Andrea Di Pietro, Felipe Huici, Diego Costantini, Saverio Niccolini:
DECON: Decentralized Coordination for Large-Scale Flow Monitoring. In: Proceedings of the IEEE Conference on Computer Communications (INFOCOM), March 2010.

DECON: Decentralized Coordination for Large-Scale Flow Monitoring

Andrea di Pietro Felipe Huici Diego Costantini Saverio Niccolini
NEC Europe, Heidelberg, Germany

***Abstract*—Monitoring at the flow level is crucial to ensure the correct operation of networks. Any sizable network relies on a number of monitoring probes, both to provide different observation points but also to scale to the ever-increasing number of flows that go through it. This situation gives rise to the difficult problem of assigning monitoring of flows to the available probes so that the network-wide coverage of flows (i.e., the number of flows actually monitored) is maximized.**

In this paper we introduce DECON, a decentralized and scalable coordination system aimed at solving this assignment problem. Unlike other approaches, DECON requires no network topology information, no traffic matrices and no packet marking. We present extensive simulation results showing that DECON scales up to large numbers of flows while requiring reasonable amounts of state from probes. Further, performance results from a prototypical monitoring probe built with commodity hardware show that even an inexpensive solution can accommodate DECON's requirements.

I. Introduction

Monitoring is essential to the correct operation of a network and the services that run on it, be it for gathering usage statistics, fault discovery, anomaly detection, or traffic engineering, to name a few. However, the unrelenting growth in IP traffic puts significant strain on the systems aimed at monitoring it; indeed, certain reports state that the volume of traffic nearly doubles every two years [1]. As a result, monitoring systems have to be scalable if they are to provide a viable mean of keeping track of the ever increasing traffic volumes.

Monitoring at higher granularities than packets, and in particular at the flow level, certainly alleviates the problem of coping with high traffic volumes. The widespread use of protocols like NetFlow [2] and sFlow [3] is evidence of the fact that monitoring at the flow level provides the necessary data to carry out essential network tasks, while at the same time reducing the load on the devices that gather such data.

While the flow abstraction helps, clearly any sizable network requires several monitoring probes to have different observation points but also in order to scale to the large number of flows that go through it. This situation raises the following question: given a set of monitoring probes and a set of flows going through them, which flows should a probe monitor at any given point in time? Such a mapping of flows to probes should be done with the aim of maximizing the number of flows actually monitored, as well as removing redundancies (for example, preventing a flow to be monitored simultaneously by two or more probes).

In essence this is a coordination problem, and in this paper we present DECON, a decentralized coordination system aimed at tackling it. Because the coordination happens in a decentralized manner, DECON scales to large numbers of flows and probes. In addition, the system requires neither topology information nor traffic matrices, as is the case with other approaches. We present extensive simulation results that show that despite having a decentralized coordination mechanism, DECON achieves a high degree of coverage (i.e., the number of flows that are actually monitored) even when faced with a large number of flows, including short-lived ones.

II. Related Work

The problem of coordinating flow monitoring tasks among a set of probes was also tackled in CSAMP [4]. Unlike DECON, CSAMP uses a centralized decision point which knows both the routing state and the traffic matrix of the network, and that is in charge of periodically computing the subset of flows each monitoring probe is responsible for. While sharing similar goals as CSAMP's, our system achieves them in a distributed, fault tolerant, and more scalable architecture with no need for detailed information about the network. Another solution in this space [5] suggests that monitoring probes use Bloom filters and a gossip protocol in order to exchange information about which flows they are monitoring, and thus coordinate their activities. While decentralized, this approach suffers from serious scalability problems, since the messaging overhead of a gossip protocol does not scale well with the number of probes nor with the number of flows to be monitored.

