OPCA: Robust Interdomain Policy Routing and Traffic Control

Sharad Agarwal†
Computer Science Division
University of California, Berkeley
sagarwal@cs.berkeley.edu

Chen-Nee Chuah†
Dept. of Electrical & Computer Engineering
University of California, Davis
chuah@ece.ucdavis.edu

Randy H. Katz
Computer Science Division
University of California, Berkeley
randy@cs.berkeley.edu

Abstract— **An increasing number of ASes have been connecting to the Internet through the BGP inter-domain routing protocol. With increasing stress on the scale of this system and increasing reliance on Internet connectivity, more participants demand additional functionality from inter-domain routing that BGP cannot handle. For example, we believe that the recent trend towards multihomed stub networks exhibits a likely intent to achieve fault tolerant and load balanced connectivity to the Internet. However, BGP today offers route fail-over times as long as 15 minutes, and very limited control over incoming traffic across multiple wide area paths. More research literature and news media are calling for stemming malicious or erroneous routing announcements. We propose a policy control architecture, OPCA, that runs as an overlay network on top of BGP. OPCA allows an AS to make route change requests at other, remote ASes to achieve faster route fail-over and provide capabilities to control traffic entering the local AS. The proposed architecture and protocol will co-exist and interact with the existing routing infrastructure and will allow for a scalable rollout of the protocol.**

I. Introduction

A. Trends in Inter-Domain Routing

The Border Gateway Protocol (BGP) [1] is the de-facto inter-domain routing protocol between Autonomous Systems (ASes) that achieves global connectivity while shielding intra-domain routing details and fluctuations from the external view. Recent studies of BGP [2], [3] have indicated a significant growth in BGP routing tables, an increase in route flapping and unnecessarily specific route announcements. The large growth in the number of ASes that participate in BGP peering sessions has been fueled by stub ASes. Our analysis of the BGP data from Routeviews [4] reveals that at least 60% of these stub ASes are *multi-homed* to two or more providers, i.e., they announce BGP routes via multiple upstream ASes.

This trend towards increased connectivity to the Internet and participation in inter-domain routing is placing more stress on the BGP infrastructure. The scale of routing is increasing, and more features are being expected out of it than BGP was designed to handle. For instance, this increasing trend towards multi-homing is intended as a solution to achieve two goals: fault tolerance and load balancing on inter-domain routes.

B. Features Absent in BGP

As an illustration, Figure 1 compares two scenarios where a stub AS is (a) single-homed and (b) multi-homed to three providers. The stub AS, W, in Figure 1(b) can choose to have its traffic go primarily through ISP X. If the link to ISP X fails, or when there are failures along the path through ISP X, W can failover to ISP Y or Z. If it were singly homed, as in case (a), it could only protect itself against upstream link failures by purchasing multiple redundant links to ISP X. In addition, W can load-balance its outgoing traffic by selecting the best route to the destinations via one of the three providers. Routescience [5] and others automate outgoing traffic balancing by selecting specific BGP announcements heard from different providers.

Achieving connectivity by subscribing to multiple providers is likely to be expensive, but Mortimer's study [6] suggests that reliability is a deciding factor. However, the effectiveness of multi-homing is limited by the slow convergence behavior of BGP. Inter-domain routes can take upto 15 minutes [7] to fail-over in the worst case. For companies that rely on Internet connectivity to conduct online transactions, such a long outage can have a severe financial impact. Furthermore, BGP allows an AS little control over how the incoming traffic enters its network.

As more networks connect to the Internet via BGP, the likelihood of router misconfiguration will increase. With more participants, the chance of having a malicious or compromised participant grows. As has been observed in the past [8], a single incorrect routing announcement can seriously impact data traffic. As yet, no protocol exists for detecting and stemming such bogus route announcements.

As more and more applications are enabled on the Internet, traffic will grow and may become more unpredictable. Higher traffic use may incur higher transit costs for stub networks that rely on ISPs for Internet service. During periods of abnormal traffic patterns or even link failures, it may be advantageous for two entities that have a business relationship for exchanging traffic to temporarily modify this agreement to improve congestion or reduce transit costs. As yet, no protocol exists for negotiating and applying such temporary agreements.

C. Solution

Instead of overloading BGP with protocol extensions, we propose to address these problems by developing an Overlay Policy Control Architecture (OPCA) running on top of BGP to facilitate policy exchanges. Our architecture relies on knowing AS relationships and the AS level hierarchy. While OPCA can be used to address the shortcomings of the current infrastructure, we focus on two main goals:

- to support fast, fine grained management of incoming traffic across multiple incoming paths, and
- to reduce the fail-over time of inter-domain paths.

†Sharad Agarwal and Chen-Nee Chuah are also affiliated with the IP & Interworking Division of the Sprint Advanced Technology Laboratories. This research was supported by Cisco Systems and the California MICRO Program, with matching support provided by Ericsson, Nokia, and Sprint.

Together, these goals serve to improve routing and traffic in the current inter-domain routing structure, and allow it to scale better to the growing number of multi-homed ASes.

OPCA consists of a set of intelligent Policy Agents (PAs) that can be incrementally deployed in all the participating AS domains. The PAs are responsible for processing external policy announcements or route-change requests while adhering to local AS policies, and enforcing necessary changes to local BGP routers. These PAs communicate with one another via a new Overlay Policy Protocol (OPP). Such an overlay architecture allows ASes to negotiate the selection of inter-domain paths for incoming traffic with remote ASes, leading to more predictable load-balancing performance. In addition, an AS can request routing changes to other ASes to expedite fail-over. Our architecture does not require any modifications to the BGP protocol or to existing BGP routers.

We will review related work in the next section. Section III describes the design of our architecture. In Section IV, we explain the rationale behind our design of OPCA. We follow this with a description of the applications of our architecture in Section V. We end with some deployment and scaling issues in Section VI and conclusions in Section VII.

