The Case for Separating Routing from Routers

Nick Feamster, Hari Balakrishnan
MIT Computer Science & AI Lab
{feamster,hari}@csail.mit.edu

Jennifer Rexford, Aman Shaikh, Jacobus van der Merwe
AT&T Labs–Research
{jrex,ashaikh,kobus}@research.att.com

ABSTRACT

Over the past decade, the complexity of the Internet's routing infrastructure has increased dramatically. This complexity and the problems it causes stem not just from various new demands made of the routing infrastructure, but also from fundamental limitations in the ability of today's distributed infrastructure to scalably cope with new requirements.

The limitations in today's routing system arise in large part from the fully distributed path-selection computation that the IP routers in an autonomous system (AS) must perform. To overcome this weakness, interdomain routing should be separated from today's IP routers, which should simply forward packets (for the most part). Instead, a separate *Routing Control Platform (RCP)* should select routes on behalf of the IP routers in each AS and exchange reachability information with other domains.

Our position is that an approach like RCP is a good way of coping with complexity while being responsive to new demands and can lead to a routing system that is substantially easier to manage than today. We present a design overview of RCP based on three architectural principles—path computation based on a consistent view of network state, controlled interactions between routing protocol layers, and expressive specification of routing policies—and discuss the architectural strengths and weaknesses of our proposal.

Categories and Subject Descriptors

C.2.2 [**Network Protocols**]: Routing Protocols; C.2.6 [**Computer-Communication Networks**]: Internetworking

General Terms

Algorithms, Design, Management, Performance, Reliability

Keywords

routing architecture, interdomain routing, BGP

1. Introduction

This paper posits that interdomain routing protocol functionality should be separated from the routers. Stated somewhat glibly, routing is too important and too complicated to be left to today's routers! IP "routers" should be "lookup-and-forward" switches, forwarding packets as rapidly as possible without being concerned about path selection. A separate entity should be responsible for computing the best BGP[1] paths on behalf of all the routers in a domain and disseminating the results to the routers.

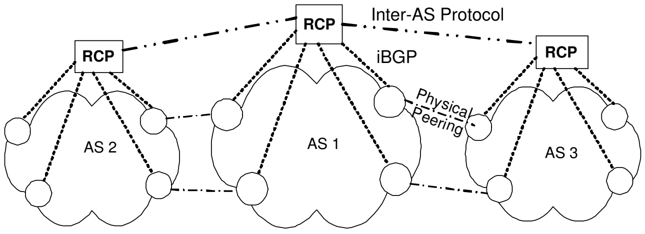

Figure 1: A Routing Control Platform (RCP) for the Internet. Circles represent conventional routers.

Separating interdomain routing from the individual routers is one way to cope with the increasing complexity of the routing system. The growth of the Internet has introduced considerable complexity into interdomain routing, as features have been added to BGP to support more flexibility (*e.g.*, new route attributes such as communities and MED) and larger scale (*e.g.*, route reflectors and route aggregation). This complexity has made routing protocol behavior increasingly unpredictable and error prone [12]. Requiring the routers to perform complex path computation introduces the potential for inconsistencies across routers, complicates the expression of routing policy, and makes troubleshooting difficult.

Instead, a separate *Routing Control Platform (RCP)* should have the information needed to select routes for each router in a domain (*e.g.*, an AS) and exchange routing information with RCPs in other domains.[2] Figure 1 illustrates this idea. Each RCP could use a new way of selecting routes for each router (rather than using today's unwieldy BGP decision process); RCPs could even exchange routes using an interdomain routing protocol other than BGP. By selecting routes on behalf of *all* routers in a domain, RCP can avoid many internal BGP-related complications (*e.g.*, forwarding loops [9] and signaling partitions [12]). This approach also facilitates traffic engineering, simpler and less error-prone policy expression, more powerful diagnosis and troubleshooting, more rapid deployment of protocol modifications and features, enforceable consistency of routes, and verifiable correctness properties. In contrast to previous approaches for centralizing interdomain routes and policies at route servers [19], RCP also preserves the autonomy of each AS for selecting paths and applying policies.[3]

SIGCOMM'04 Workshops, Aug. 30-Sept. 3, 2004, Portland, Oregon, USA.
Copyright 2004 ACM 1-58113-942-X/04/0008 ...$5.00.

[1] The Border Gateway Protocol (BGP) [1] is the *de facto* standard interdomain routing protocol.

[2] In this paper, we use the term "RCP" to refer to both the architecture as a whole and to the specific instance of RCP within a routing domain.

[3] RCP more closely resembles the Network Control Point (NCP), introduced in the telephone network in the early 1980s to simplify network management and support the rapid introduction of new features (*e.g.*, enhanced 1-800 service) [24, 27].

RCP's deployment path is as interesting as the envisioned end state. The deployment of RCP can proceed in three stages, offering the following benefits to network operators as RCP becomes more widely deployed:

1. **Control over protocol interactions:** RCP customizes the distribution of BGP routes within an AS by replacing internal BGP route reflectors. This stage does not require cooperation from neighboring domains. Because RCP has a complete view of the intra-AS topology and selects routes on behalf of all routers in the domain, it can prevent internal BGP routing anomalies and control traffic flow more directly.
2. **Network-wide path selection and policy:** By establishing BGP sessions directly with the routers in neighboring ASes, RCP can perform all routing decisions for an AS, bypassing the BGP decision process on the routers. This approach simplifies configuration and allows an AS to select routes based on high-level goals, rather than obscure manipulation of BGP route attributes.
3. **Redefinition of inter-AS routing:** Using RCPs, rather than routers, to exchange routes between ASes (as shown in Figure 1) enables the design of a new routing protocol because interdomain routing is now separated from IP routers. For example, RCP can be used to implement a control overlay that selects paths based on prices or performance statistics.

