Morpheus: Making Routing Programmable

Yi Wang
Princeton University
yiwang@cs.princeton.edu

Ioannis Avramopoulos
Princeton University
iavramop@cs.princeton.edu

Jennifer Rexford
Princeton University
jrex@cs.princeton.edu

ABSTRACT

This paper presents Morpheus, a modular, open routing platform that supports flexible control of routing policies of a network. With Morpheus, network operators can realize many useful policies that are infeasible today through composition of multiple policy modules and programming the route-selection algorithms. Morpheus can be readily deployed without requiring changes in other domains.

Categories and Subject Descriptors

C.2.2 [**Network Protocols**]: Routing Protocols; C.2.3 [**Network Operations**]: Network Management

General Terms

Routing, Policy, Control, Management, BGP

Keywords

Programmability, Flexibility, Modularity

1. INTRODUCTION

Interdomain routing policies play a critical role in many aspects of Internet Service Provider (ISP) backbone management, including business relationships with neighboring domains, end-to-end performance offered to customers, security of the network infrastructure and its customers, and scalability of the routing protocols [1]. Collectively, these *policy objectives* determine which routes the ISP uses, and which neighboring domains are permitted to use these routes. Routing policies determine the kinds of services ISPs offer to their customers, therefore have direct influence in their revenues. Although ISPs strive to provide new, value-added services, many desirable policies simply cannot be realized today [6]. We argue this lack of flexible support for policies is the result of limitations from both the intradomain routing architecture, and the software architecture of the Border Gateway Protocol (BGP) implementation in the routers.

INM'07, August 27–31, 2007, Kyoto, Japan.

Today, a BGP-speaking router selects and propagates a single best route for each prefix; the rest of candidate routes remain invisible to other routers of the same Autonomous System (AS). This restriction not only precludes an edge router from offering different routes to different customers, but also prohibits each router from making its own independent choice from the set of candidate routes.

In addition, the BGP standard [5] and its current implementations also impose restrictions on the set of policies that can be realized, for three main reasons:

Overloading of BGP attributes: Today, many different policy objectives are intertwined into a few BGP attributes (e.g., "local preference", used to enforce business relationships and perform traffic engineering). Overloading of attributes makes it difficult to incorporate new policy objectives without modifying the configuration of existing ones.

Difficulty in incorporating "side information": Policy objectives often depend on external information, like measurement data or business relationships with neighbor ASes. Satisfying policy objectives also sometimes requires updating state, such as a history of (prefix, origin AS) pairs or statistics about route instability over time. However, importing and updating state is very difficult today.

Restrictive step-by-step route-selection algorithm: The BGP route-selection algorithm selects the best route by considering one attribute at a time (e.g., first local-preference, then AS-path length, etc). This strict prioritization of BGP attributes limits ISPs to policies that rank one attribute over another, precluding policies that try to strike a balance between different policy objectives.

2. THE DESIGN OF MORPHEUS

2.1 Routing Architecture Supports

To support flexible policies, Morpheus changes the way routes are propagated and selected within an AS as follows.

Complete Visibility of BGP Routes: Morpheus achieves *full visibility* by directing all external BGP (eBGP) routes to a small collection of servers that make decisions on behalf of the routers, as shown in Figure 1 and inspired by the Routing Control Platform (RCP) [2]. Morpheus make decisions on behalf of the routers by assigning a best route for each prefix to every internal router individually.

Flexible Egress Selection Per Router: By using IP-in-IP tunnels or MPLS label-switched paths to direct traffic between edge routers (as shown in Figure 1), Morpheus can freely assign different BGP routes to different edge routers, without concern for inconsistent forwarding.

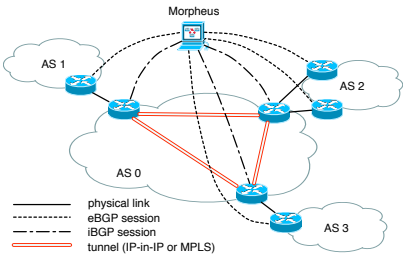

Figure 1: Morpheus servers select and propagate BGP routes within an AS

Multipath Routing and Forwarding: Morpheus is able to send multiple routes to each edge router [6], so that different customers could use different paths to reach the same destination. Such support for multipath routing and forwarding would allow ISPs and their customers to capitalize on these diverse paths for better performance and reliability, and even better security [7].

2.2 Programmable Software Architecture

Morpheus cleanly separates two operations that are intertwined today—classifying routes according to policy objectives and deciding how to weigh and make trade-offs between these objectives in picking the best routes. Policy classifiers are programmable modules that tag the routes, perhaps consulting or updating local state. Programmable route-selection algorithms allow network operators to make trade-offs between policy objectives in selecting the best routes.