In [6] the authors propose a technique for choosing the monitoring points and their associated sampling rates according to optimality criteria. Unfortunately, the approach requires a-priori knowledge of the network routing state and does not address the issue of duplicate measurements (the authors assume that duplicates can be detected at the collector). The work in [7] proposes a double-hash based approach whose purpose is to ensure that the same packets are monitored by all of the probes, in order to provide multi-point measurements. Although this can be also achieved by our scheme, the reverse is not true: a double-hash based schema cannot ensure that every flow is monitored only once, unless the path of each flow is known beforehand.

III. DECON's Architecture

DECON's architecture is in charge of making decisions about which monitoring probes in the network should monitor which set of flows going through them. The aim is to spread the load across the available resources in order to increase coverage, which is the number of flows actually monitored during a certain time period. Further, DECON achieves this

goal in a decentralized way, without the need of calculating traffic matrices nor having knowledge of network topology.

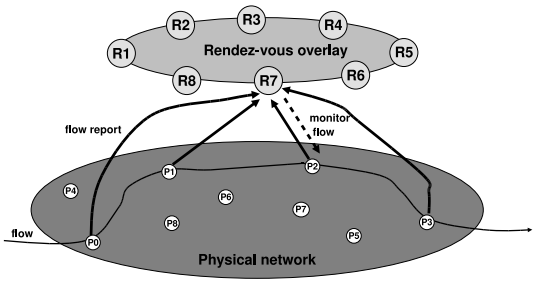

Fig. 1. DECON's architecture. Monitoring probes (P) send reports about flows to the rendez-vous overlay, which then decides which of the probes seeing a flow should monitor it.

To achieve this, DECON relies on a peer-to-peer network called the *rendez-vous* overlay (see figure 1). When a new flow arrives in the network and goes through a set of monitoring probes (P0, P1, P2 and P3 in the figure), each probe computes the flow's hash[1]. Each probe then sends a small report to the node in the overlay responsible for the value resulting from the hash, called a *rendez-vous point* (RP). The RP (R7 in the example) receives messages from all probes seeing a particular flow, and decides which of these (P2) should do the actual monitoring; it then sends messages back communicating the decision. For negative decisions, probes stop monitoring the flow and remove any state associated with it.

Clearly, a number of strategies are possible when deciding which probe should monitor the flow. Perhaps the simplest one is *first-fit*, where the RP assigns the flow to the probe whose message arrives first, a likely less-than-optimal strategy that has the advantage of reducing the decision delay (the time between when a flow is first seen at a probe and the decision message arriving at that probe). A more advanced strategy is *best-fit*, in which probes send a metric in the message reflecting their current load (e.g., the current number of flows being monitored, CPU utilization, etc), and have the RP choose the least-loaded probe as the one that should monitor the flow. This strategy spreads the monitoring load better across the probes, but increases the decision delay, since the RP now has to wait to make sure that all reports from probes seeing a flow have arrived before making a decision. In section IV we evaluate these two strategies, leaving more advanced strategies as future work.

DECON's decentralized decision process as well as its reliance on a p2p overlay allows it to scale to large networks while being resilient to failure. In addition to this, DECON's coordination mechanism has a couple of other beneficial features. First, the system can easily cope with flows changing their path through the network. Suppose that in figure 1 the path changed so that the flow went through P5 instead of P2. In this case, P5 would send a message to R7 telling it that it has seen a "new" flow. Because R7 keeps state about flows and its previous decisions, it knows that this is not a new flow. As a result, it will send a message to P2 to ensure that it is still seeing the flow. If it is, it may evaluate whether P5 is a better choice (e.g., less loaded) or decide to do nothing, keeping P2 as the "active" probe; if, on the other hand, P2 no longer sees the flow, R7 will evaluate which of P0, P1, P3 and P5 is the best choice to monitor the flow.

The second feature of the system is that, by nature of the decision process, it prevents undesired duplicate monitoring. It may, of course, sometimes be desirable to monitor a flow more than once (for example, to measure performance statistics at various points in the network). One of DECON's strengths is that it can accommodate a number of different decision strategies in order to suit different monitoring needs.

