Anemone: Using End-systems as a Rich Network Management Platform

Richard Mortier
Microsoft Research
Cambridge, UK
mort@microsoft.com

Rebecca Isaacs
Microsoft Research
Cambridge, UK
risaacs@microsoft.com

Paul Barham
Microsoft Research
Cambridge, UK
pbar@microsoft.com

ABSTRACT

Enterprise networks contain hundreds, if not thousands, of cooperative end-systems. We advocate devoting a small fraction of their idle cycles, free disk space and network bandwidth to create *Anemone*, a platform for network management. In contrast to current approaches which rely on traffic statistics provided by network devices, Anemone combines end-system instrumentation with routing protocol collection to provide a semantically rich view of the network.

Categories and Subject Descriptors: C.2.3 Network Operations—Network management

General Terms: Management, Measurement, Performance.

Keywords: Distributed enterprise network management.

1. INTRODUCTION

Many network management tasks require deeper understanding of the state of the network that can be acquired solely from information available in the core of the network. Modern networks are becoming more and more difficult to understand from the network core due to the increasing use of tunnelling and encryption. The effect of the network on an individual application's end-to-end performance (e.g. the delay associated with a VoIP call) requires data *only* available in the end-systems hosting the application (e.g. IPSec decryption keys). Consequently, we claim that augmenting end-systems with in-band monitoring will provide a more complete view of the network, support sophisticated network management queries, and supply the global statistics necessary to automate network control. This paper describes Anemone, its potential benefits and challenges, and results from an initial evaluation.

2. SYSTEM OVERVIEW

Essentially, Anemone treats the end-systems in the network as a set of 'traffic sensors'. It combines flow data from these systems with topology data recovered from the routing protocol to provide a rich dataset for mining by network management applications. Initially, we focus on enterprise networks, using their centralized host administration to get a coherent picture of the traffic entering and leaving the network.

It is highly appropriate to locate support for flow-based network management in end-systems for several reasons. Each end-system has all the information required to decrypt and de-encapsulate opaque protocols that use tunnelling and encryption (e.g. IPSec keys). Dynamic port negotiation, the need for temporal correlation between flows, and client/server performance variation all make it difficult to understand the impact of network behaviour on an application's end-to-end performance when monitoring in the network core. However, end-systems can accurately assign flows to applications – even particular sub-parts of applications – as well as gathering data from a variety of sources (e.g. TCP round-trip-time estimators, per-flow bandwidth estimators). This enables a much richer set of queries to be posed and answered (e.g. 'what is the contribution of link l to the delay experienced by VoIP calls between hosts h_1 and h_2?'). Finally, end-systems have plentiful CPU, memory and disk, in contrast to resource starved core routers.

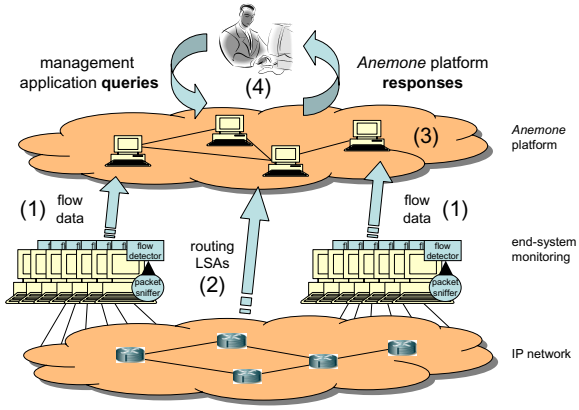

Figure 1: Anemone system architecture: (1) end-system instrumentation provides traffic flow statistics, (2) routing protocol monitoring provides current topology data, for (3) a data mining platform to (4) respond to queries posed by applications.

Anemone comprises 3 components, depicted in Figure 1:

1. End-system instrumentation recording locally transmitted and received flows together with the associated service or application.

2. A passive routing protocol monitor collecting link state updates to reconstruct the dynamic routing topology.

3. A data mining platform combining these two datasets (flow data and topology), and providing APIs support-

SIGCOMM'05 Workshops, August 22–26, 2005, Philadelphia, PA, USA.
ACM 1-59593-026-4/05/0008.

ing the queries generated by a variety of network management applications.

More detail on the end-system instrumentation and OSPF routing protocol monitor prototypes can be found in our associated technical report [1]. Anemone contrasts with current approaches by leveraging end-system control to monitor network traffic in combination with its dynamic topology, gathered by passive monitoring of routeing protocols rather than relying heavily on extensive device support to monitor from the core of the network. A variety of tools could be built on such a platform, covering areas such as:

Visualization and logging, giving both real-time *and* historical views of network topology and traffic distribution through that topology. This would be invaluable for network monitoring and management, and for collecting and accessing the historical data required by facilities such as traceback, capacity planning and anomaly detection.

