Anemone: using end-systems as a rich network management platform

Richard Mortier, Rebecca Isaacs, Paul Barham
Microsoft Research, Cambridge, UK

May 18, 2005

1 Introduction

Enterprise networks contain hundreds, if not thousands, of cooperative end-systems. This paper advocates 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 that 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.

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 deploying more and more protocols that are difficult to track from the network core due to the use of tunnelling and encryption. The effect the network has on an individual application's end-to-end performance (e.g. the delay associated with a VoIP call) requires data that is only available in the end-systems actually 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, discusses potential benefits and challenges, and presents an initial evaluation of the platform.

2 System overview

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

End-systems are a highly appropriate place to locate support for flow-based network management 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). It also has plentiful CPU, memory and disk, in contrast to resource starved core routers. 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. Host-based instrumentation can make use of a variety of data sources (e.g. TCP round-trip-time estimators, per-flow bandwidth estimators), and can accurately assign flows to applications, or even particular sub-parts of applications. 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?').

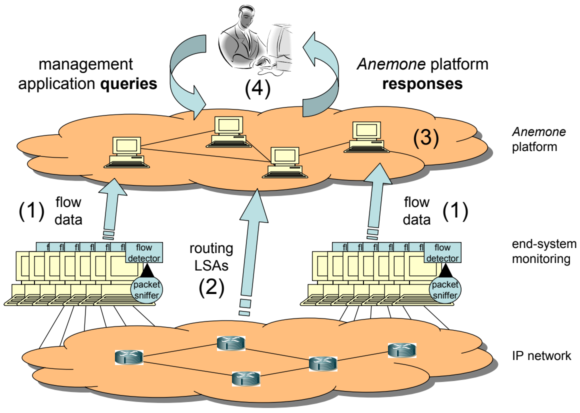

Figure 1. Anemone system architecture: (1) end-system instrumentation provides traffic flow statistics, (2) routing protocol monitoring provides current topology data, for (3) a data mining platform to (4) respond to queries posed by applications.

The Anemone platform comprises 3 main 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 that collects link state advertisements to reconstruct the current routing topology.

3. A data mining platform that combines these two datasets (flow data and topology), and provides

APIs that support the queries generated by a variety of network management applications such as simulation, visualization and planning tools.

More detail on our end-system instrumentation and OSPF routing protocol monitor prototypes can be found in our associated technical report [1].

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 first aggregates and combines flow information collected by 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 finally executes queries supplied by applications against this dataset. Distributed query systems augmented by the ability to perform the necessary route computation might provide a sound basis for the data mining platform in Anemone.

One 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 trade-offs 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.

We used a traffic model that captures flow inter-arrival time, flow size (a heavy-tailed distribution as expected), and flow transmission rate. This is coupled with the OSPF data to 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). Using this model, we synthesized multiple 1 hr simulated traces placing 2000 hosts in the given topology and varying the proportion of instrumented hosts between 10–100%. The instrumented hosts were selected at random from the complete set of 2000 hosts.

A key result from this study is 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. At any given moment, perhaps 5% of hosts are observing 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 reference to 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].

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.