[1]The hash we used is based on the flow 5-tuple <src/dst IP address, src/dst port and protocol ID>, but any other flow definition can be used.

A. Batch Optimization

In order to reduce messaging overhead, reports can be batched. Since each probe accesses the rendez-vous overlay through a single *ingress node*, it is possible for the probe to bundle these reports into a single report, and send this batch report to the ingress node (the reports consist of very little information, so it is possible to store many of them in a single packet). Upon receival, the node parses the reports and sends each to the responsible RP. This same optimization can be implemented in order to reduce the number of response messages directed to a monitoring probe: the RP sends the response messages to the corresponding ingress points for each reporting probe; the ingress point then can, in turn, bundle the response messages into a single batch response.

As a result of this mechanism, the number of exchanged messages outside the overlay would then depend only on the batching period and no longer on the number of flows in the network. Further, this mechanism keeps most packets related to reports within the overlay, an infrastructure which has to fulfill only the coordination task and that can be easily scaled. Of course, such an optimization may increase the decision delay, as the reports are queued waiting for a batch message to be sent; however, we will show in the evaluation section that the overall performance is only marginally affected.

IV. Evaluation

We conducted extensive simulations to show that DECON can scale to a large number of flows. In this section we describe the simulation setup, the simulation results, and performance results from a prototypical monitoring probe that show that even commodity-hardware can fulfill DECON's requirements.

A. Simulation Setup

In order to assess the performance of our solution we implemented a special-purpose discrete-event simulator which models all the variables that affect the behavior of our system even under heavy traffic load. We simulated several network topologies composed of hundreds to thousands of nodes; to this end, we leveraged the simple and well-known Barabasi-Albert model [8], which allows to build huge scale-free graphs with a preferential attachment procedure. Even if such a model

does not exactly represent all of the topological features of a real network, it nonetheless reproduces a topology where a few hub nodes are crossed by a large number of paths, as is common in real networks.

Regarding link delays, we generated them randomly within a range of values that spanned up to ten milliseconds; we chose such a range of values after observing delay statistics published by the Internet2 network observatory (such values usually never exceed a few dozen milliseconds). For the communication between probes and the overlay we used larger latencies of up to 20 milliseconds, since we assumed that reports could cross several links before reaching the overlay. As for the overlay, we assumed the rendez-vous points to be organized in a Chord ring, where the delays for each hop are in the order of a few milliseconds.

We generated flows by picking up a random pair of end-points within the generated topology and by assuming a Pareto-distributed duration (the simplest mathematical model for a heavy tail distribution), with mean values of around 30 seconds. We made such a choice after analyzing traffic traces published by the Mawi group; such traces were captured on a trans-Pacific line in early 2009, thus representing up-to-date samples of real backbone traffic. As for the number of flows, we once again relied on Internet2 data which had about 9 million flows over a 5 minute time span, or about 30,000 flows/sec. Since we would like our system to scale up to very large topologies, we actually simulated much larger values (hundreds of thousands of flows per second over the whole topology).

B. Simulations

We used the simulator to evaluate several performance parameters of the system. One of the most relevant is the achievable flow coverage, in other words, the percentage of flows that can be monitored with a fixed amount of resources (we assume that each probe can monitor up to a certain limit of flows at the same time). In greater detail, we simulated a network with 300 monitoring probes, each of them capable of monitoring up to 10,000 flows. We evaluated the flow coverage that can be achieved by using our coordination scheme under the two flow assignment strategies mentioned in section III: first-fit and best-fit. Further, in order to more clearly illustrate DECON's impact, we ran simulations to see what happens when no coordination is used at all; the results for different traffic loads are shown in figure 2. It is evident that, while without coordination the number of missed flows grows quickly with the network load, DECON keeps these misses almost constant and significantly lower. In particular, the best-fit strategy, as expected, achieves the best performance when faced with very high flow rates.