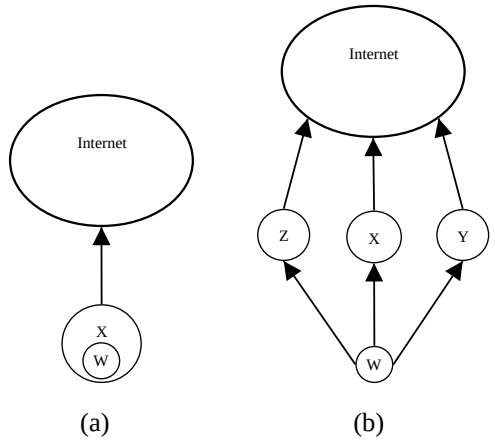

Fig. 1. (a) Company W with ISP X. (b) Company W with ISP's X, Y and Z

II. Related Work

Huston [9] suggests ways to address the problems of load balancing and route fail-over. There are two main approaches: (a) extending the current BGP protocol/implementations or (b) use alternate routing through overlay networks or replace BGP with a new interdomain routing protocol.

Various BGP-based solutions have proposed to limit the advertisement scope of route announcements [10], [11], [12]. For example, BGP can be modified to allow bundling of routes or to specify aggregation scopes. These proposals may limit the ill-effects of multi-homing but do not solve the issues of fast fail-over and inbound load balancing that we are concerned with. Pei [13] proposes modifications to BGP to reduce convergence time by identifying which withdrawals are due to actual failures and filtering out announcements of invalid or conflicting paths during the transient periods. He employs a new community attribute to convey routing policy information over the traditional BGP session. However, it does not address the issue of wide-area traffic balancing. These schemes require a new version of BGP to be deployed or many BGP routers to be reconfigured. This is difficult to accomplish given the widespread use of BGP. Mortier [14] proposes a new route attribute that expresses a price for carrying traffic on the advertised route that one neighbor charges another. While this scheme adds another interesting attribute that can be used in route selection, it does not address the goals of our work in reducing route failover times and managing incoming traffic.

All the afore-mentioned approaches rely on "in-band" signaling, i.e., they embed and distribute policy information in BGP routing messages. In this paper, we explore an orthogonal approach by introducing "out-of-band" signaling through OPCA to permit more flexibility and control of the policy distributions and negotiations between AS domains. This results in more predictable performance for inbound traffic engineering. In fact, OPCA could potentially leverage these other BGP modifications to obtain accurate connectivity information in a more efficient manner, and based on this, make better policy decisions.

In the early days of BGP, Estrin [15] proposed a centralized routing arbiter that collects all routing entries, centrally computes the "best" routes, and re-distributes the final routing entries. We believe that such an architecture is not deployed in the Internet today due to its complexity and scalability issues. Alternative routing architectures have been proposed, such as RON [16], Nimrod [17], and BANANAS [18]. RON is an overlay network that uses *active probing* and *global link state* in a fully *meshed* network to customize routing between overlay nodes. RON is designed for applications with a small number of participating nodes and cannot scale to the number of ASes that exist today. Nimrod [17] was proposed in 1996 as an alternative inter-domain routing architecture. It would distribute link-state information and support three forms of routing: MPLS-like flow routing, BGP-like hop by hop routing and data packet specified routing. BANANAS [18] also distributes link state information and allows a sender to specify the full path for each packet. Although sender specified routing can help achieve load balancing and fault tolerance, packets will no longer go through the fast path in routers (such as Cisco Express Forwarding) and will require more processing and hence delay. Nimrod and BANANAS also introduce the difficulty of timely link-state propagation which has not yet been addressed. These solutions propose to change the underlying routing protocol itself to route traffic either on optimal or source-dictated paths. Our approach reuses the existing BGP routing protocol and builds an overlay layer to achieve explicit control over policy distributions and load balancing.

Frameworks and protocols for distributing policies within a domain that are based on MPLS, DiffServ or IntServ have been proposed, e.g., COPS and Bandwidth Broker [19], [20], [21], [22]. MPLS Fast-Reroute maintains backup paths and switching entries for every link computed from link state information flooded through the local network. In our solution, we focus on the *inter*-domain case and do not rely on the widespread deployment of DiffServ or MPLS, but instead rely on the already widespread deployment of BGP.

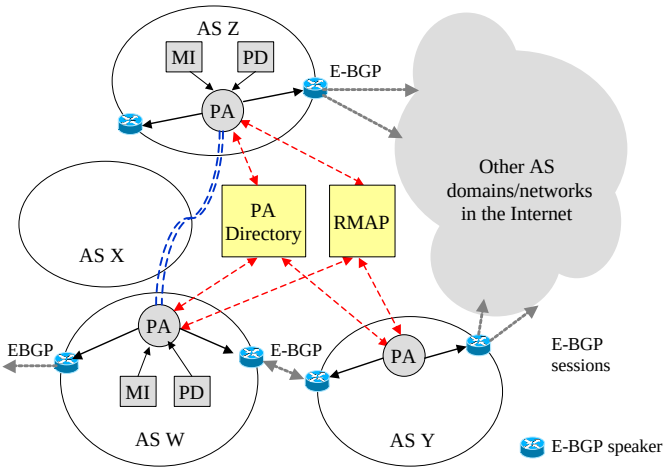

Fig. 2. Overlay Policy Control Architecture

III. Overlay Policy Control Architecture (OPCA)

A. Overview

The Overlay Policy Control Architecture (OPCA) is designed to support fault-tolerant and efficient wide-area routing. Our approach leverages intra- and inter-AS measurement and monitoring systems that are commonly deployed in ASes to obtain simple data on traffic load changes and network performance. As shown in Figure 2, OPCA consists of five components: Policy Agents (PAs), Policy Databases (PDs), Measurement Infrastructures (MIs), the PA directory and the AS Topology and Relationship Mapper (RMAP). Policy Agents (PAs) are intelligent proxies that reside in each AS domain that agrees to participate in the overlay policy distribution network. Most Internet Service Providers (ISPs) today have deployed a measurement infrastructure to monitor the performance of their network. They also maintain a database of service level agreements (SLAs) and peering arrangements. We assume that OPCA can reuse any of such existing PDs and MIs within an AS domain. The PA directory and RMAP are new query-services introduced by OPCA. The next section describes each of these components in detail. We have completed the design of the PA protocol and the RMAP component. We will reuse existing MIs and use the already deployed DNS for the PA directory. We are currently implementing the PA and PD and designing the evaluation methodology.

