

Performance Based Traffic Control with IDIPS

Damien Saucez
Université catholique de Louvain
ICTEAM
2 Place Sainte-Barbe
B-1348 Louvain-la-Neuve, Belgium
damien.saucez@uclouvain.be

Olivier Bonaventure
Université catholique de Louvain
ICTEAM
2 Place Sainte-Barbe
B-1348 Louvain-la-Neuve, Belgium
olivier.bonaventure@uclouvain.be

ABSTRACT

Nowadays Internet is ubiquitous resulting in an increasing path diversity and content duplication. However, while content can be retrieved from many different places, the paths to those places are not equivalent. Indeed, some paths offer better bandwidth while others are less expensive or more stable. In addition, a new range of applications is sensitive to the performance of the paths that carry their traffic. To support this evolution of the Internet, we propose ISP-Driven Informed Path Selection (IDIPS). Any ISP can easily deploy IDIPS to help its customers to select the paths that best meet their requirements in order to reach their content. IDIPS helps in this selection through pro-active measurements and ISP-defined policies. IDIPS is scalable and can support thousands of clients. IDIPS is also flexible and can thus be used by the ISP to optimize its routing decisions to take the performance of its inter-domain links into account.

Categories and Subject Descriptors

C.2.1 [Computer-Communication Networks]: Network Architecture and Design

General Terms

Measurement, Performance

Keywords

route control, traffic engineering, XORP

1. INTRODUCTION

Content is often replicated and distributed with the help of efficient content distribution networks (CDN) [3, 5] or even end-user based replication with peer-to-peer applications. This creates several problems in Internet Service Provider (ISP) networks where client-server asymmetry does not hold anymore. In addition, due to the transition from IPv4 to IPv6 many hosts will be dual-stack for the foreseeable future. Furthermore, measurements show that IPv4 and IPv6 do not always provide the same performances, even for a single source-destination pair. To reach a destination supporting both IPv4 and IPv6, a source host can achieve better performance by selecting the stack that provides the

best performance. In such a context, applications can improve their performance by selecting the most appropriate paths. For instance, bulk data transfer peer-to-peer clients will favor paths with the largest bandwidth so that the targeted file will be downloaded faster.

A way to enable efficient path selection for applications would be to allow the network to cooperate with them [2, 7, 8]. Such a cooperation would also give the opportunity to operators for managing incoming and outgoing traffic on their networks. Indeed, according to their traffic engineering needs, operators could balance traffic from one link to another.

We propose an informed path selection service called *ISP-Driven Informed Path Selection (IDIPS)*. IDIPS is generic and does not require fundamental changes in the current Internet architecture (only the clients need to implement a library for using the service).

IDIPS is designed as a request/response service. The network operators deploy servers that are configured with policies and that collect routing information (e.g., OSPF/ISIS, BGP) and pro-actively measure the performance of popular destinations. The clients that need to select a path send requests to a server. A request contains a list of sources and destinations and a ranking criterion. The server replies with an ordered list of $\langle \text{source}, \text{destination}, \text{rank} \rangle$ tuples to the client. This ranking is based on the current network state and policies. The client then uses the first pairs of the list and potentially switches to the next one(s) in case of problem.

An IDIPS client refers to any entity that has the possibility to select paths to reach a destination or retrieve some content. *Peer-to-peer* applications are clear candidate for such an informed path selection service. In Peer-to-peer, IDIPS can be used to select the peers that minimize the download completion time without harming the ISP by using costly links. The IDIPS related work in such a context is P4P, Oracle and ALTO [8, 2, 7]. However, unlike these related works, IDIPS is not limited to the peers selection in P2P and can measure path performances. Consequently, IDIPS can, for example, be used by *dual stack hosts* to determine when IPv6 is faster than IPv4. Moreover, an increasing number of networks are *multihomed* [1] and an important benefit of multihoming as shown by Akella et al. is that multihoming allows sites to choose better quality paths over the Internet [4]. In this context, BGP can be coupled with IDIPS to dynamically use the most qualitative links to send traffic to or to attract traffic from, instead of relying on static policies.

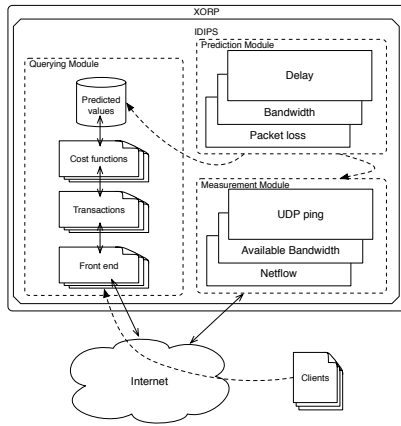


Figure 1: IDIPS architecture

2. IDIPS ARCHITECTURE

IDIPS is a service that can be used in many situations. This is why IDIPS presents a simple but powerful programming interface and is implemented in XORP.

