

Morpheus: Making Routing Programmable

Yi Wang, Ioannis Avramopoulos, Jennifer Rexford
Princeton University

{yiwang, iavramop, jrex}@cs.princeton.edu

Interdomain routing policies play a critical role in many aspects of Internet Service Provider (ISP) backbone management, including the business relationships with neighboring domains, the end-to-end performance offered to customers, the security of the network infrastructure and its customers, and the 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. 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 BGP implementation in the routers. We propose Morpheus, a modular, open routing platform that addresses these limitations by changing the way BGP routes are propagated and selected within an AS, and making the route-selection algorithm programmable. Morpheus can be readily deployed without requiring changes in other domains.

1 Limited Support for Flexible Policy

1.1 Limitations of the Routing Architecture

Today, an ISP expresses its policies by configuring the Border Gateway Protocol (BGP). The use of a single-path, path-vector routing protocol and hop-by-hop forwarding imposes several restrictions on how routes are selected *within a single AS* that fundamentally limit the policies an ISP can realize:

Propagating only one best route: Despite learning multiple routes for the same prefix, a BGP-speaking router only announces a single best route to its neighbors, making the rest of the candidate routes invisible to other routers. This restriction precludes each router from making its own independent choice from the set of candidate routes.

Selecting only one best route: Each router can only select one BGP route for forwarding data traffic. This not only limits the ability of routers to balance load over multiple paths, but also precludes an edge router from offering different routes to different customers.

Coupling of decisions across routers: Today, traffic entering the AS is forwarded to egress points in a hop-by-hop fashion. Edge routers connected to the same internal router are forced to direct traffic toward the same egress point.

1.2 Limitations of the BGP Protocol

In addition to the intrinsic limitations of the routing architecture within an AS, the current BGP standard [6] and its *de facto* implementations of BGP 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 from all candidate routes by considering one attribute at a time (e.g., first local-preference, then AS-path length, and so on). 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

Morpheus overcomes the three limitations of the routing architecture by changing the way routes are propagated and selected within an AS.

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 earlier work on the Routing Control Platform (RCP) [2]. Each server has a (multi-hop) eBGP session with each external neighbor router, in lieu of having direct eBGP sessions between the edge routers in the two ASes. Morpheus assigns a BGP route for each prefix to every internal router individually, using internal BGP (iBGP) sessions for backwards compatibility. Since the routers are no longer responsible for propagating BGP routing information to neighbor ASes, Morpheus does not need to send all of the route attributes—

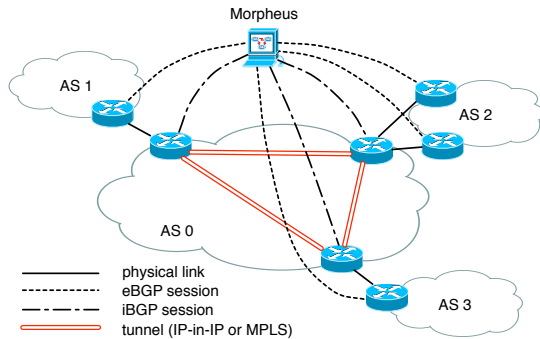


Figure 1: Morpheus server has BGP sessions with routers

only the destination prefix and next-hop address are strictly necessary. This enables a significant reduction in the amount of BGP information the routers must receive and store. Morpheus also ensures that the BGP routes propagated to eBGP neighbors are consistent with the route assigned to the associated edge routers, and include all route attributes expected by the neighbor ASes.

Flexible Egress Selection Per Router: Morpheus relies on IP-in-IP tunnels or MPLS label-switched paths to direct traffic between edge routers, as shown in Figure 1. This design choice offers several important advantages. First, Morpheus can freely assign different BGP routes to different edge routers, without concern for inconsistent forwarding. Second, Morpheus can rely on the IGP to determine how traffic flows between ingress and egress routers, reducing the complexity of Morpheus and ensuring fast reaction to internal topology changes. Third, Morpheus does not select BGP routes for the internal routers, reducing the total number of routers it has to manage. Fourth, tunneling also allows the ISP to configure the ranking of egress points for each ingress router to achieve traffic-engineering goals. These *stateless* tunneling technology is readily available at line rate in commercial routers supporting MPLS or IP-in-IP encapsulation, and a “BGP-free core” is increasingly common in large ISPs.