B. Components of OPCA

B.1 Policy Agent (PA)

Intelligent Policy Agent (PAs) are introduced in each AS domain that employs OPCA to form an overlay policy distribution network. The PAs are responsible for processing external policy announcements, processing local AS policies, and enforcing these policies at border routers participating in external BGP sessions. This implies that PAs should have administrative control over the E-BGP speakers within their respective AS domains. The E-BGPs are dynamically (re)configured to reflect policy changes, and continue to perform route selection based on these policies. However, some form of synchronization may be needed with PAs to prevent simultaneous conflicting changes from network operators. A centralized or distributed PA directory is needed to allow distant PAs (PAs belonging to different ASes) to communicate with each other. Each PA should be accessible at an IP address and port that can be reached from distant ASes.

A routing policy will be influenced by traffic measurements between the local AS and one or more distant ASes that are important, such as application level customers. To impose a change on the routing between these sites, a local PA will have to negotiate with the remote PA, and possibly other intermediate policy agents. Contacting the appropriate intermediate PAs will be important, since conflicting routing and filtering policies in other ASes along the route can severely impact the routing change. Having an understanding of the relationships between ASes along prospective routes [23] will be important to ensure the effectiveness of protocol.

The protocol that the PAs use to communicate with one another is the new overlay policy protocol (OPP). The design of PAs and the OPP protocol is subject to certain constraints:

- The PAs should communicate with BGP speakers via conventional means, and should not require any modifications to the routers. This is important for the acceptability and deployment of this protocol.
- The PAs should detect and identify policy conflicts at runtime, and avoid potential BGP oscillations or divergence.
- OPP should co-exist with the widely deployed IGP/EGP today such as BGP, IBGP, IS-IS or OSPF.
- The use of OPP should not increase BGP route flapping and the number of routing table entries.
- The correctness and effectiveness of OPCA should not rely on every AS employing PAs. We strive to support incremental deployment, i.e., early adopters of the new PAs should not be at a disadvantage compared to those continuing to use only BGP, even though the utility of the new system may increase as more ASes subscribe to OPCA.

B.2 Policy Database (PD)

The policy database is a local repository of information that will help the local PA decide how to change routing for its domain. The PD should provide the following data:

- *Ordered list of remote ASes containing the local domain's main customers.* This list identifies the target ASes that the PA should focus its load balancing efforts at. This list can easily be formed by examining the logs of the service provided by the local domain [24].
- *List of local application servers.* The PA needs to know which set of IP addresses serve content or any other service to the remote customers. The majority of traffic will likely be destined for or originate from these local servers. The PA will be concerned with the load balancing and fail-over of routes to these addresses.
- *Pricing constraints and SLAs.* The PA will try to balance traffic across multiple ISP links evenly weighted by actual link capacity. However, the local domain may prefer to weight traffic by pricing structures imposed by the SLAs it is operating under. If this is the case, the PA will need to be informed about these

price constraints.

B.3 Measurement Infrastructure (MI)

Most ISPs and customer bases already employ some form of a measurement infrastructure (MI). Some may use it to verify Service Level Agreement (SLA) specifications with a third party, or may use it to manage their network. In our architecture, we assume that such an infrastructure already exists in each domain employing OPCA. The MI helps the PA keep track of the effects of the PAs alterations to routes and decide when such an alteration is necessary. The MI should provide the following data:

- *E-BGP link characteristics.* The PA needs data on each of the links connecting the local AS to the Internet. This data should include actual link capacity and current bandwidth load. This allows the PA to decide which links are underutilized and to verify the effect of policy changes that it imposes. PA control traffic will also go over these links. Simple SNMP statistics or even periodic polls of the Cisco IOS interface byte counters will suffice.
- *Customer-server traffic characterization.* The MI should also provide data outlining characteristics of traffic (such as total bandwidth, average latency) that each main customer sends/receives to/from each local server. This helps the PA understand the current traffic distribution and how to best redistribute this load. Tools such as Cisco Netflow should be able to provide such data.

B.4 PA Directory

The PA directory is a means by which a local domain's PA can locate the address for a distant AS's PA. This is necessary because some ASes may not have PAs, since we do not rely on immediate wide scale deployment of OPCA, and those that do have PAs can place them anywhere inside their network. The design of the directory is not a goal of our work. The system can use one or multiple directories, which may or may not be coherent with each other. From the architecture's point of view, there is logically one PA directory. A simple solution such as using the already deployed DNS can be used. The PA directory's fully qualified domain name is known to all PAs and PAs need only make DNS queries to locate other PAs.

B.5 AS Topology & Relationship Mapper (RMAP)

The RMAP is a repository of the inter-AS relationships and Internet hierarchy. These relationships determine how routing and traffic flows on the Internet as governed by route export rules. Export rules dictate that when sending routes to a customer AS, a provider has to send it all the routes it knows of. When sending routes to a provider or peer, an AS does not send other peer or provider routes. In the RMAP, we deduce these relationships from multiple BGP routing tables by applying heuristics. Our heuristic begins by ranking the ASes based on their distance in the AS paths from the AS where we collect each routing table. We then infer whether an AS-AS link represents a peer-peer or a provider-customer relationship based on the relative ranks of each AS pair that shares a link. Using pruning, greedy ordering and weak cuts designed to expose the different business classes of Internet service providers, we also infer the hierarchical structure of the AS topology that exists today. Details of our algorithm can be found in our prior work [23].

The RMAP is an essential component to OPCA. The architecture is indifferent to whether there is only one global RMAP or if multiple RMAPs exist for different ASes. The only requirement is that the RMAP have access to enough diverse BGP routing table dumps as to be mostly accurate. We will revisit this issue in Section IV. PAs need the RMAP to find the likely path a distant AS uses to reach it and the intermediate AS relationships that determine how routes propagate to the distant AS. This also helps to determine if a policy request will conflict with a distant AS's local routing policy.