Besides improving flow coverage, our solution prevents two or more probes from unnecessarily monitoring the same flow (DECON can of course also allow a flow to be observed at several probes when needed). In figure 3 we show the average number of times a single flow is measured when no coordination mechanism is used: even if such a figure improves with higher traffic rates (there are simply not enough resources for duplicate measurements) it is clear that, on average, even under high load, each flow is wastefully monitored more than once.

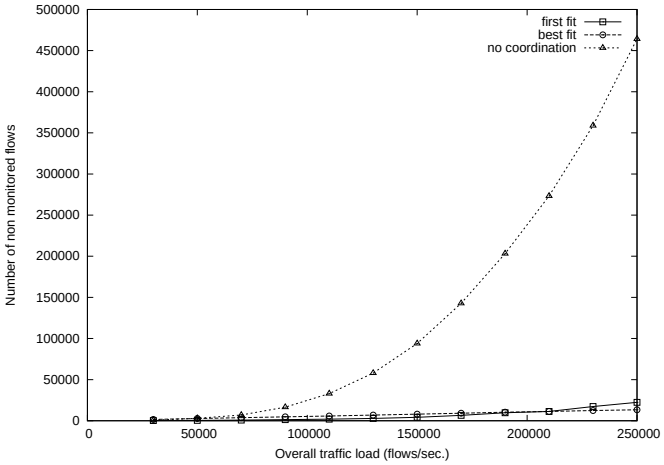

Fig. 2. Number of total flows actually monitored without coordination and using two different coordination strategies.

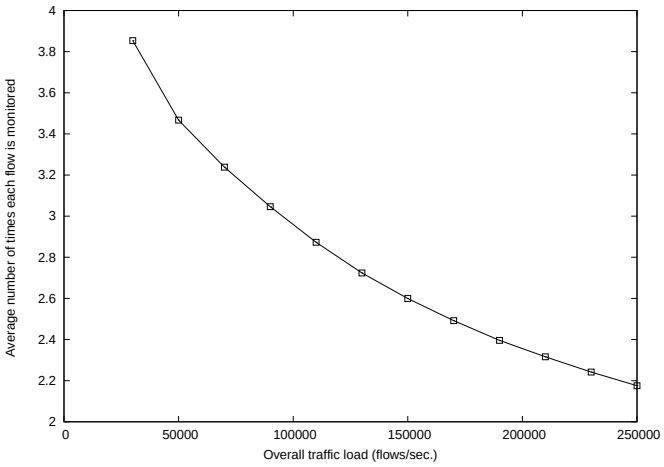

Fig. 3. Number of times a single flow is monitored without coordination.

We also evaluated the ability of our system to balance the burden of the monitoring activity among all the probes. Load balancing is not trivial to achieve because some nodes in the topology act as hubs, and, without a proper coordination scheme, are likely to be overloaded. Figure 4 shows the histogram of the average number of monitored flows for each probe in a scenario with 200 probes, each one able to concurrently monitor up to 10,000 flows and with a rate of 190,000 new flows per second over the overall network. Again, we plot the results achieved by the two different allocation strategies and those obtained with no coordination.

As expected, the best-fit scheme achieves the best balance among all the probes (it has the highest number of probes with a similar number of flows), while, with the first-fit allocation strategy, a small number of the probes (likely the hub nodes) are overloaded. With no coordination scheme, the mean resource occupation is much higher and a large fraction of the monitoring probes is always overloaded.