Analysis and simulation, enabling 'what-if' analysis of the network to investigate and *predict* potential changes in topology, configuration, and traffic distribution. This would be a powerful tool for answering questions such as *"what happens to the network if we consolidate all the mail servers?"*, or concerning network response to failures or planning decisions. By feeding live flow and topology information into a flow-optimization solver the current network configuration could be evaluated for efficiency and robustness, and proposed network modifications could be tested for days or weeks using real time traffic and topology data.

Traffic engineering, automating network configuration to control traffic to meet network-wide latency targets, link utilization targets, etc. This would allow *dynamic* reconfiguration of the network as it and its traffic patterns evolve to ensure that service level agreements are always met in the most efficient way. It might even be possible to actively respond to detected traffic anomalies to reconfigure the network to contain the effects of malicious traffic.

3. DATA MINING PLATFORM

The tension in designing the data mining platform is between efficient resource usage and the robustness and ease of management of the data store. Logically, Anemone aggregates flow data from each end-system to construct the complete traffic matrix, $A_{ij} = \{\text{bandwidth from src } i \text{ to dst } j\}$, annotates each entry, (a_{ij}), with the route from i to j, and executes application-supplied queries against this dataset. A distributed query system augmented by the ability to perform the necessary route computation might provide a sound basis for the data mining platform in Anemone.

A significant challenge for our approach is determining the required proportion of hosts to instrument in order to give acceptable network coverage and accuracy in query results. The problem is analogous to that of sampling packets in order to infer flow volumes, but in this case we can exploit the asymmetric nature of enterprise network traffic patterns and topologies to overcome incomplete monitoring coverage. Data distribution among nodes will depend very much on the characteristics of the data and on the queries executed on the data. Given the datasets we have collected so far, it appears reasonable to distribute the relatively static topology data to end-systems, where it can be locally combined with the much more dynamic flow data that each end-system collects. The development of our data mining platform is work in progress, but we have performed an initial exploration of these issues through simulation.

The simulation constructs a centralized database containing the augmented traffic matrix, and exposes a simple API allowing the database to be queried. The database is populated with real topology data recovered from our network by our prototype OSPF monitor, and synthetic traffic traces. This permits us to explore various design decisions and tradeoffs concerning the degree of data distribution and aggregation required, some of the communication overheads of the platform, and the nature of the APIs provided to query the platform. Currently we are studying the most efficient way to compute basic queries including 'what is the load on link l?', 'what is the load {forwarded,sourced,sunk} at router r?', 'which are the top-N busy links?'. A number of these queries utilize an optimization made possible by knowing the network topology: the predecessor matrix (an output of the Dijkstra computation) allows the set of hosts that might possibly be using a link to be pre-computed, reducing the communication overhead required by such queries.

The synthetic traffic model used in the simulation captures flow inter-arrival time, flow size and flow transmission rate. By coupling this with the OSPF data, we also incorporate a simple notion of the distribution of end-points of a flow (whether, given a flow's source, its destination is within the subnet, within the area, within the AS, or external to the area). We used this model to generate multiple 1 hr simulated traces for 2000 hosts in the given topology, varying the proportion of randomly selected instrumented hosts from 10–100%.

From this study we see that the accuracy of the system does not depend linearly on the proportion of instrumented hosts: unsurprisingly, both the particular hosts that are instrumented (equivalently, the topology of the network) and the traffic patterns combine to make this relationship quite non-linear (perhaps 5% of hosts observe over 97% of the traffic). To begin to validate this, we took a 24 hour packet trace of inter-VLAN and WAN traffic originating on our LAN containing 447.5 GB of transmitted traffic, and referencing 15,184 transmitting or receiving hosts. This trace shows that 40 hosts, 16 of which are servers, observe 95% of this traffic, and the top 5 hosts, all of which are local servers, account for 66% of the traffic. We believe that this high degree of asymmetry is typical of enterprise networks, and thus careful selection of hosts to instrument (i.e. instrumenting the servers) should allow us to achieve high coverage using only a small percentage of instrumented machines.

4. SUMMARY

We have described *Anemone*, an end-system platform for network management. Anemone uses end-systems as real-time 'network sensors' to collect data about the network's topology, and traffic in terms of flows. These datasets permit a variety of sophisticated network management queries to be answered, and allows the impact of the network on application performance to be better understood. More details can be found in the associated technical report [1].

5. REFERENCES

[1] R. Mortier, R. Isaacs, and P. Barham. *Anemone*: using end-systems as a rich network management platform. Technical Report MSR-TR-2005-62, Microsoft Research, Cambridge, 7, JJ Thomson Ave, Cambridge, CB3 0FB. UK., May 2005.