C. Overlay Policy Protocol

C.1 Message Propagation

OPP carries inter-PA messages from the source PA to the destination PA over UDP by rendezvousing first through the PA directory. Therefore, PA control messages do not have to through every PA in the intermediate ASes the same way as BGP announcements propagate from one BGP speaker to another, because the originating PA has the burden of determining the appropriate destination PA to contact directly. The advantages of this approach are

- eliminate intermediate PA processing
- keep convergence time of PA messages lower than that of BGP messages
- give originating PA more control
- give more privacy to originating PAs

C.2 Connections and Sessions

We choose unreliable, connectionless UDP over reliable, connection-based TCP because the reverse route may be unavailable during link failure. Consider the case in Figure 5 where X uses the path X $\rightarrow$ F $\rightarrow$ C $\rightarrow$ A to reach A. Suppose the C $\rightarrow$ A link fails and A wants X to failover to the F $\rightarrow$ E $\rightarrow$ B $\rightarrow$ A path. A's PA has to contact F's PA to make the routing change. A already has an alternate path to F via B and can use it to send the message to F. However, until F receives the PA request and implements it, it will not have a working return path to A to send a response. This is why OPP uses UDP messages because TCP connections cannot be setup in some circumstances. In some conceivable scenarios of multiple concurrent outages, even multi-hop PA communication may be required. During link failure, a PA may not be able to receive replies from the PD. If there are DNS servers available via backup links, then this is not an issue. We are exploring how a PA can proactively cache the locations of various other PAs that it would need to contact during link failure.

C.3 OPP Messages

Since all the OPP messages will be connectionless UDP messages, they will have to identify the sending PA's IP address, UDP port and AS number. In Figure I, we list the messages that OPP will support for fast failover and incoming traffic load balancing. The error codes will include such errors as invalid request, conflict with peering policy, unknown address range

TABLE I
OPP MESSAGES

Message	Description
PA_locate(AS) e.g., PA_locate(25)	Message from a PA to the PA directory requesting the address of a PA in a remote AS
PA_locate_reply(AS,ipaddr,port,timeout) e.g., PA_locate_reply(25,169.229.62.121,8919,6000)	Reply from the PA directory, containing the IP address and port of the requested PA and the number of seconds after which the reply is invalid
PA_route(prefix) e.g., PA_route(128.32.0.0/16)	Request from a PA to another PA for the best route chosen for a particular prefix
PA_route_reply(prefix,AS_Path) e.g., PA_route_reply(128.32.0.0/16,11423 25)	Reply from a PA giving the AS path chosen for a network prefix
PA_block(prefix,AS1,AS2) e.g., PA_block(128.32.0.0/16,25,11423)	Request from a PA to another PA to block all announcements for a particular prefix (or all addresses if null) that contain "AS1 AS2" anywhere in the AS path
PA_block_reply(error_code,prefix,AS1,AS2) e.g., PA_block_reply(0,128.32.0.0/16,25,11423)	Reply from a PA giving the status of a previous block request
PA_select(prefix,X,Y) e.g., PA_select(128.32.0.0/16,7018,50)	Request from a PA to another PA to select the announcement for a particular prefix from AS-X when re-announcing to AS-Y
PA_select_reply(error_code,prefix,X,Y) PA_select_reply(1,128.32.0.0/16,7018,50)	Reply from a PA giving the status of a previous select request

and address range ownership verification failed. Each PA will have the option of using publicly available address allocation registries to verify that a request to change routing for an address range came from the AS that owns it. Further, we expect a deployed system to use a public key authentication system to provide additional protection against malicious use. The PA directory will have a public key that is known widely. All replies from the PA directory will have to be signed by its private key and will contain the public key of the PA who's address is being queried. All PAs that receive requests will verify that they were signed by the private key belonging to the originating PA.

We are continuing to define all the error codes and additional messages that will be needed for augmenting peering relationships, detecting bogus routes and blocking bogus routes.

IV. OPCA DESIGN RATIONALE

In this section, we discuss the design rationale behind OPCA. We begin by examining why using a separate control path from BGP helps us achieve our goals. We also discuss the use of multiple BGP views to improve the accuracy of OPCA.

A. Separating Policy and Routing

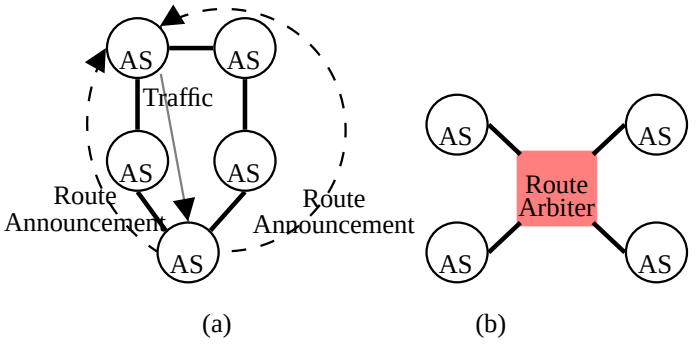

Fig. 3. (a) Current Routing Scenario (b) Route Arbiter Scenario

The key design constraint of the current inter-domain routing system as shown in Figure 3(a) is that domains advertising reachability in BGP have little control over how those advertisements propagate and are applied. This influences how traffic reaches the advertising domain. The end AS has better aggregate knowledge on the characteristics of the traffic entering its network but can do little to change routing at distant ASes to affect the traffic flow.

The alternate structure shown in Figure 3(b) has been proposed in the research literature as we described in Section II. While in this scenario an end AS may be able to influence which routes are used to reach it, there are many issues that need to be resolved, such as scalability of link state information, computational complexity, full disclosure of routing policies and loss of route selection control for each AS.

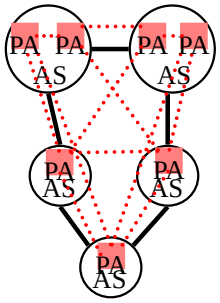

Fig. 4. Our Approach

Instead we propose to separate routing and policy as shown in Figure 4. We continue to use the BGP infrastructure to disseminate and select routes, thereby avoiding issues such as the link state propagation, route selection control, computational complexity and full disclosure of routing policies. We augment the system with an overlay protocol to explicitly request routing policy changes between ASes directly, where the two ASes can negotiate policy conflicts and exchange more routing and traffic information.