In addition to providing substantial improvements over today's routing architecture, RCP has a compelling deployment incentive (*i.e.*, a "tipping point"), so that an individual AS could deploy RCP and still realize significant benefits. Because the first two stages of deployment substantially reduce management complexity for BGP routing *within a single AS*, network operators have a compelling incentive to deploy RCP regardless of whether other ASes do so. Managing routing configuration requires constant vigilance from network operators. Although network management systems can often automate the most frequent tasks, working around and within the constraints of the existing routing protocols makes these systems much more complicated than necessary. Additionally, the complexity of modeling and managing the distributed configuration state in today's routers has itself impeded the evolution of automated management systems. In addition, because it communicates routes to each router in the AS using BGP, RCP is backwards compatible with existing routers; deploying RCP requires no changes to router hardware and software, only to router *configuration*.

The rest of the paper proceeds as follows. Section 2 presents background on today's interdomain routing infrastructure. In Section 3, we propose three architectural principles and explain how the existing routing infrastructure fails to meet them. Building on these insights, Section 4 describes the RCP architecture in detail, focusing on how each stage of deployment simplifies router configuration and management. In Section 5, we discuss the risks and challenges of having the RCP in the critical path of IP routing decisions. Section 6 reviews related work, and Section 7 concludes.

2. BGP Routing in an Autonomous System

An AS uses external BGP (eBGP) to exchange reachability information with neighboring domains and internal BGP (iBGP) to distribute routes inside the AS, as shown in Figure 2. Each router invokes the BGP decision process to select a single "best" route for each destination prefix from the candidate routes learned from eBGP and iBGP. The router combines the best BGP route with information about the internal network topology from the Interior Gateway Protocol (IGP) to construct a forwarding table that maps destination prefixes to outgoing links. Most of the flexibility and complexity of BGP routing comes from the following three areas:

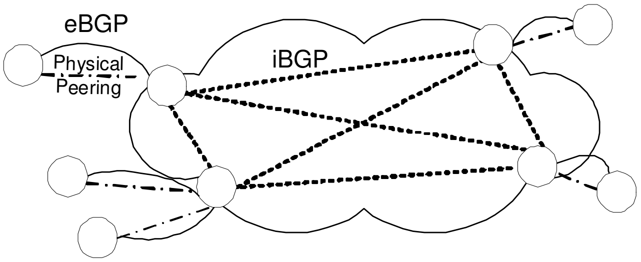

Figure 2: Operation of BGP routing inside an AS. Most small networks use a "full mesh" iBGP configuration, where every router in the AS has an iBGP session to every other router.

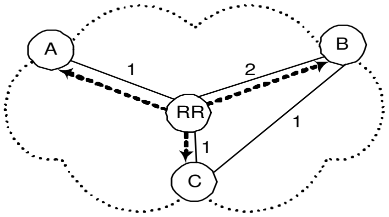

Figure 3: An example of where iBGP with route reflection does not emulate full-mesh iBGP; numbers represent IGP path costs, and arrows indicate an iBGP session from a route reflector to its client. In a full-mesh, router C would prefer routes learned from B over routes learned from A because its IGP path cost to B is smaller. However, in the example shown, RR prefers A, and, thus, C must also select A.

Path selection: A route to a destination prefix includes attributes such as the AS path, local preference, origin type, and multi-exit discriminator (MED). Each router applies a decision process [1] that consists of a sequence of rules that ranks the routes. After preferring routes with highest local preference, smallest AS path length, lowest origin type, and smallest MED, the decision process favors eBGP-learned routes over iBGP-learned routes. If multiple equally-good routes remain, the router favors the BGP route learned from the nearest border router—the *egress point* with the smallest IGP path cost—following the common practice of "hot-potato" routing. The final tiebreak is vendor-dependent and may depend on the age of the routes or an arbitrary router ID.

Intra-AS route distribution: Network operators can propagate eBGP-learned routes throughout an AS in many different ways.[4] Small networks typically have a "full mesh" of iBGP sessions, as shown in Figure 2. To avoid the n^2 scaling problem, a large AS may have a more complex iBGP topology. For example, although a router does not normally forward iBGP-learned routes to its other iBGP neighbors, it can be configured as a *route reflector*, which forwards routes learned from one route-reflector client to another. A router forwards only its *best* route to its iBGP neighbors, making the choices available at one router depend on decisions made by its iBGP neighbors, as shown in Figure 3.

Routing policy: Network operators influence path selection by configuring import and export policies on the eBGP sessions to neighboring domains. An *import* policy filters unwanted routes and

[4] In most IP backbone networks, every router needs to receive BGP routing information to construct a complete forwarding table. In a Multi-Protocol Label Switching (MPLS) network, only the border routers need to send and receive the BGP routes; the internal routers would simply forward packets on label-switched paths from the ingress router to the egress point.

manipulates the attributes of the remaining routes; for example, the policy could assign a small local preference to routes learned from one neighbor to make these routes less attractive than routes learned from other neighbors. After selecting a single best route, the router applies an *export* policy to manipulate the attributes and decide whether to propagate the route to a neighbor. For example, a router may be configured to export routes learned from a private peer to a customer but not to another private peer.

3. Architectural Principles for Routing

In this section, we present three architectural principles for reducing interdomain routing complexity:

1. The routing architecture must base its routing assignments on a consistent view of routing state.
2. The interfaces between the routing protocols must minimize unexpected or unwanted interactions.
3. The interdomain routing mechanisms must directly support flexible, expressive policies.

Each subsection in this section discusses one of these principles. For each principle, we present a high-level rationale, followed by specific examples of how today's interdomain routing architecture violates the principle. For each of these examples, we suggest how adhering to the architectural principle helps solve the problem.

3.1 Compute Routes Using Consistent State

Routing state and logic should be co-located with the system components that are assigning routes. The logical participants in an interdomain routing protocol are the *ASes*, not the individual routers. The interdomain routing architecture should view each AS as a single participant and base routing decisions on a network-wide view of available routes and configuration state; the routers, on the other hand, should forward data traffic without concern about how the routes are computed. The current interdomain routing system violates this architectural principle in the following three ways:

Decomposing the routing configuration state across the routers unnecessarily complicates policy expression. Although distributing state to achieve scalability and reliability makes sense, many aspects of configuration are not replicated, but rather *decomposed* across routers. Configuration state should be logically centralized because it simplifies policy expression without compromising scalability or reliability.