In order to provide a way of dimensioning our system, we

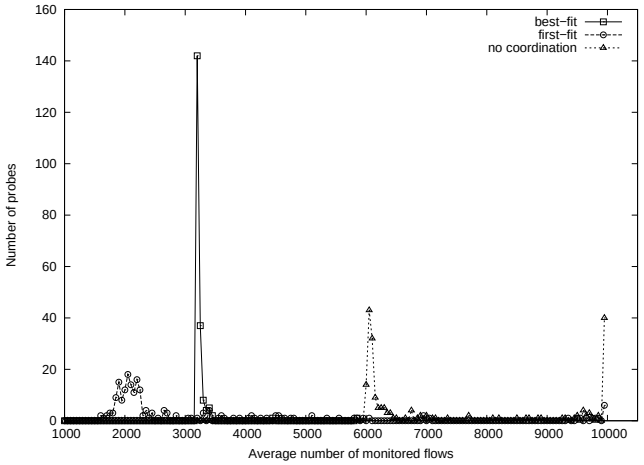

Fig. 4. Load-balancing histogram showing the number of flows each probe has to monitor with and without coordination.

ran a series of simulations without imposing any resource limitation on the probes (in terms of number of flows monitored), measuring how many resources would be needed on the probes in order to monitor all the traffic with no (or negligible) losses. More specifically, we computed the 99-percentile of the number of monitored flows with a varying number of probes (reaching up to 1,500 probes) and with a fixed load of 100,000 new flows/sec. Further, we used a best-fit allocation strategy, since, without buffer limitation, first-fit would simply allocate a flow to the first probe reporting it. The results are plotted in figure 5 and show that, by leveraging a large number of measurement probes and a proper coordination scheme, DECON can monitor high traffic volumes while requiring a small amount of resources from each probe. In the next section we will show that such a resource constraint can be met by using cheap commodity hardware.

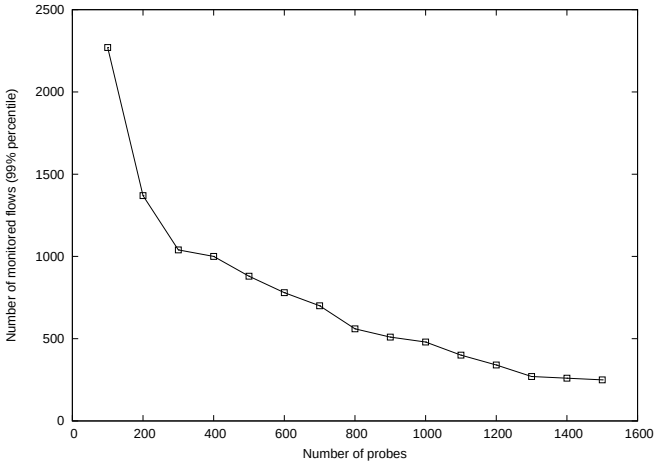

Fig. 5. 99-percentile of the number of monitored flows per probe for a varying number of probes and a fixed load of 100K flows/sec.

Another important parameter that we evaluated is message overhead (i.e., the number of messages that probes send to the coordination overlay). In particular, we computed the average number of messages per probe when having 200 and 400 probes and with different number of flows; the results are plotted in figure 6. As shown, the number of generated reports is well below 10,000/sec even for 180,000 flows. Since each report has a very small payload, this number corresponds to a rate of less than 1 MB/s.

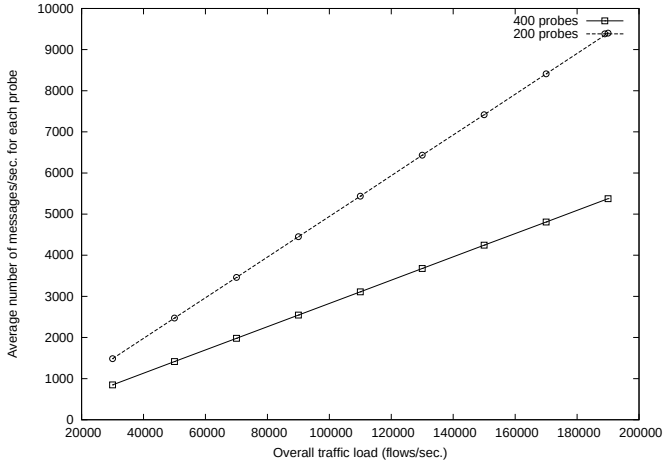

Fig. 6. Average number of messages per probe.