In theory, we could propose our own BGP community attribute and distribute policy control information along with BGP routes. However, the advertiser will not know a priori which remote BGP speakers adhere to the new community specification, which makes the results of such approaches less predictable.

Furthermore, the system would suffer from the same BGP convergence times.

The routing policy conveyed by BGP is essentially "control signaling" that attempts to influence the route selection process at the remote router. The basic routing functionality only requires exchange of connectivity information (e.g., AS Path and Nexthop). Therefore the key premise behind OPCA is that separating the policy (the control part) from routing (basic connectivity information) will provide much more flexibility in controlling inter-domain paths, which can potentially lead to better performance, new services, and prevention of bogus routes.

The specific advantages of this approach include:

- *Reduced response time for failure recovery*

When there is a failure or a change in routing, BGP today can take up to 15 minutes to converge to a new route. By the time traffic is forwarded correctly on the new path, network conditions could be vastly different from what prompted the change in the first place. During transient fail-over periods, traffic can get caught in routing loops or become black-holed, resulting in increased packet loss or delay jitter. This can adversely impact the perceived quality of networked applications. OPCA helps to hasten failure recovery by determining the distant AS that sees both (or more) possible paths. It allows the local AS to contact this distant AS and switch the route instead of waiting for normal BGP failure recovery. Hence, the response time of route changes in the data plane can be greatly reduced.

- *Enhanced traffic engineering performance*

Two major difficulties in inter-domain traffic engineering are AS Path asymmetry and localized routing decisions. For example, when an AS-X supplies hints of how AS-Y can reach it via a set of AS paths, it is up to AS-Y to decide which path to choose, and hence could diminish the intended load balancing results. The outgoing path from AS-X to AS-Y can be different from the return path, and may experience different performance in terms of delay and losses. OPCA allows ASes to directly negotiate routing policies with one another and determine the best possible inter-domain path. Using the same example, once AS-X and AS-Y reach a consent, OPCA will configure their BGP routers to select the agreed-upon BGP route. The intermediate ASes can also be updated if necessary.

- *Augmenting peering relationships*

Two ASes are said to have a peering relationship if they agree to exchange traffic directly without exchanging money. However, this also implies certain routing constraints that do not allow transit traffic to flow through a peering link. Through a separate layer of policy distribution and negotiation, OPCA can allow two ASes to allow temporary passage of certain transit traffic in exchange for money to reduce congestion or mitigate network partitions.

- *Better support for detecting and handling misconfigured or bogus routes*

In OPCA, participating PAs communicate with one another via an overlay protocol. Authentication mechanisms can be easily added during the initial hand-shakes to verify route announcements from a particular AS. A PA can configure its BGP routers to reject bogus or misconfigured routes that are not "authenticated" by a specific originating AS. Also, if an AS detects a bogus route, it can use OPCA to inform other ASes and stem the spread of the bogus route by requesting route filtering at an AS close to the source of the bogus route.

B. Accuracy, Correctness and Security

Our architecture's core algorithm relies on inferring the AS-level topology, AS relationships, the likely paths between two points and the path convergence points. The efficiency of our algorithm will rely partly on the accuracy of these inferences from the RMAP.

We built the RMAP because no oracle exists that can be queried about the AS structure of the Internet and the relationship between any pair of ASes. Due to the lack of such an oracle, we cannot check the accuracy of our inferences. We have verified our inferences by comparing them to additional routing table views [23] and in most cases found fewer than 3% errors. We also have contacted two major ISPs and verified the list of AS relationships that involve them. We exploit access to multiple routing tables to improve both the completeness and the accuracy of our inferences. If more ASes participate in this architecture, then more views can be used. Since peering relationships between ASes do not change rapidly, the RMAP does not need to be recomputed often with fresh BGP tables.

The RMAP improves the efficiency of the protocol by quickly identifying likely route paths, route convergence points, and routing relationships between ASes at and near the convergence points. If no RMAP existed, OPCA can continue to function, but with much more signaling overhead. A PA would have to contact several distant PA's to find out where routes that it is announcing are propagating and find convergence points for its routes. Once found, the PA would have to make several policy change requests because many of them may be denied due to conflicts with neighboring AS relationships.

PAs will not apply routing policies that conflict with their own local peering policies. In this way, the ill effects of having an inaccurate RMAP or having malicious PAs can be kept to a minimum. The OPP protocol will only allow simple requests to be made such as a request for route information, for route selection change or termination of route propagation. None of these requests would violate the BGP protocol. The BGP protocol already allows each AS to use practically any form of route selection by the application of local policies. We can incorporate a standard form of authentication and address ownership verification to the PA protocol as is proposed for BGP itself [25]. This would ensure that an AS can only influence routing for its own address space and limit OPCA's exposure to misconfigured or malicious PAs.

V. Applications of OPCA

In this section, we briefly describe how OPCA achieves fast fail-over of inter-domain routes and inbound load-balancing.

A. Improving Route Fail-Over Time

From Labovitz [7], we know that the fail-over time for BGP routes can be as long as 15 minutes. He showed this through experiments where he injected routes into BGP speakers on the Internet and measured their fail-over time. It is currently infeasible to measure the route fail-over times of all the routes on the Internet to determine how common this long convergence time

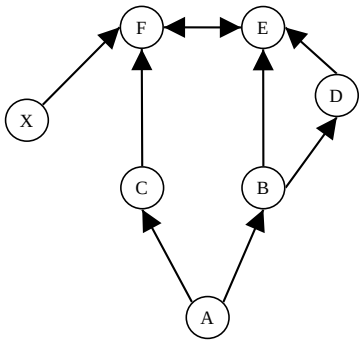

Fig. 5. Routing Scenario

is. However, Labovitz has shown that the fail-over time depends on the length of the longest alternative path. Consider the trivial example shown in Figure 5. An arrow from AS A to B indicates that B is A's provider. A double-ended arrow indicates a peer-peer relationship. Suppose X is using the path X → F → E → B → A to reach A. If the A → B link fails, A would want X to quickly switch over to using the route through C, because most of A's important customers reside in X. Traditional BGP fail-over would involve the following sequence of steps:
1. B's BGP session with A resets
2. B sends a route withdrawal to E and D
3. E receives the route withdrawal, re-runs route selection, chooses the previously heard route through D and announces it to F
4. F receives the announcement, chooses it and announces it to X
5. D receives the route withdrawal, re-runs route selection and sends a withdrawal to E
6. E receives the withdrawal, re-runs route selection and sends a withdrawal to F
7. F receives the withdrawal, re-runs route selection, chooses the previously heard route through C and announces it to X