Problem: Network operators must often implement high-level policies, such as preventing routes learned from one AS from being advertised to another. Implementing this policy currently requires modifying the configurations of multiple routers: the import policies must "tag" eBGP-learned routes appropriately, and the export policies of other routers must filter routes with this tag when advertising to eBGP neighbors.

Solution: Defining routing policy on a network-wide basis would obviate the need for this level of indirection. A network-wide configuration management entity could know the origin of all routes based on the eBGP sessions that advertised them, which would allow a direct expression of policies based on sessions.

Distributed path selection causes routing decisions at one router to depend on the configuration of other routers. Subtle configuration details affect the route that a router selects or whether that router learns a route at all. Computing routes on a network-wide basis using a consistent view of routing state can reduce interdomain routing's dependencies on these subtle details.

Problem: Omitting a single iBGP session in a full-mesh configuration can leave a router with no route for certain destinations, even if the intradomain topology is connected. Distributed path selection also makes predicting the effects of configuration changes on traffic flow difficult [15].

Solution: An entity that performs path assignment on behalf of all routers could control path assignment to ensure that every router is assigned a route for every destination.

Each router is unaware of the state at other routers; this lack of information may result in incorrect or suboptimal routing.. Implementing BGP's many features on the routers makes these features difficult to reason about. For example, replication of functionality that is intended to improve reliability can cause forwarding loops, and a feature intended to prevent routing instability can slow convergence. A routing architecture should implement these features in a module that has a complete view of the network state, rather than in the routers (each of which only has a partial view of network state); doing so would allow that module to ensure sensible, consistent network-wide route assignment and override any feature interactions that cause incorrect routing.

Problems: A router typically has iBGP sessions to multiple route reflectors to improve reliability. When a route reflector fails, protocol oscillation and forwarding loops can arise if the second route reflector has a different view of the best routes. Placing the two route reflectors close to each other reduces these kinds of inconsistencies but introduces fate sharing (*i.e.*, the risk of shared failures). As another example, BGP route flap damping suppresses unstable routes that change frequently [40]. Unfortunately, sometimes a single failure can trigger many advertisements that can mistakenly activate route flap damping [29]. Network operators must work backwards to select configuration parameters that prevent erroneous damping.

Solution: An entity that performs route computation using a consistent view of available routes and network topology can be replicated using standard distributed systems algorithms. Unlike route reflectors, each replica would assign the same route to each router, independently of its location in the network. A module with knowledge of the routes assigned to every router in the AS could also detect when route changes are caused by path exploration and avoid unnecessarily suppressing a route.

3.2 Control Routing Protocol Interaction

Dividing functionality into distinct modules with clear interfaces can control complexity. In the routing system, the *IGP* computes paths between routers in an AS, *eBGP* computes paths between ASes, and *iBGP* propagates eBGP-learned routes throughout an AS. At a higher layer, *overlay networks* route traffic along one or more end-host hops, abstracting the IP substrate entirely. Unfortunately, the modules in today's interdomain routing system interact in the following undesirable ways:

Hard-wired interactions between eBGP and the IGP constrain an operator's control over path selection. Although the internal topology should have some influence on BGP routing decisions (*e.g.*, it allows nearest-exit routing), a router's choice of egress point should be relatively insensitive to small IGP changes.

Problem: The BGP decision process uses the IGP path cost to break the tie between two "equally good" routes. Internal events, such as link failures, planned maintenance, or traffic engineering often lead to changes in the IGP path costs. These IGP changes can cause a router to change its best *BGP* route, causing abrupt, unwanted traffic shifts [39]. Additionally, an operator may sometimes *want* to redirect traffic from one egress link to another. Today, this requires complex manipulation of the BGP import policies to make some egress points less attractive than others [13].

Solution: With better control over the interactions between eBGP and IGP, an operator could directly assign new routes to some routers without changing BGP routing policies.

Inconsistencies between iBGP and IGP can cause forwarding loops and route oscillation. Operators can test that their iBGP configuration satisfies sufficient conditions for correctness [21], but this approach is not robust because operators commonly misconfigure iBGP [12]. The routing architecture should explicitly *enforce* correctness constraints.

Problem: An iBGP route reflector selects and distributes one best BGP route for each destination prefix. As a result, the route-reflector clients do not necessarily make the same BGP routing decisions as they would in a full-mesh iBGP configuration. In particular, a route reflector and its clients may have different IGP path costs to the egress routers, leading to different BGP routing decisions, as shown previously in Figure 3. These inconsistencies can lead to protocol oscillations or persistent forwarding loops [5, 21, 31] if a router forwards a packet toward one egress point via a router that has selected a BGP route with a *different* egress point. These "deflections" can also cause the AS-level forwarding path to differ from the BGP AS path, which can complicate debugging [30].

Solution: Rather than being agnostic about IGP forwarding paths, the routing architecture could use the available knowledge to explicitly enforce consistency in router-level forwarding paths.

Interactions between overlay networks and the underlying network can degrade performance. Overlay networks measure end-to-end path performance and tune routing at the edge of the network, but they typically lack (1) detailed *measurements* of traffic and routing that would help them make better decisions and (2) direct *control* over IP-layer protocols and mechanisms. The routing architecture should provide the information and control that overlays need via a well-defined interface.

Problem: Route control products [34, 36] help multihomed ISPs select upstream routes for each destination, whereas end-host overlays such as RON [4] circumvent failures and congestion by directing traffic through an intermediate host. Because they lack complete information about routing and traffic-engineering optimizations, these overlays sometimes increase congestion and decrease the effectiveness of traffic engineering in the underlay network [33], which can degrade user performance.