Separation of Policy Objectives: Morpheus implements policy objectives as independent *import policy modules*, as shown in Figure 2. Each module is a classifier for a particular policy objective. The module receives as input a route and produces as output a tag that is affixed to the route. For example, a business-relationship module may tag a route as "customer", "peer", or "provider"; a stability module may tag a route with an number (e.g., an integer between 0 and 99), where a bigger number implies higher stability. Each import policy module has its own tag space, obviating the need to overload the same attributes. These tags are purely local to Morpheus, and are not disseminated to the routers; as such, using these tags does not require any changes to the routers.

Incorporating "Side Information": Many useful policy objectives require certain *side information*, including *external information* such as business relationships, measurement data, and registry of prefix ownership, and *internal states* such as a history of (prefix, origin AS) pairs. However, to incorporate side information today, network operators have to either "hack" their BGP configurations in an indirect and clumsy way (e.g., re-configuring filters and community attributes), or wait for software updates from router vendors. In Morpheus, each policy classifier can import external information and update internal state. For example, the business relationships module can access a table with the ISP's business relationships with neighboring ASes. A security module can access a registry of prefixes and their corresponding owners. A performance module can get periodic updates from a monitoring system. A route stability module can maintain statistics about route update frequencies. A route security module that implements Pretty Good BGP [4] can flag suspicious routes based on a history of (prefix, origin AS) pairs.

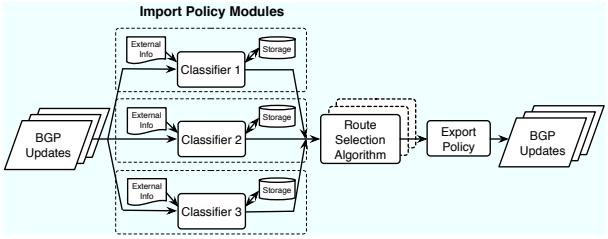

Figure 2: The modular architecture of Morpheus

Programmable Route Selection: The BGP route-selection algorithm applies a series of tie-breaking steps, one route attribute at a time. The ordering of the steps is built in to the routers and is relatively difficult to change, though some vendors enable operators to disable some steps. Imposing a strict priority on the attributes is especially restrictive, as it precludes policies that make trade-offs across different objectives. Morpheus allows operators to write their own route-selection algorithm that selects best routes based on the tags set by the import policy modules. However, many network operators would prefer not to write a "program" (e.g., in C or another higher-level language) every time they change their routing policies. In addition, expressing arbitrary route-selection algorithms in a higher-level language introduces a variety of risks (e.g., that the program never terminates). Instead, we envision that operators would use a simple configuration interface to control how Morpheus weighs the policy objectives. The underlying (configurable) algorithm should be flexible enough to support most useful policies, and efficient enough to terminate quickly.

Furthermore, in order to realize such policies that require multipath routing, Morpheus supports the parallel execution of multiple route-selection algorithms. Each algorithm can be configured to realize a different policy and select a potentially different best route for the same prefix. With this feature, an ISP running Morpheus can offer different types of routes to its customers as a revenue-generating service.

We implemented Morpheus as an extension to XORP [3]. More information can be found in [6].

3. REFERENCES

[1] M. Caesar and J. Rexford. BGP policies in ISP networks. *IEEE Network Magazine*, October 2005.

[2] N. Feamster, H. Balakrishnan, J. Rexford, A. Shaikh, and J. van der Merwe. The case for separating routing from routers. In *Proc. ACM SIGCOMM Workshop on Future Direction in Network Architecture*, August 2004.

[3] M. Handley, E. Kohler, A. Ghosh, O. Hodson, and P. Radoslavov. Designing extensible IP router software. In *Proc. Networked Systems Design and Implementation*, May 2005.

[4] J. Karlin, S. Forrest, and J. Rexford. Pretty good BGP: Improving BGP by cautiously adopting routes. In *Proc. International Conference on Network Protocols*, November 2006.

[5] Y. Rekhter, T. Li, and S. Hares. A border gateway protocol 4 (BGP-4). RFC 4271, January 2006.

[6] Y. Wang, I. Avramopoulos, and J. Rexford. Morpheus: Making routing programmable. Technical Report TR-784-07, Dept. of Computer Science, Princeton Univ., June 2007.

[7] D. Wendlandt, I. Avramopoulos, D. Andersen, and J. Rexford. Don't secure routing protocols, secure data delivery. In *Proc. ACM SIGCOMM Workshop on Hot Topics in Networking*, November 2006.