Multipath Routing and Forwarding: Today, customers connected to the same edge router are forced to use the same path to reach a destination. Support for multipath routing and forwarding (i.e., the ability of providing different paths to different customers to reach the same destination) would allow ISPs and their customers to capitalize on these diverse paths for better performance and reliability, and even better security [8]. It also ensures that each router has a backup path for faster failover (e.g., in case the next-hop address of the primary route becomes unreachable). With full visibility into the eBGP-learned routes and the internal topology, Morpheus can easily pick the best routes on behalf of every edge router and neighbor AS individually. Morpheus sends multiple routes to each edge router¹, and the “virtual rout-

¹This can be achieved by using the “route target” attributes commonly used with VRF in MPLS-VPN [5], or having multiple iBGP sessions between a Morpheus server and an edge router. Other options include using the BGP “add-paths” capability [7] or a new message dissemination protocol, which may be more efficient at the expense of backwards compatibility.

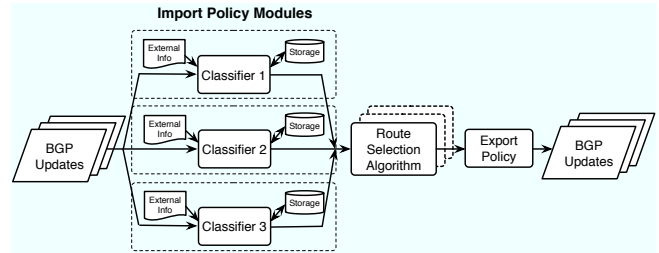


Figure 2: The modular architecture of Morpheus

ing and forwarding (VRF)” or “virtual router” features commonly available in commercial routers can be configured to allow different customers to use different paths [5].

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 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: Routing policies must balance multiple objectives such as business relationships, AS path length, and traffic engineering. Today, policy objectives are usually translated into specific values of BGP attributes to influence the route-selection algorithm. For example, different business relationships (customer, peer, provider) are typically represented using different local-preference (`local-pref`) values. Unfortunately, today’s BGP implementations do not provide an easy way to add new policy objectives into the system. As a result, multiple policy objectives must share a limited number of BGP attributes. For example, it is a common practice to also use `local-pref` to achieve traffic-engineering objectives. Overloading the attributes leads to complex, convoluted routing policies, where the network operators cannot easily reason about the policy objectives independently. The problem is exacerbated when operators need to add new policy objectives, further overloading the same attributes.

Morpheus addresses this issue by implementing policy objectives as independent *import policy modules*, as shown in Figure 2. Each module works as 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 (evaluating the stability of routes based, for example, on a route-flap damping algorithm) 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. By tagging the routes, rather than filtering or sup-

pressing them, the route-selection algorithm is guaranteed to have full visibility of the candidate routes. 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 BGP protocol implemented on 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, there is no systematic mechanism to incorporate side information into routers 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 (if the need for certain side information becomes compelling) and then upgrade a large number of routers.

In Morpheus, each policy classifier can import external information and update internal state. For example, the business relationships module can have access to up-to-date information about the ISP’s business relationships with neighboring ASes through a configuration file, or a database. A security module can have access to 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 announcement/withdrawal frequencies. A route security module that implements Pretty Good BGP [4]—a simple algorithm that can effectively detect BGP prefix and subprefix hijacks—can keep past history of (prefix, origin AS) pairs in the past 24 hours.

Programmable Route Selection: Today’s 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 policy objectives.

Morpheus allows network 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 want to 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 prefix². 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 at: <http://www.cs.princeton.edu/~yiwang/papers/morpheus.pdf>

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] I. Pepelnjak and J. Guichard. *MPLS and VPN Architectures*. Cisco Press, 2000.
- [6] Y. Rekhter, T. Li, and S. Hares. A border gateway protocol 4 (BGP-4). RFC 4271, January 2006.
- [7] D. Walton, A. Retana, and E. Chen. Advertisement of multiple paths in BGP. Internet Draft draft-walton-bgp-add-paths-05, March 2006.
- [8] 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.

²Alternatively, a route-selection algorithm can return a set of best routes (e.g., top k routes) instead of one, if desired.