Solution: With more direct control, overlays could operate more efficiently (*e.g.*, by not sending the same traffic over congested links at the network edge [25]). With more information about routing dynamics, overlays could pre-emptively avoid some outages [10].

3.3 Support Flexible, Expressive Policies

The interdomain routing architecture must support flexible, expressive policy. The need for greater flexibility in selecting and exporting routes has driven many of the extensions to BGP over the past fifteen years, and we believe this trend is likely to continue. Although BGP is highly configurable, its operation is controlled by *indirect* mechanisms that expose details rather than abstracting them. Architectural simplifications and better abstractions can simplify configuration languages and make policy specification simpler and more expressive. The following points illustrate why today's routing architecture does not satisfy these goals:

BGP's mechanisms preclude the expression of certain policies and make others difficult to express. Network operators influence the outcome of the BGP decision process by configuring policies that modify the attributes of BGP routes. Better configuration languages would be helpful, but the architecture should also provide more flexible support for assigning paths to routers.

Problem: Moving traffic from one inter-AS link to another requires [13]: (1) identifying the subset of prefixes that carries the desired amount of traffic, (2) determining how to express that subset (*e.g.*, by a common AS path regular expression), (3) modifying the import policies on one or more routers to assign a smaller "local preference" for routes matching those expressions, and (4) observing the resulting traffic flow and iterating as necessary.

Solution: Although "what if" tools can help predict the effects of policy changes [15], the routing architecture should allow an operator to move traffic by *explicitly* assigning paths.

BGP's mechanisms impede multiple ASes from cooperating in selecting routes that satisfy their goals. ASes must cooperate to ensure end-to-end reachability, but today's routing architecture does not directly support this type of cooperation. Interdomain routing policies are a tussle space [7]: an AS must balance the dependence on its neighbors for good connectivity to the rest of the Internet and competition with neighbors for customers and revenue. Operators must currently resolve these conflicts outside of the infrastructure, but the architecture should directly support route selection based on negotiated preferences or financial incentives.

Problem: Suppose one AS wants to advertise a backup route to its neighbor. These two ASes must first negotiate a backup "signal" out of band. The AS advertising the route must then modify its export policies to attach this signal to the backup route, and the neighbor must modify the import policies on its routers to lower the "local preference" value for routes with this community.

Solution: Because route negotiation is fundamental to inter-AS cooperation, the interdomain routing should support it directly.

4. Routing Control Platform (RCP)

Building on the principles from Section 3, this section proposes a *Routing Control Platform* (RCP), which separates the control-plane logic from the routers that forward packets. We describe RCP as a single, logically-centralized entity in each domain. This centralized function must actually be implemented in a reliable, physically distributed fashion to avoid introducing a single point of failure and ensuring robust route distribution. We believe that existing distributed systems techniques may be applicable; this paper does not address this issue in detail, but we briefly discuss it in Section 5.

We describe RCP in terms of three phases: (1) controlling routing protocol interactions by replacing iBGP route reflection with RCP, (2) gaining flexibility over route selection by making RCP the endpoint of all eBGP sessions with neighboring ASes, and (3) enabling changes to interdomain routing by using RCPs, rather than routers, to exchange routes between ASes using eBGP or some new protocol. By describing RCP in terms of three stages, we demonstrate that RCP is incrementally deployable *within an AS* and, more importantly, provides significant benefits to an individual AS even if other ASes have not deployed RCP. In addition to being steps of incremental deployment, each phase provides new functionality while remaining backwards compatible with BGP.

4.1 Control Over Protocol Interactions

The first phase of RCP deployment, shown in Figure 4, involves only minor changes to the iBGP *configuration* inside an AS. First,

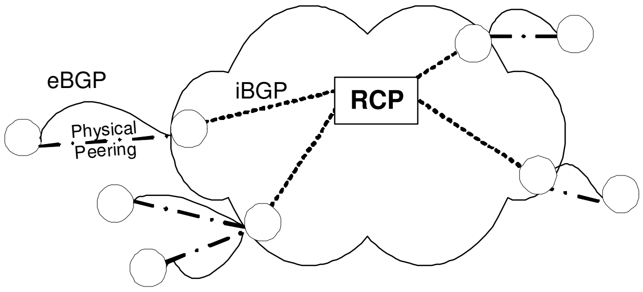

Figure 4: The first phase replaces the pairwise iBGP sessions between routers with iBGP sessions to RCP. RCP uses knowledge about the IGP topology and the best routes from each border router to make routing decisions on behalf of each router. RCP distributes the path assignment to the routers via iBGP.

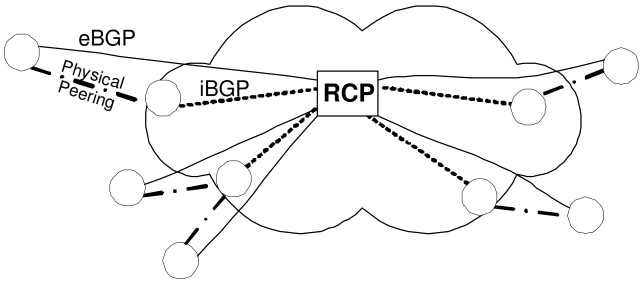

Figure 5: The second deployment phase of RCP operates in a similar manner as the first phase, but now RCP itself has eBGP sessions to routers in other ASes, rather than relying on border routers to learn routes from other ASes and apply local policies.

RCP monitors the IGP to maintain an accurate, up-to-date view of the IGP topology; previous work explains how to monitor an IGP without disrupting the operation of the network [35]. Next, instead of having routers propagate eBGP-learned routes through an iBGP hierarchy, a router sends its best route for each eBGP-learned destination to RCP via an iBGP session. Finally, RCP computes a route for each router and conveys that route via the iBGP session. Using an RCP does not require any changes to the routers themselves (aside from the configuration of iBGP sessions to RCP) or the configuration of routers in other ASes. Many ISPs already deploy a monitoring infrastructure to keep track of network state and routing protocol behavior. At this stage, RCP is essentially an IGP and BGP monitoring infrastructure that also *controls* route selection.