We also extended our simulator in order to support the batching optimization described in the previous section. In particular, we tried to evaluate the impact on the overall flow coverage that the additional delays incurred by this scheme had. To this end, we ran several simulations with different traffic loads and different batching periods. In each scenario, we evaluated the ratio between the number of missed flows with batching and the number of missed flows without batching for varying time periods (see figure 7). As expected, the performance gets worse with increasing batching periods, as responsiveness is being traded-off against lower overhead. However, we point out that even for fairly large batching periods (0.1 seconds corresponds to 10 messages per second per probe) the loss is relatively small, and this figure only improves with higher numbers of flows.

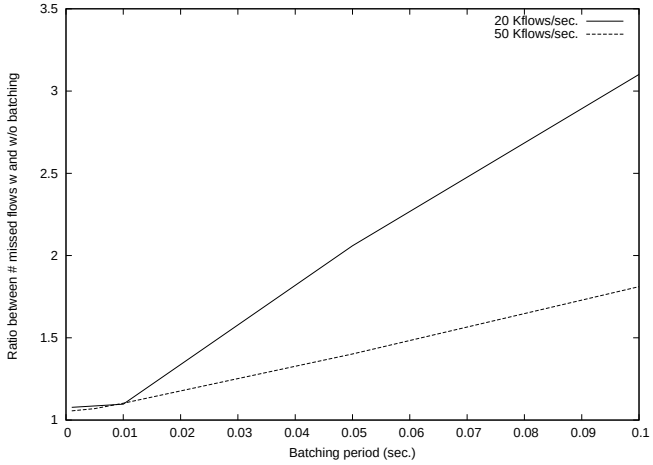

Fig. 7. Ratio between the number of missed flows with batching and the number of missed flows without batching.

C. Monitoring Probe

As shown in the simulations, DECON puts certain state requirements on monitoring probes, more specifically in terms

of how many flows a probe has to keep track of at any given point in time. In order to demonstrate that these are not unreasonable, we built a simple monitoring probe using the Click modular router software [9]. Click is based around the concept of *elements*, which are small units that perform different kinds of packet processing such as looking up an entry in a forwarding table, responding to ARP queries, or queueing packets; a Click configuration file then specifies how elements should be connected to each other.

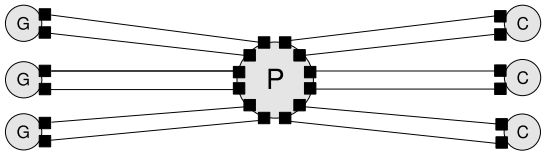

Fig. 8. Network topology used to test the probe's performance. G stands for generator, P for probe, and C for counter.

To implement the probe, we created a new Click element called *FlowMon*. The element is based around a hash, and keeps track of all flows that the probe is in charge of, updating simple statistics about them such as packet and byte counts. While the probe has timeout counters for detecting flow expiration and for when a decision takes too long to arrive from a rendez-vous point, we disabled these during this evaluation in order to test the worst-case performance where flows do not expire and the probe is responsible for all flows it sees.

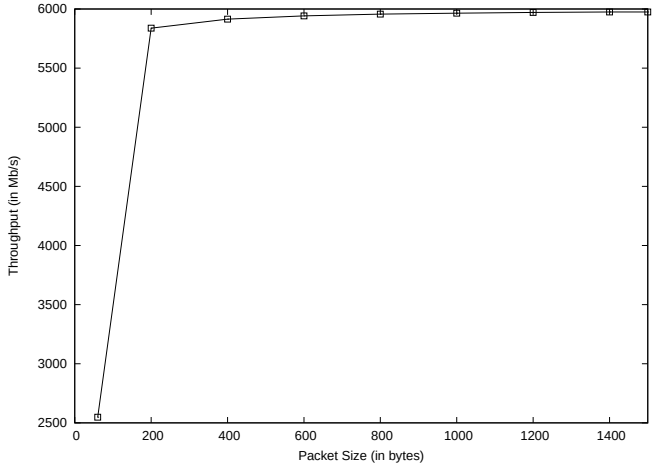

Fig. 9. Click monitoring probe throughput performance while monitoring 15,000 flows of different packet sizes.

