

PaDIS Emulator: An Emulator to Evaluate CDN-ISP Collaboration

Ingmar Poesse
T-Labs/TU Berlin
ingmar@net.t-labs.tu-berlin.de

Benjamin Frank
T-Labs/TU Berlin
bfrank@net.t-labs.tu-berlin.de

Simon Knight
University of Adelaide
simon.knight@adelaide.edu.au

Niklas Semmler
TU Berlin
nsemmler@net.t-labs.tu-berlin.de

Georgios Smaragdakis
T-Labs/TU Berlin
georgios@net.t-labs.tu-berlin.de

Abstract

We present PaDIS Emulator, a fully automated platform to evaluate CDN-ISP collaboration for better content delivery, traffic engineering, and cost reduction. The PaDIS Emulator enables researchers as well as CDN and ISP operators to evaluate the benefits of collaboration using their own operational networks, configuration, and cost functions.

The PaDIS Emulator consists of three components: the network emulation, the collaboration mechanism, and the performance monitor. These layers provide scalable emulation of the interaction between an ISP or a number of ISPs with multiple CDNs and vice versa. PaDIS Emulator design is flexible in order to implement a wide range of collaboration mechanisms on virtualized or real hardware, and evaluate them before introduction to operational networks.

Categories and Subject Descriptors

C.2.1 [Computer-Communication Networks]: Network Architecture; C.2.3 [Network Operations]: Network Management

General Terms

Performance, Experimentation, Management.

Keywords

Traffic Engineering, Content Distribution, Network Optimization.

1. BACKGROUND

Recent studies [5, 6] show that a large fraction of Internet traffic originates from a small number of Content Delivery Networks (CDNs). To cope with increasing demand for content, CDNs have deployed massively distributed hosting infrastructures. These infrastructures pose challenges for CDNs as they must dynamically map end-users to servers without full awareness of network conditions within an ISP and end-user locations [1]. ISPs also struggle to cope with the rapid traffic shifts caused by the dynamic server selection policies of CDNs. Their traffic engineering is based on the presumption that traffic is predictable. It adapts the routing and operates on time scales of several hours [3], thus, it is too slow to react to rapid traffic changes caused by CDNs.

The challenges faced separately by CDNs and ISPs can become an opportunity for collaboration. This will improve operation, re-

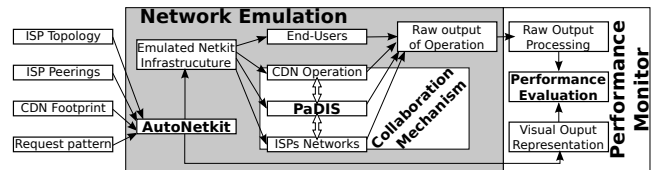


Figure 1: PaDIS Emulator Workflow.

duce costs, and enhance end-user experience. In previous work, we designed and implemented a recommendation system called Provider-aided Distance Information System (PaDIS), run by an ISP to enable CDN-ISP collaboration [9]. PaDIS offers an interface for CDN-ISP collaboration without the need to share any operational information from either party and therefore avoids the confidentiality concerns about leaking operational secrets. Our previous work showed significant performance gains for ISPs, CDNs, and end-users [4]. Also, CDNs' mapping improved and ISPs gained another tool in their traffic engineering suite.

In this demo we present a fully-automated framework that emulates the topology, operation, and policies of both ISPs and CDNs. Our emulation framework, PaDIS Emulator, can be used to quantify the benefits of CDN-ISP collaboration. It also helps identify the best practices to achieve the maximum benefit through collaboration. Our architecture is scalable to emulate a large number of CDNs and ISPs. Thus, it can be used as a collaboration and decision making tool of both ISPs and CDNs.

2. ARCHITECTURE

PaDIS Emulator has been designed with three-layers: (1) Network Emulation, (2) the Collaboration Mechanism, and (3) a Performance Monitor, see Figure 1. Only the first layer uses emulation; the other two can run either on an emulated or hardware-based infrastructure. This logical separation allows us to use real router and network service software in an emulated environment for evaluation, but with the opportunity to easily apply it to a real deployment.

2.1 Network Emulation

In the Network Emulation, both the ISP topologies and CDN footprints are created using virtual machines and links. To simplify the emulated setting, a graphical editor, Yed, is used for modeling. Alternatively, the configuration can be converted directly from operational data, such as IGP, BGP.

For the basic setup, PaDIS Emulator can run on high-level representation of ISPs and their interconnection, the deployment foot-

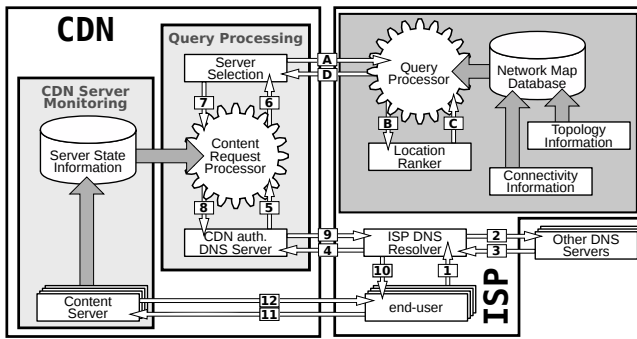


Figure 2: PaDIS Collaboration Scheme.

print of CDNs and the demand patterns of users to CDNs. However, additional information, such as specific DNS deployment, BGP policies, peering costs, etc., can easily be integrated. Once the specifications have been completed, we use AutoNetkit [8], an auto-configuration tool for Netkit, to allow the entire virtual setup to be fully automated. Netkit runs the Quagga routing daemon inside User-Mode Linux virtual machines to emulate ISP networks. Furthermore, it can run any additionally needed software. AutoNetkit allows for a fully automated setup of emulated networks, including IP addressing, routing, peering and DNS configuration. We added extra features to support PaDIS Emulator including auto-deploying software, and configuring link properties and meta information on peering links.