This stage of RCP closely resembles an architecture based on route reflection [6], but, unlike route reflectors, RCP can return a *different* best route to each router. For example, RCP could compute the route that each router would have selected in a full-mesh iBGP topology. RCP also offers more flexibility than route reflection because it is not limited to emulating a full-mesh iBGP scenario: RCP could intentionally select other routes to control the interactions between iBGP and the IGP. RCP may also appear similar to previous work on route servers [23] that forward *all* BGP-learned routes to their clients, but, because it forwards only *one* route to each client, RCP remains backwards compatible with BGP and enables customized path selection.

In the rest of this subsection, we present several examples to show how this stage of RCP deployment simplifies important network management tasks.

Enforceable correctness constraints and invariants. With complete knowledge of the iBGP and IGP topologies, RCP can enforce a clean separation of routing layers. For example, RCP can ensure that each router along a forwarding path selects the same best BGP route for a destination prefix, which prevents the forwarding loops and protocol oscillations that can arise in conventional iBGP configurations [9, 21]. RCP can also be useful for detecting persistent oscillations caused by the MED attribute [20], which occurs because routers do not have a total ordering over the set of candidate routes. With a complete view of the best routes from each border router, RCP can recognize when each router would not have a single, consistent ordering and can force the system into a stable path assignment.

Avoiding unintentional hot-potato routing changes. Small changes in the IGP topology (*e.g.*, due to traffic engineering, failures, or planned maintenance) can trigger large, unnecessary shifts in eBGP routes because of BGP's "hot potato" routing behavior [39]. RCP can allow a network operator to add or remove internal links, or modify the IGP costs, without worrying about the side effects on BGP path selection. By controlling the path selection for each router, RCP can force routers to continue using an egress point even when a link failure or small IGP cost change makes another egress point become slightly "closer". (RCP must take care to ensure that each router along the forwarding path to the egress point for that router continues to pick the same egress point.) In addition to avoiding unnecessary traffic shifts, preventing these abrupt routing changes improves global routing stability by reducing the number of eBGP routing changes propagated to downstream neighbors.

More flexible traffic engineering. RCP can *intentionally* change the egress point for a router to move traffic to a lightly-loaded edge link or a less congested downstream path. This approach allows the AS to balance the traffic load without any changes to the import policies on eBGP sessions or the IGP link costs. In addition to controlling the egress point, RCP could dictate the entire forwarding path through the AS rather than relying on the IGP. For example, RCP could send a router a BGP route with a "next hop" that corresponds to an immediate neighbor in the IGP topology, which would cause that router to create a forwarding table entry that maps the destination prefix to the outgoing link connecting directly to the neighboring router. This kind of fine-grained control is useful for planned routing changes. RCP could make these routing changes incrementally (*i.e.*, one router at a time) to avoid creating transient forwarding loops during convergence.

4.2 Network-Wide Path Selection and Policy

In the first deployment phase, the AS's border routers continue to exchange routes with neighboring domains. The AS's border routers still apply local import and export policies and forward a single best route for each prefix to RCP. In the second stage, RCP exchanges routes directly with the border routers in other ASes, as shown in Figure 5. Neighboring ASes must modify the configuration of their eBGP sessions to peer with RCP, rather than with individual routers.[5] This change only involves changes to the router *configuration*, not the underlying hardware or software, and it offers significant benefits because (1) RCP has access to *all* routes learned via eBGP from other ASes, (2) all routing policies for the AS are applied directly at the RCP, and (3) the border routers do not need any BGP configuration beyond their iBGP session to RCP. This phase of RCP significantly simplifies network management.

Simpler routing configuration. With all of the eBGP-learned routes in one place, the configuration of the routing policies can re-

[5] Each router on the path between RCP and routers must also have routes for both endpoints. These routes can be established by injecting routes for the endpoints into the routing protocol or by configuring static routes.

side entirely at the RCP. Rather than using BGP communities to tag routes at one router to ensure the correct handling at another router, RCP can classify and select the routes itself. For example, suppose eBGP routes learned from one peer should not be advertised to another. RCP could maintain a local registry of peer and customer AS numbers and ensure that routes where the neighboring AS is a peer AS are not advertised via eBGP sessions to other peer ASes. In today's routing infrastructure, the auxiliary information about peer and customer ASes would be expressed indirectly (*i.e.*, in the import policies that tag the routes learned on certain eBGP sessions and export policies that filter routes based on these tags). With the RCP performing all routing decisions, this type of decomposition is unnecessary.

Network-wide traffic engineering. In the second phase, RCP has access to *all* of the eBGP-learned routes, not just the best paths selected by the border routers. With complete control over the selection of paths, RCP can disregard the unwieldy BGP decision process. RCP can influence the routing decisions of various routers directly, without meddling with local preference settings at individual routers. Rather than generating complex import policy rules that manipulate the local preference attribute, RCP could explicitly decide which path each router should select for any destination prefix. In addition to comparing the eBGP-learned routes, RCP could base routing decisions on auxiliary information such as measured traffic volumes, performance statistics (*e.g.*, observed packet loss), and commercial relationships with neighboring domains (*e.g.*, the pricing model).

Intelligent route-flap damping. Many BGP update sequences are caused by routers performing "path exploration": upon learning of a route's withdrawal from a neighboring AS, a router will readvertise its second best path until it receives the corresponding withdrawal for that path, and so on. RCP can prevent route-flap damping from discarding an otherwise stable route. Rather than having routers implement route-flap damping independently, RCP could damp routes on behalf of routers in the AS based on a network-wide view of the eBGP-learned routes. Additionally, RCP could determine when advertisements appear to stem from path exploration and use this information to delay readvertisement, thus preventing routers in neighboring ASes from receiving a flurry of transient advertisements during path exploration.

