE International Symposium on Integrated Management (IM 2007), Munich, Germany, May, 2007.

[82] T. Erlebach, ” Autonomous Systems in the Internet: A Potential Subject for Studying Self-* Aspects”, International Workshop on Self-* Properties in Complex Information Systems. Bertinoro, Italy, May-June 2004.

[83] D. Jurca and R. Stadler, "Computing Histograms of Local Variables for Real-Time Monitoring using Aggregation Trees", KTH Technical Report, August, 2008.