Instead, in OPCA, A's PA can contact F's PA directly and request the change, thereby shortcutting the long BGP convergence. The following sequence of events would take place:
1. A stops receiving traffic on the link from B and assumes the link is down
2. A's PA queries the RMAP, determines that the best feasible route for X is through C and that F is the best control point
3. A's PA sends PA_locate(F) to the PA directory and receives PA_locate_reply(F,ipaddr,port,timeout)
4. A's PA sends PA_block(prefix,A,B) to F's PA via the link A-C
5. F's PA applies the change at it's BGP routers, sends PA_block_reply(0,prefix,A,B) to A's PA
6. F sends the new route to X

Alternatively, if F does not have a PA, A can contact E and have it block the incorrect routes. This is a very simple example. A more complicated scenario will be more common, where various routing pathologies can cause normal BGP route convergence to be especially delayed. It is important to note that in the case of BGP, the route announcements add to intermediate routers' CPU and memory load, and are subject to dampening. On the other hand, the OPP messages are only exchanged between OPCA components, and do not experience per-hop BGP delay. Once a PA successfully convinces a remote PA (usually at an aggregation point) to switch to an alternate route, the remote PA can configure the local BGP router(s) to select this new path as the best route. In the example shown above, the time taken for the system to stabilize is the time taken to send and process OPCA messages from Step 2-5 and for F to propagate new BGP routes to X. Hence, the overhead cost of OPCA is fixed regardless of the length of Internet paths.

The advantage of our architecture is that PAs can directly communicate with other PAs when necessary instead of using a multi-hop, dampened protocol like BGP. This allows the benefits of a fully meshed communication architecture, without the pitfalls because the actual routes still propagate through the underlying hierarchical BGP structure. It is important to note here the role of the RMAP component. It is important for A's PA to know what the topology, as shown in Figure 5, is in order to contact the relevant remote PAs. Also, it needs to know if alternate routes are feasible. In this scenario, X → F → C → A and X → F → E → B → A and X → F → E → D → B → A are all valid. However, if B and E were peers instead of customer and provider, then the second path would be invalid based on commonly followed route export rules [23].

B. Incoming Traffic Load Balancing

Direct PA to PA communication also helps achieve incoming traffic balancing. Consider again the example in Figure 5 where AS A is load balancing its links to C and B, primarily for some application level customer X. With respect to X, F is the aggregation point for routes to A via either C or B. Pathological route updates need not be sent in BGP by A to "game" route selection at F. Instead, A's PA will contact F's PA directly and request specific route selection at F. A's PA has to first determine the structure of this graph and that F is the aggregation point. It also has to determine whether its route selection requests will conflict with F's local route selection policies based on F's peering arrangements. The RMAP helps the PA in these respects.

C. Other Applications

OPCA can be applied to solve other problems beyond our stated goals of fast fail-over and traffic balancing. It can be used to query the BGP table at remote ASes to help in the diagnosis of router misconfigurations or find the source of malicious BGP advertisements. It can be used to request filtering of certain routes to prevent the spread of bogus BGP announcements. It can also be used to also arrange per address prefix micro-peering [26]. We do not explore these applications here.

VI. Performance Evaluation and Future Work

A. OPCA Deployment

Ideally, we would like to deploy OPCA in all ASes to maximize the benefits of the architecture, but this may not be practical. Instead, we consider the case of incremental deployment and we want to analyze the marginal utility of introducing PAs in selected ASes. We find that most of the benefits can be gained by deploying PAs at a) multi-homed stub ASes, and b) ASes where most inter-domain routes converge. This is because these convergence points control which of the multiple possible routes

TABLE II
INFERRED RELATIONSHIPS FOR 23,935 AS PAIRS

Relationship	# AS pairs	Percentage
Provider-customer	22,621	94.51%
Peer-peer	1,136	4.75%
Unknown	178	0.74%

TABLE III
DISTRIBUTION OF ASES IN THE HIERARCHY

Level	# of ASes
Dense core (0)	20
Transit core (1)	129
Outer core (2)	897
Small regional ISPs (3)	971
Customers (4)	8898

are used and propagated to their neighbors. Due to the AS level hierarchy that exists today [23], most of these aggregation points lie in the core of the Internet. We will describe the AS hierarchy that is obtained from the RMAP and analyze the scalability of OPCA in Section VI-B. If OPCA offers enough of a gain over the current routing architecture, we believe that stub ASes will urge their upstream providers to deploy PAs, and such pressure will cascade up to the core of the Internet.

B. Preliminary Results

B.1 RMAP Implementation

We have completed the design of the RMAP as described in Section III-B.5. Table II shows the relationships that we inferred between the 23,935 AS pairs that we extracted from several routing tables [23]. Table III shows the AS hierarchy that we inferred among the 10,915 ASes that were present in the routing tables.

Using our hierarchy characterization, we can also study the growth of multihoming. We use our current list of customer ASes and identify multihomed customer ASes as those making route announcements via multiple upstream ASes. Note that these values are lower bounds as some ASes and links may not be visible from the BGP tables that we use. We use routing tables from many time periods [4] to show the change in multi-

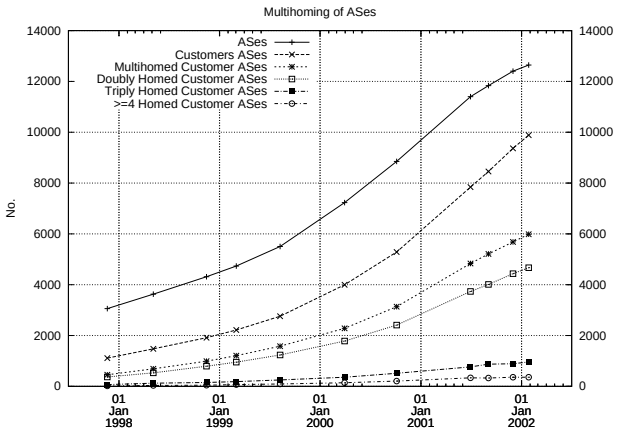

Fig. 6.