Coalescing routing table entries with customized aggregation. Networks often advertise multiple subnets in the same address block to balance the flow of traffic over several incoming links, which can lead to large routing tables and a larger number of BGP update messages. An individual router cannot typically safely aggregate subnets with the same next-hop, because another router in the AS may need to treat the subnets differently. As such, operators are often conservative in aggregating routes to prevent unintentional blackholes and forwarding loops. Giving RCP control over which BGP routes are sent to each router permits more aggressive aggregation. For example, if RCP discovers that the BGP routes for 12.1.2.0/24 and 12.1.3.0/24 at some router will use the same outgoing interface, it can send a single 12.1.2.0/23 route to the router, which can substantially reduce the memory requirements for the routing and forwarding tables.[6] (Note that RCP can send an aggregated route to a router even if the two initial routes have *different* AS paths, since the individual routers no longer act on this information.) This technique can also reduce the number of BGP updates, since many BGP routing changes affect attributes such as AS path, community, and MED that do not affect forwarding.

4.3 Redefinition of Inter-AS Routing

In the third phase, multiple ASes with RCPs can exchange interdomain routing information directly through their RCPs, as previously shown in Figure 1. As in the first two phases, RCP makes routing decisions on behalf of the routers in its AS. RCP could simply use eBGP to exchange routing information, but exchanging routes with eBGP is not strictly necessary. RCP could also enable ASes to better coordinate when diagnosing routing problems and selecting paths.

Better network diagnostics and troubleshooting. RCP could provide diagnostic information to a neighboring AS (or even remote ASes) upon request. Network operators regularly send email to mailing lists (*e.g.*, NANOG) to ask other operators about possible reachability problems and diagnose problems as they arise. With RCPs deployed in many ASes, the collection of RCPs could be treated as a distributed database of routing information, where each AS maintains a portion of and provides a query interface to that information [38]. An AS could allow other ASes to query the routes that it has learned from other ASes for debugging (*i.e.*, using RCP query interface as a sort of master "looking glass" server for the entire AS) or verification [11] (*e.g.*, verifying an AS path by asking other ASes along that path if they have learned corresponding route). Of course, the diagnostic information need not be limited to BGP data. For example, RCP could maintain information about intra-AS topology changes, link congestion, and performance statistics to help explain disruptions in end-to-end performance.

New interdomain routing protocols. RCP enables a variety of proposals for fundamental changes to interdomain routing. Recent proposals have advocated modifying the way the interdomain routing protocol selects and propagates routes. For example, a new routing protocol could attach prices to advertised routes [16] or explicitly support inter-AS negotiation to select the routes [28]. RCPs could also base their routing decisions on measured end-to-end performance, as proposed in work on overlay networks [4] and even make this performance information available to end-host overlays through appropriate interfaces [32]. Other proposals have suggested ways to improve security by performing path authentication [37] or origin authentication [3]. Until now, many of these proposals have had no feasible deployment path because they require fundamental protocol changes and would not be backwards compatible with the installed base of routers. RCP allows the deployment of new routing protocol changes without modifying or replacing the existing infrastructure.

5. Challenges Introduced by RCP

Separating routing state from the routers can potentially introduce robustness, scalability, speed, and consistency problems. The RCP architecture must address these challenges to be viable. In this section, we briefly highlight these issues and sketch possible solutions to these problems. We are addressing these problems in greater detail in our current work, and we are implementing a prototype of RCP using OSPF and BGP data from AT&T's domestic IP backbone [14].

It might seem that moving complexity out of the routers into RCP creates new problems because of the additional flexibility in path assignment and because we are adding a component to the routing system. However, management systems and verification tools for BGP configuration already exist today, but they are more complicated and constrained because they must work around the artifacts

[6] An individual router can coalesce subnets when constructing its local *forwarding* table [8], but this approach does not reduce the size of the BGP routing table or the number of BGP update messages.

of today's routing system. Thus, adding RCP to the routing system does not really constitute "more functionality"; rather, RCP moves routing functionality to a part of the system where complexity can be better managed.

Robustness. To avoid introducing a single point of failure, RCP should be distributed across multiple *RCP servers* (RCSes). These servers must maintain a consistent view of the available routes to ensure that all routers receive consistent, loop-free paths. The RCSes must employ a protocol that recognizes when an AS becomes partitioned and guarantees that each partition receives routing information that is consistent within its partition. We are currently studying the types of inconsistencies that can result from various combinations of partitions. Our preliminary results suggest that even if a network is partitioned, RCSes in separate partitions cannot create a forwarding loop. This result follows from the fact that network partitions are caused by partitions of the IGP (*e.g.*, OSPF) topology, and RCSes rely on the IGP to exchange routes with each other and with BGP routers. Thus, a protocol that elects an RCS for each partition guarantees correct, loop-free forwarding.

Scalability. RCP must be able to handle thousands of eBGP sessions and hundreds of iBGP sessions, each with thousands of routes. Today's high-end desktop machines satisfy the memory and computational requirements for RCP. In our current work, we are exploring ways to distribute RCP functionality across many physical machines. One design idea we are currently pursuing involves dividing the RCP into a *BGP engine*, which is responsible for establishing the (possibly large number of) BGP sessions to routers within the AS (and, ultimately, across ASes) and whose sole responsibility is state management; and an *RCP engine*, which receives the routing information from the machines running BGP engines and implements the logic that we have discussed in this paper (*e.g.*, path computation, configuration management, maintaining consistency, etc.).

Convergence speed. RCP must compute routes using BGP and IGP information for every router in the AS and propagate the results of this computation in a timely fashion as BGP and IGP topologies change. Because RCP is an active participant in both the BGP and IGP protocols, delays due to message passing should be no worse than in today's routing architecture.

Transient inconsistencies. Transient inconsistencies might occur if routers do not receive updates from RCP in a certain order. For example, if a router's path to a destination includes routers for which RCP has already assigned a new path, transient forwarding loops could result. Although this pathology is likely no worse than the transient loops that occur during iBGP convergence today, it deserves further attention. In the future, routers might be modified to support a "commit" operation to allow for all routers along a path to execute an update at the same time.