As illustrated in Fig. 1 IDIPS is composed of three independent modules: the *Querying* module, the *Prediction* module, and the *Measurement* module. The Querying module is directly in relation with the client as it is in charge of receiving the requests, computing the path ranking based on the ranking criterion provided by the client and the ISP traffic engineering requirements, and replying with the ranked paths. The Measurement module is in charge of measuring path performance metrics if required. Finally, the Prediction module is used for predicting path performance (i.e., future performance metrics of a given path based on the past measurements).

The ranking criterion provided by clients in their requests might require measuring the network to obtain path performance metrics, such as delay or bandwidth estimation. The Measurement module pro-actively measures the most important destinations on behalf of the clients. Those measurements can be active (i.e., probes are sent in the network) or passive (i.e., no additional traffic is injected). The definition of important can be adapted to the needs (e.g., the destination that carry most of the traffic).

It is possible to predict the performance of a given path if it has been previously measured. This prediction task is achieved by the Prediction module. Note that a given measurement can be used in several different predictions. For instance, the previous delay measurements can serve for predicting the delay, the jitter, or for determining whether the path is reachable or not.

To enable flexibility, ease of implementation and performance, IDIPS clearly separates the Querying, Measurement and Prediction modules. Each module communicates with the other modules thanks to a standardized XRL interface. Therefore, the handling of requests from the clients is strictly separated from the prediction of path performance and path performance prediction is separated from path measurements.

The module interfaces are abstracted into XORP XRL interfaces such that anyone can add new measurement and prediction modules to IDIPS. The choice of implementing IDIPS in XORP makes the interaction between IDIPS, the control planes and the data plane easy.

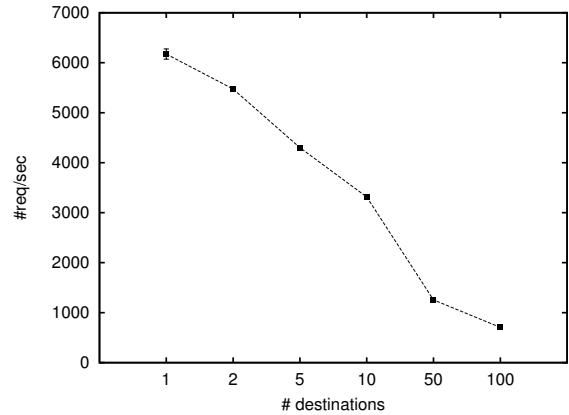


Figure 2: Load supported by the XORP implementation of IDIPS

Fig. 2 shows the achievable performance of our XORP implementation of IDIPS on a Intel Xeon E5430 at 2.66GHz processor with 4GB of RAM. To determine the achievable performance of our implementation, we repeated, for various size of random destination lists, ten times a run of 1,000,000 ranking requests. Fig 2 shows the mean (and its 95th confidence interval) rate supported by IDIPS. We can see that IDIPS can handle more than one thousands of requests per second for reasonable destination list sizes which should be sufficient to support a few tens of thousands clients. The degradation of performance with the size of the destination list is mostly due to the marshaling and unmarshaling events inside XORP. A detailed description and evaluation of IDIPS can be found in [6].¹

3. ACKNOWLEDGMENTS

This work is supported by the ECODE European Project.

4. REFERENCES

- [1] S. Agarwal, C.-N. Chuah, and R. H. Katz. OPCA: Robust interdomain policy routing and traffic control. In *Proc. IEEE OPENARCH*, April 2003.
- [2] V. Aggarwal, A. Feldmann, and C. Scheidele. Can ISPs and P2P users cooperate for improved performance. *ACM SIGCOMM CCR*, 37(3):29–40, July 2007.
- [3] Akamai. Akamai. <http://www.akamai.com>.
- [4] A. Akella, B. Maggs, S. Seshan, A. Shaikh, and R. Sitaraman. On the performance benefits of multihoming route control. *IEEE Transactions on Networking*, 16(1):96–104, February 2008.
- [5] Limelight Networks. Limelight. <http://www.limelightnetworks.com/>.
- [6] D. Saucez, B. Donnet, and O. Bonaventure. ISP-Driven Informed Path Selection. Technical report, 2011. <http://inl.info.ucl.ac.be/idips>.
- [7] J. Seedorf and E. Burger. Application-Layer Traffic Optimization (ALTO) Problem Statement. RFC 5693 (Informational), 2009.
- [8] H. Xie, Y. Yang, A. Krishnamurthy, Y. Liu, and A. Silberschatz. P4P: Provider portal for applications. In *Proc. ACM SIGCOMM*, August 2008.

¹IDIPS code on <http://inl.info.ucl.ac.be/idips>