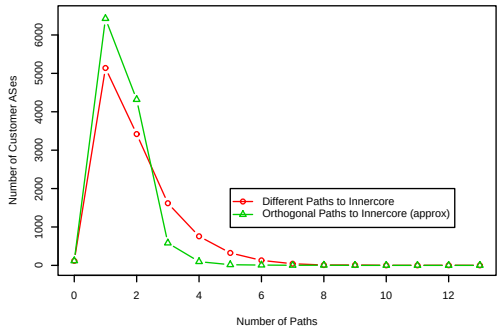

Fig. 7. Orthogonality of Customer AS → Dense Core Paths

homed customer ASes over time. In Figure 6, we show that the large growth in the number of ASes has been fueled by customer ASes, most of which are multihomed.

B.2 Scalability Analysis

We also need to show that the system will scale to the number of ASes that exist today and can accommodate projected future growth based on today's trends. If stub ASes can choose among multiple providers for fail-over, they will want to choose providers that do not also have the same upstream provider. This will produce a set of AS paths are as uncorrelated during failure as possible. In this case, most of the aggregation points where multi-homed paths converge will be in the dense core of the Internet that consists of about 20 large ISPs [23].

If only 20 ASes receive the PA requests from all the stub ASes, then their PAs may not be able to keep up with the request load. OPCA is flexible in that each AS can have multiple PAs, where PA requests are segregated by the requesting AS or IP address range, as shown in Figure 4. This will reduce the load on each PA. However, an additional synchronization step may be needed to coordinate configuration changes at the same BGP router. If a small number of ASes receive most of the PA requests, they may be able to do more intelligent global optimizations across the different requests.

In Figure 7, we attempt to measure the diversity of AS level paths on the Internet. We first use BGP routing tables from October 19, 2002 to generate the AS level hierarchy and inter-AS relationships using our prior work [23]. We then analyze the paths present in the BGP routing tables collected on October 19 from the ASes in Table IV. We split up all the paths that we see in the BGP tables into uphill paths from the customer ASes to the dense core ASes. We do not distinguish between the dense core ASes, so we treat two paths that are identical except for the last dense core AS hop as identical paths. We see a total of 193,778 different paths from any customer AS to the dense core ASes. In Figure 7, the "Different Paths to Dense Core" line shows how many different paths exist from any one customer AS to the dense core. The "Orthogonal Paths to Dense Core" shows an underestimate of the number of paths for each customer AS where only the starting AS is the same and the rest of the path is different before reaching the dense core. We see that for just under half of the customer ASes, only one path to the dense core exists, while for about another half of the customer ASes, approximately two orthogonal paths exist to the dense core.

TABLE IV
TELNET LOOKING GLASS SERVERS

AS #	Name
1	Genuity
50	Oak Ridge National Lab.
210	Utah Education Network
553	Belwue, Germany
852	Telus
1838	AT&T, Cerfnet
3257	Tiscali International
3582	University of Oregon
3741	Internet Solutions, South Africa
3967	Exodus, US
4197	Global Online Japan
5388	Energis Squared
5511	France Telecom
6539	Group Telecom, Canada
7018	AT&T, US
8220	COLT Internet
8709	Exodus, Europe
15290	AT&T, Canada

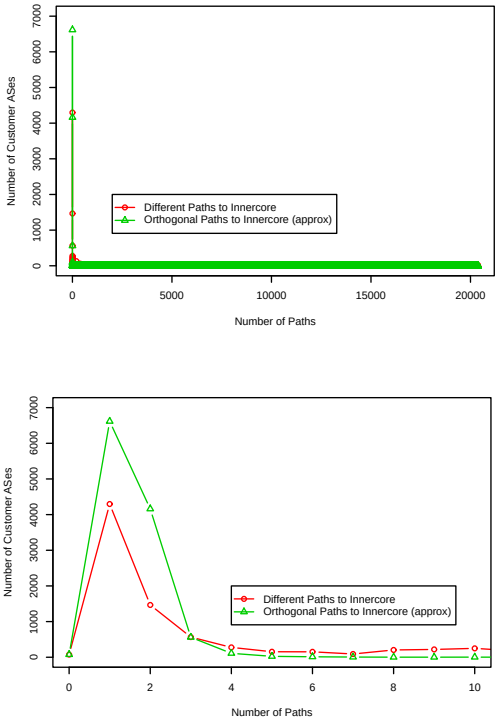

Fig. 8. Enumerated Orthogonality of Customer AS → Dense Core Paths (second graph zooms in on left portion)

Since BGP does not distribute link state, we may be seeing only a small percentage of the possible paths in these routing tables. Using our AS relationship inferences [23] from October 19, we can enumerate all the possible paths from customer ASes to the dense core that do not violate peer-peer or customer-provider export rules. We have enumerated 1,321,470 such paths. We plot the orthogonality of these paths in Figure 8. We still see that just under half of the customer ASes have only a single orthogonal path to the dense core and most of the other half have only two orthogonal paths to the dense core. The graphs exhibit a heavy tail, so there are a few customer ASes with many orthogonal paths to the dense core.

These results indicate how much of a load the ASes in the dense core can expect from customer ASes using the OPP architecture. Few customer ASes will have the need to switch between several BGP paths, and most that do will switch between two paths. We are continuing to examine whether customer ASes have more orthogonal paths to the second tier of ISPs (the transit core) than the number of orthogonal paths all the way to the dense core. However, if OPP provides a significant improvement to the current routing scenario, more customer ASes may multihome in a way to achieve more orthogonal paths to the dense core and the graphs may shift to the right. The architecture should still allow the scaling techniques that we previously described to handle this possible scenario.