6. Related Work

This section briefly surveys related research in routing architectures. Other approaches have been proposed for distributing routes within an AS and between ASes. Route reflectors [6] eliminate the need for a full mesh between iBGP speakers, but they do not correctly emulate full mesh iBGP: route reflectors forward only a single route for each prefix on behalf of its cluster, which may not be the route that each client of that route reflector would have selected in a full mesh. To address this shortcoming, RFC 1863 proposed that route servers forward *all* routes to clients, rather than just a single best route [23]. This proposal suggested using an "advertiser" attribute to allow recipients to know who advertised the routes. Similarly, Basu *et al.* proposed modifying route reflectors to advertise all routes that are equally good up to the MED step in the selection process to prevent iBGP route oscillation [5]. Because these proposals require modifying BGP, they have not been widely adopted. The route arbiter project proposed placing route servers at exchange points [19] to obviate the need for a full mesh eBGP topology (*i.e.*, at the exchange point) by applying policy once at the route server. This architecture facilitates centralized application of BGP routing policies at a single exchange point; RCP also focuses on improving other aspects of interdomain routing within an AS.

Several projects have advocated moving routing complexity to end hosts, which query route servers to discover routes [26, 42]. These projects share our goal of separating routing complexity from the infrastructure, but RCP also simplifies aspects of intra-AS routing and, unlike these proposals, does not focus on moving route selection to end hosts Others have proposed working around the existing infrastructure using an overlay to improve BGP's security [18] or robustness [2]. RCP could be a reasonable platform for deploying these architectures and overlay-based solutions.

The XORP project recognized that Internet research has suffered because router platforms are closed and has proposed an open software router interface to make all aspects of routing and forwarding both open and extensible [22]. In contrast, we propose making routing open and extensible by separating the routing protocol logic from the routers themselves. The IETF ForCES working group has also recognized that innovation has suffered because of the coupling between routing and forwarding [17]. In response, the group has proposed a framework that separates an *individual* network element into separate control and forwarding elements, which can communicate over a variety of media (*e.g.*, a backplane, Ethernet, etc.). The framework dictates that routing protocols be implemented in the control elements [41]. RCP is complementary to the ForCES framework: for example, RCP's algorithms for path selection could be implemented within one or more ForCES control elements.

7. Research Agenda

In addition to addressing the challenges discussed in Section 5, we intend to design specific algorithms and techniques for how RCP can improve interdomain routing in the following areas:

Configuration languages. RCP simplifies the underlying routing mechanisms, which can in turn simplify configuration languages. For example, configuring routing policy using RCP obviates the need for implementing high-level tasks with communities and complex import and export policies on individual routers. We believe that locating configuration state at the RCP should make it easier for operators to specify high-level tasks, leaving the mechanistic details of *how* these tasks are accomplished to RCP.

Correctness and security. Correctness and security should be intrinsic to the interdomain routing architecture. RCP should impose *invariants* on network configuration to guarantee correctness. For example, RCP can enforce consistent path assignment, as we described in Section 4. RCP could also enforce other correctness properties [11] by enforcing invariants. Defining what those invariants should be is an area for future work.

Troubleshooting and diagnostics. Because RCP is effectively a repository of the routing state for an AS, it can help operators debug routing and performance problems. Of course, for RCP to be a useful tool for troubleshooting and diagnostics, we must determine: (1) the problems that network operators commonly need to diagnose and (2) the state that RCP must maintain to be able to answer these questions.

Routing efficiency. We intend to explore how RCP could improve routing efficiency. For example, RCP could make routing more efficient by aggregating prefixes for a particular router's forwarding table if it could determine that the router would make the same forwarding decision for all of the more specific routes. An open question is how RCP can efficiently determine when aggregating contiguous prefixes is possible. Additionally, because RCP has a complete view of network state within an AS, we believe that it could be used to selectively advertise more specific prefixes for backup or inbound traffic engineering.

Acknowledgments

We would like to thank Albert Greenberg, Han Nguyen, and Brian Freeman at AT&T for suggesting the idea of an "NCP for IP networks", as a way to simplify network management and enable new services. We are also grateful to Matt Caesar, Albert Greenberg, Frans Kaashoek, Chuck Kalmanek, Z. Morley Mao, Rodrigo Rodrigues, and Renata Teixeira for their thoughtful feedback and discussions on earlier drafts of the paper.

REFERENCES

[1] A Border Gateway Protocol 4 (BGP-4). Internet Draft draft-ietf-idr-bgp4-24.txt, work in progress, November 2003.

[2] Agarwal, S., Chuah, C.-N., and Katz, R. H. OPCA: Robust interdomain policy routing and traffic control. In *Proc. IEEE OpenArch* (April 2003).

[3] Aiello, W., Ioannidis, J., and McDaniel, P. Origin authentication in interdomain routing. In *Proc. 10th ACM Conference on Computer and Communication Security* (Washington, DC, October 2003).

[4] Andersen, D. G., Balakrishnan, H., Kaashoek, M. F., and Morris, R. Resilient Overlay Networks. In *Proc. 18th ACM SOSP* (Banff, Canada, October 2001), pp. 131–145.

[5] Basu, A., Ong, C.-H. L., Rasala, A., Shepherd, F. B., and Wilfong, G. Route oscillations in IBGP with route reflection. In *Proc. ACM SIGCOMM* (August 2002).

[6] Bates, T., Chandra, R., and Chen, E. BGP Route Reflection - An Alternative to Full Mesh IBGP. RFC 2796, April 2000.

[7] Clark, D., Wroclawski, J., Sollins, K., and Braden, B. Tussle in cyberspace: Defining tomorrow's Internet. In *Proc. ACM SIGCOMM* (August 2002).

[8] Draves, R., King, C., Venkatachary, S., and Zill, B. Constructing optimal IP routing tables. In *Proc. IEEE INFOCOM* (March 1999).