In terms of hardware, we used a Dell 2950 with two Intel Xeon X5355 2.66GHz quad-core processors, 8 GB of main memory and 3 quad-port Intel 82571EB PCI express network cards. In addition, we used Dell 1950s to both generate and count traffic. Since the Dell 2950 acting as the probe had a maximum of 12 network interfaces, we connected three traffic generators and three traffic counters to it, as shown in figure 8 (the dell 1950s can generate packets at line rate for all packet sizes out of a maximum of two interfaces).

The aim was to test the performance of the monitoring probe when faced with a large number of flows of different packet sizes. While our generators (x86 servers running Click) could send packets at line rate for all packet sizes, due to memory limitations each of them could only generate 5,000 flows, for a total of 15,000 flows going through the monitoring probe. With this in place, we measured the probe's throughput for different packet sizes while keeping track of statistics for all of these flows (figure 9). As can be seen, even for minimum-sized packets the probe reaches a very reasonable 2.5Gb/s; this figure quickly ramps up to the line rate value of 6Gb/s for 200-byte packets and larger. These results show that the state requirements arising from DECON's coordination (recall from figure 5 a maximum of about 2,300 flows going through any one probe) can be met even by inexpensive, off-the-shelf hardware.

V. Conclusions

We presented DECON, a decentralized and scalable coordination system that dynamically assigns the monitoring of a set of flows to a set of probes. In contrast with previous approaches, DECON requires no traffic matrices, no network topology, and no packet marking. We have shown through extensive simulation that DECON is scalable to large numbers of flows, and that the requirements it places on monitoring probes can even be accommodated by a probe built on commodity hardware.

One issue that we did not discuss is the handling of multi-path flows, whereby some packets from a flow go through one probe, while others through a different probe; we leave a solution to this problem as future work. In future work we also intend to investigate more advanced decision strategies than first-fit and best-fit, as well as evaluating the system's performance for other network topologies.

References

[1] Cisco Systems, "Approaching the zettabyte era," "http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-481374.pdf", June 2008.

[2] Cisco, "Cisco IOS Netflow," "http://www.ciscom.com/web/go/netflow", October 2009.

[3] sFlow.org, "Making the Network Visible," "http://www.sflow.org", October 2009.

[4] V. Sekar, M. K. Reiter, W. Willinger, H. Zhang, R. R. Kompella, and D. G. Andersen, "CSAMP: a system for network-wide flow monitoring," in *NSDI'08: Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation*. Berkeley, CA, USA: USENIX Association, 2008, pp. 233–246.

[5] M. Sharma and J. Byers, "Scalable Coordination Techniques for Distributed Network Monitoring," *Passive and Active Measurement Conference*, 2005.

[6] G. R. Cantieni, G. Iannaccone, C. Barakat, C. Diot, and P. Thiran, "Reformulating the monitor placement problem: Optimal network-wide sampling," in *In Proc. of CoNeXT*, 2006.

[7] R. Serral-Gracia, P. Barlet-Ros, and J. Domingo-Pascual, "Distributed sampling for on-line sla assessment," in *Local and Metropolitan Area Networks, 2008. LANMAN 2008. 16th IEEE Workshop on*, Sept. 2008, pp. 55–60.

[8] A.-L. Barabsi and R. Albert, "Emergence of Scaling in Random Networks," *Science*, vol. 286, no. 5439, pp. 509–512, 1999. [Online]. Available: http://www.sciencemag.org/cgi/content/abstract/286/5439/509

[9] R. Morris, E. Kohler, J. Jannotti, and M. F. Kaashoek, "The click modular router," *SIGOPS Oper. Syst. Rev.*, vol. 33, no. 5, pp. 217–231, 1999.