C. Proposed Emulation Study

We plan on evaluating our architecture in an emulation testbed. We will use about 50 Linux PCs connected via a high speed LAN. We will run multiple software Zebra BGP routers [27] and PAs on each machine which we can connect over TCP into any arbitrary AS-level topology, extracted from real, sample scenarios we have observed [23]. We will inject the BGP routing tables that we have collected into the software routers. We can then induce faults by shutting down specific routers and measure the convergence time. We plan on using NIST Net [28] to allow us to emulate wide area packet delays. We will use fictitious traffic traces to evaluate OPCA's load balancing capabilities.

One main metric of success is the reduction in the time to fail-over from a primary route to a backup route for a destination prefix. The other metric is the time it takes to shift incoming traffic from one route to another. We will also quantify how effective OPCA can be given the current AS-level structure of the Internet. That is, the current deployment of ASes and inter-AS links will determine how many diverse paths exist for each stub AS and how many upstream aggregation ASes exist. This will also determine how well OPCA will scale.

VII. SUMMARY

We believe that the motivation behind the large increase in multi-homed stub ASes is in achieving fast fail-over and traffic load balancing. The increased participation in BGP also makes more urgent the need to quickly identify and block misconfigured or malicious BGP route announcements. Better connectivity to the Internet may also be obtained if micro-peering is supported for specific address ranges for short periods of time to avoid congestion. BGP today offers slow fail-over, limited control over incoming traffic, little or no automated mechanisms for bogus route detection and prevention and does not support the dynamic change in route and traffic filtering between peers. We address these issues by developing an overlay control architecture that will coexist with and interact with the existing BGP infrastructure. We have explained how the protocol allows our goals to be met and outlined some design decisions that affect deployment, scaling, choice of control path and accuracy. We plan on developing and testing this system in an emulation testbed running software BGP speakers and using real BGP routing data.

ACKNOWLEDGEMENTS

We would like to thank the anonymous reviewers for their detailed and helpful feedback. We would also like to thank Ion Stoica (UC Berkeley), Anthony D. Joseph (UC Berkeley), Yasuhiko Matsunaga (NEC), Supratik Bhattacharyya (Sprint ATL) and Jennifer Rexford (AT&T Research) for their valuable feedback on early versions of this work.

REFERENCES

[1] J. W. Stewart, *BGP4: Inter-Domain Routing in the Internet*. Addison-Wesley, 1998.
[2] T. Bu, L. Gao, and D. Towsley, "On routing table growth," in *Proc. IEEE Global Internet Symposium*, 2002.
[3] G. Huston, "Analyzing the Internet's BGP Routing Table," Cisco Internet Protocol Journal, March 2001.
[4] "University of Oregon RouteViews project." `http://www.routeviews.org/`.
[5] "Route Science Web Page." `http://www.routescience.com/`.
[6] R. Mortimer, "Investment and Cooperation Among Internet Backbone Firms," Ph.D. dissertation, Economics Department, UC Berkeley, 2001.
[7] C. Labovitz, R. Wattenhofer, S. Venkatachary, and A. Ahuja, "The impact of Internet policy and topology on delayed routing convergence," in *Proc. IEEE INFOCOM*, April 2001.
[8] "NANOG: North America Network Operators Group mailing list." `http://www.merit.edu/mail.archives/nanog/`.
[9] G. Huston, "Architectural requirements for inter-domain routing in the Internet," Internet Draft 01, Internet Architecture Board, May 2001.
[10] G. Huston, "NOPEER community for route scope control," Internet Draft 00, August 2001.
[11] O. Bonaventure, S. Cnodder, J. Haas, B. Quoitin, and R. White, "Controlling the redistribution of BGP routes," Internet Draft 02, February 2002.
[12] T. Hardie and R. White, "Bounding longest match considered," Internet Draft 02, November 2001.
[13] D. Pei, X. Zhao, L. Wang, D. Massey, A. Mankin, S. F. Wu, and L. Zhang, "Improving BGP convergence through consistency assertions," in *Proc. IEEE INFOCOM*, 2002.
[14] R. M. Mortier, "Multi-timescale internet traffic engineering," in *IEEE Communication Magazine*, October 2002.
[15] D. Estrin, J. Postel, and Y. Rekhter, "Routing arbiter architecture," 1994.
[16] D. G. Andersen, H. Balakrishnan, M. F. Kaashoek, and R. Morris, "Resilient overlay networks," in *Proc. ACM SOSP*, October 2001.
[17] I. Castineyra, N. Chiappa, and M. Steenstrup, "The Nimrod Routing Architecture," RFC 1992, Network Working Group, August 1996.
[18] S. K. et al, "BANANAS: A new connectionless traffic engineering framework for the internet," RPI Technical Report, 2002.
[19] R. Neilson, J. Wheeler, F. Reichmeyer, and S. Hares, "A discussion of bandwidth broker requirements for Internet2 Qbone deployment."
[20] R. Yavatkar, D. Pendarakis, and R. Guerlin, "A framework for policy-based admission control," RFC 2753, IETF, January 2000.
[21] D. Durham, J. Boyle, R. Cohen, S. Herzog, R. Rajan, and A. Sastry, "The COPS (COmmon Open Policy Service) Protocol," RFC 2748, IETF, January 2000.
[22] D. O. Awduche, A. Chiu, A. Elqalid, I. Widjaja, and X. Xiao, "A Framework for Internet Traffic Engineering," draft 2, IETF, 2000.
[23] L. Subramanian, S. Agarwal, J. Rexford, and R. H. Katz, "Characterizing the Internet hierarchy from multiple vantage points," *Proc. IEEE INFOCOM*, 2002.
[24] B. Krishnamurthy and J. Wang, "On network-aware clustering of web clients," in *Proc. ACM SIGCOMM*, August 2000.
[25] S. K. Charles, "Secure Border Gateway Protocol (S-BGP) — Real World Performance and Deployment Issues."
[26] R. Mortier and I. Pratt, "Efficient network routeing," unpublished project proposal, 2000.
[27] "GNU Zebra, free routing software." $http://www.zebra.org/$.
[28] "NIST Net Emulation Web Site." `http://snad.ncsl.nist.gov/itg/nistnet/`.