[9] Dube, R. A comparison of scaling techniques for BGP. *ACM Computer Communications Review 29*, 3 (July 1999), 44–46.

[10] Feamster, N., Andersen, D., Balakrishnan, H., and Kaashoek, M. F. Measuring the effects of Internet path faults on reactive routing. In *SIGMETRICS* (San Diego, CA, June 2003).

[11] Feamster, N., and Balakrishnan, H. Towards a logic for wide-area Internet routing. In *ACM SIGCOMM Workshop on Future Directions in Network Architecture* (August 2003).

[12] Feamster, N., and Balakrishnan, H. Verifying the correctness of wide-area Internet routing. Tech. Rep. MIT-LCS-TR-948, Massachusetts Institute of Technology, May 2004.

[13] Feamster, N., Borkenhagen, J., and Rexford, J. Techniques for interdomain traffic engineering. *Computer Communications Review 33*, 5 (October 2003).

[14] Feamster, N., Rexford, J., Shaikh, A., and van der Merwe, J. Routing control platform: Architecture and practical concerns. `http://www.research.att.com/~kobus/rcp-tr.pdf`, June 2004.

[15] Feamster, N., Winick, J., and Rexford, J. A model of BGP routing for network engineering. In *SIGMETRICS* (June 2004).

[16] Feigenbaum, J., Papadimitriou, C., Sami, R., and Shenker, S. A BGP-based mechanism for lowest cost routing. In *Proc. 21st Symposium on Principles of Distributed Computing* (July 2002).

[17] Forwarding and Control Element Separation (ForCES) Charter. `http://www.ietf.org/html.charters/forces-charter.html`.

[18] Goodell, G., Aiello, W., Griffin, T., Ioannidis, J., McDaniel, P., and Rubin, A. Working around BGP: An incremental approach to improving security and accuracy of interdomain routing. In *Proc. Network and Distributed Systems Security 2003, Internet Society* (February 2003).

[19] Govindan, R., Alaettinoglu, C., Varadhan, K., and Estrin, D. Route servers for inter-domain routing. *Computer Networks and ISDN Systems 30* (1998), 1157–1174.

[20] Griffin, T., and Wilfong, G. Analysis of the MED oscillation problem in BGP. In *Proc. International Conference on Network Protocols* (Paris, France, November 2002).

[21] Griffin, T. G., and Wilfong, G. On the correctness of IBGP configuration. In *Proc. ACM SIGCOMM* (August 2002).

[22] Handley, M., Hudson, O., and Kohler, E. XORP: An open platform for network research. In *Proc. SIGCOMM Workshop on Hot Topics in Networking (HotNets)* (October 2002).

[23] Haskin, D. A BGP/IDRP Route Server alternative to a full mesh routing. RFC 1863, October 1995.

[24] Horing, S., Menard, J. Z., Staehler, R. E., and Yokelson, B. J. Stored program controlled network: Overview. *Bell System Technical Journal 61*, 7 (September 1982), 1579–1588.

[25] Jannotti, J. *Network Layer Support for Overlay Networks*. PhD thesis, Massachusetts Institute of Technology, 2002.

[26] Lakshminarayanan, K., Stoica, I., and Shenker, S. Routing as a Service. Tech. Rep. UCB-CS-04-1327, UC Berkeley, 2004.

[27] Lawser, J. J., LeCronier, R. E., and Simms, R. L. Stored program controlled network: Generic network plan. *Bell System Technical Journal 61*, 7 (September 1982), 1589–1598.

[28] Mahajan, R., Wetherall, D., and Anderson, T. Interdomain routing with negotiation. Tech. Rep. CSE-04-06-02, University of Washington, May 2004.

[29] Mao, Z. M., Govindan, R., Varghese, G., and Katz, R. Route flap damping exacerbates Interet routing convergence. In *Proc. ACM SIGCOMM* (August 2002).

[30] Mao, Z. M., Rexford, J., Wang, J., and Katz, R. H. Towards an accurate AS-level traceroute tool. In *Proc. ACM SIGCOMM* (August 2003).

[31] McPherson, D., Gill, V., Walton, D., and Retana, A. Border gateway protocol (BGP) persistent route oscillation condition. RFC 3345, August 2002.

[32] Nakao, A., Peterson, L., and Bavier, A. A routing underlay for overlay networks. In *Proc. ACM SIGCOMM* (August 2003).

[33] Qiu, L., Yang, R., Zhang, Y., and Shenker, S. Selfish routing in Internet-like environments. In *SIGCOMM* (August 2003).

[34] RouteScience. Whitepaper available from `http://www.routescience.com/technology/tec_whitepaper.html`.

[35] Shaikh, A., and Greenberg, A. OSPF monitoring: Architecture, design, and deployment experience. In *Proc. First Symposium on Networked Systems Design and Implementation (NSDI)* (San Francisco, CA, March 2004).

[36] Sockeye. `http://www.sockeye.com/`.

[37] Subramanian, L., Roth, V., Stoica, I., Shenker, S., and Katz, R. Listen and whisper: Security mechanisms for BGP. In *Proc. First Symposium on Networked Systems Design and Implementation (NSDI)* (San Francisco, CA, March 2004).

[38] Teixeira, R., and Rexford, J. A measurement framework for pin-pointing routing changes. In *ACM SIGCOMM Workshop on Network Troubleshooting* (September 2004).

[39] Teixeira, R., Shaikh, A., Griffin, T., and Rexford, J. Dynamics of hot-potato routing in IP networks. In *Proc. ACM SIGMETRICS* (June 2004).

[40] Villamizar, C., Chandra, R., and Govindan, R. BGP Route Flap Damping. RFC 2439, November 1998.

[41] Yang, L., et al. Forwarding and Control Element Separation (ForCES) Framework. RFC 3746, April 2004.

[42] Yang, X. NIRA: A new Internet routing architecture. In *ACM SIGCOMM Workshop on Future Directions in Network Architecture* (August 2003).