The footprint of a CDN is installed into the PaDIS Emulator by deploying custom or generic CDN software. Today, the CDN deployment is very complex [7]. Some CDNs, e.g., Limelight and Google, rely on large datacenters and connect to ISPs either through private peering or Internet Exchange Points. Others, such as Akamai, deploy their infrastructure deep inside an ISP's network. PaDIS Emulator can configure both of these CDN strategies, either individually, joined or as hybrids. In fact, PaDIS Emulator allows multiple independent CDNs, with different combinations of deployment and strategies to be run inside the same emulation.

A crucial task in CDN operation is the mapping of users to the server. This is mostly done through DNS and/or HTTP redirection. PaDIS Emulator contains a fully-functional DNS hierarchy. The positioning of DNS servers and resolvers can either be done automatically or through a user-specified custom configuration. This feature is important: the performance of content delivery relies heavily on the deployment and operation of ISP-operated DNS [9, 1]. Finally, request patterns and profiles of ISP users for content hosted in the CDNs can be pre-defined or modelled as statistical function.

2.2 Collaboration Scheme

The Collaboration Scheme layer is the heart of PaDIS Emulator. ISPs run PaDIS, while CDNs are enabled to utilize this novel service. PaDIS uses its network monitoring to collect *Connectivity* and *Topology Information* and keeps an up-to-date network database. This information is used to aid the CDN in its server selection process by ranking candidate servers in relation to the network state, requested content, and the desired performance metric. On the CDN side, operation works as it does today (see Figure 2, steps 1-10), with the option of obtaining ISP recommendation when users are assigned to servers, (steps A-D). To quantify the efficiency of PaDIS, the collaboration scheme can be dynamically enabled or disabled on a per-CDN basis.

When the collaboration mechanism is enabled, the set of servers, determined by the Server Selection, able to satisfy the content request is sent to PaDIS. The CDN, instead of performing its own server selection, utilizes PaDIS to obtain ISP guidance. If needed, the CDN can also specify the performance metrics relevant for the mapping, e.g., delay, bandwidth, etc., or any combination. The CDN also sends the location of the request: either the DNS server that issued the request, or, if the DNS client extension is supported [2] the subnet/IP address of the end-user. When PaDIS receives the request, it ranks the candidate servers based on the end-user location, the performance metrics and the state of the network. Finally, the ranked list is returned to the CDN which, in turn, compiles the final answer for the end-user.

2.3 Performance Monitor

The Performance Monitor layer consists of a number of network and content delivery performance metrics which are measured and reported. The monitor will show the layout of the network used in the *Network Emulation*, annotated with regularly updated statistics, e.g., link utilization, traffic load on peering points and estimated costs. Furthermore, CDN performance statistics are shown, including the server load, delivering cost, and the user-server mapping efficiency.

3. DEMONSTRATION

Our demonstration runs on virtualized hardware, but the general concept is not limited to it. CDNs and ISPs are able to evaluate the benefits of collaboration through PaDIS Emulator while taking into account CDN peerings, capacity and mapping strategies as well as ISP topology, routing, peering agreements and policies. PaDIS Emulator allows the utilization of different optimization functions by CDNs and ISPs. To visualize the results, PaDIS Emulator collects and reports the utilization of each ISP link, the load on the CDNs, the stress on the peering links and the end-user download time. Furthermore, to quantify the effect of CDN-ISP collaboration, it allows the user to enable and disable the collaboration at run time, and reports the gains due to collaboration.

During the demo session we present to the audience a pre-configured network in PaDIS Emulator and how the reported performance metrics change when modifying e.g., the mapping strategy, the optimization function or even completely disabling the collaborative optimization. We also encourage everyone to try their own settings/modifications in PaDIS Emulator and observe the impact on the emulated network and the reported performance values.

4. REFERENCES

- [1] B. Ager, W. Mühlbauer, G. Smaragdakis, and S. Uhlig. Comparing DNS Resolvers in the Wild. In *ACM IMC*, 2010.
- [2] C. Contavalli, W. van der Gaast, S. Leach, and D. Rodden. IETF draft: Client IP Information in DNS Requests, 2011.
- [3] B. Fortz and M. Thorup. Optimizing OSPF/IS-IS Weights in a Changing World. *IEEE JSAC*, 20(4), 2002.
- [4] B. Frank, I. Poese, G. Smaragdakis, S. Uhlig, and A. Feldmann. Content-aware Traffic Engineering. In *ACM SIGMETRICS*, 2012.
- [5] A. Gerber and R. Doverspike. Traffic Types and Growth in Backbone Networks. In *OFC/NFOEC*, 2011.
- [6] C. Labovitz, S. Lelkel-Johnson, D. McPherson, J. Oberheide, and F. Jahanian. Internet Inter-Domain Traffic. In *ACM SIGCOMM*, 2010.
- [7] T. Leighton. Improving Performance on the Internet. *CACM*, 2009.
- [8] H. Nguyen, M. Roughan, S. Knight, N. Falkner, O. Maennel, and R. Bush. How to Build Complex, Large-Scale Emulated Networks. In *TridentCom*, 2010.
- [9] I. Poese, B. Frank, B. Ager, G. Smaragdakis, and A. Feldmann. Improving Content Delivery using Provider-Aided Distance Information. In *ACM IMC*, 2010.