MIRO: Multi-path Interdomain ROuting

Wen Xu and Jennifer Rexford
Department of Computer Science
Princeton University

ABSTRACT

The Internet consists of thousands of independent domains with different, and sometimes competing, business interests. However, the current interdomain routing protocol (BGP) limits each router to using a single route for each destination prefix, which may not satisfy the diverse requirements of end users. Recent proposals for source routing offer an alternative where end hosts or edge routers select the end-to-end paths. However, source routing leaves transit domains with very little control and introduces difficult scalability and security challenges. In this paper, we present a multi-path interdomain routing protocol called MIRO that offers substantial flexibility, while giving transit domains control over the flow of traffic through their infrastructure and avoiding state explosion in disseminating reachability information. In MIRO, routers learn default routes through the existing BGP protocol, and arbitrary pairs of domains can negotiate the use of additional paths (bound to tunnels in the data plane) tailored to their special needs. MIRO retains the simplicity of BGP for most traffic, and remains backwards compatible with BGP to allow for incremental deployability. Experiments with Internet topology and routing data illustrate that MIRO offers tremendous flexibility for path selection with reasonable overhead.

Categories and Subject Descriptors:
C.2.6 [Communication Networks]: Internetworking

General Terms: Design, Experimentation.

Keywords: BGP, flexibility, inter-domain routing, multipath routing, scalability.

1. INTRODUCTION

The Internet consists of thousands of independently administered domains (or Autonomous Systems) that rely on the Border Gateway Protocol (BGP) to learn how to reach remote destinations. Although BGP allows ASes to apply a wide range of routing policies, the protocol requires each router to select a single "best" route for each destination prefix from the routes advertised by its neighbors. This leaves many ASes with little control over the paths their traffic takes. For example, an AS might want to avoid paths traversing an AS known to have bad performance or filter data packets based on their contents. This is the situation in Figure 1, where thick lines represent the paths chosen to reach AS F. AS A does not want AS E to carry its traffic, but it has no choice because B and D have both selected paths through E. Simply asking B to switch to the route BCF is not an attractive solution, since this would not allow AS B and its other neighbors to continue using BEF.

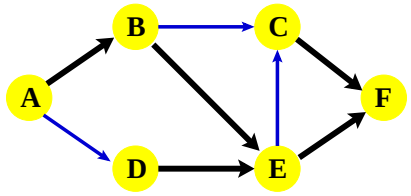


Figure 1: Single-path routing to AS F

Recent research has considered several alternatives to single-path routing, including source routing and overlay networks. In source routing, an end user or AS picks the entire path the packets traverse [1–5]. In overlay networks, packets can travel through intermediate hosts to avoid performance or reliability problems on the direct path [6]. However, these techniques do not give transit ASes, such as Internet Service Providers (ISPs), much control over the traffic traversing their networks. This control is important for ASes to engineer their networks to run efficiently, and to maximize revenue. The lack of control for ISPs is a significant impediment to the eventual adoption of source routing. In addition, both source routing and overlay networks may not scale to a network the size of the Internet. Instead, we explore an alternative solution where the interdomain routing protocol supports multi-path routing, while providing flexible control for transit ASes and avoiding state explosion in disseminating routing information.

Our solution is motivated by several observations about today's interdomain-routing system:

- Having each router select and advertise a single route for each prefix is not flexible enough to satisfy the diverse performance and security requirements. In Figure 1, today's routing system does not enable AS A to circumvent AS E in sending traffic to AS F.
- The existing routes chosen by BGP are sufficient for a large portion of the traffic. In Figure 1, AS B and its other customers may be perfectly happy with the path BEF.
- End users need control over the *properties* of the end-to-end path, rather than complete control over which path is taken. In Figure 1, AS A only wants to avoid AS E and does not care about the rest of the path.

SIGCOMM'06, September 11–15, 2006, Pisa, Italy.

- The existing BGP protocol already provides many candidate routes, although the alternate routes are not disseminated. In Figure 1, AS B has learned the route BCF but simply has not announced it to AS A.

- An AS selects routes based on business relationships with neighboring domains, but might be willing to direct traffic to other paths, for a price. In Figure 1, AS B may prefer BEF for financial reasons, but may be willing to send AS A's traffic over BCF.

- Today's Internet provides limited methods for one AS to influence another AS's choice. For example, if AS F is a multi-homed stub AS which wants to control how much incoming traffic traverse link CF and EF respectively, it can only advertise smaller prefixes or prepend its AS number [7]. However those methods may be easily nullified by other ASes' local policy, making their effectiveness limited.

Inspired by these observations, we propose a multi-path interdomain routing protocol, called MIRO, with the following features:

- **AS-level path selection:** An AS represents an institution, such as a university or company, and business relationships are easily defined at the AS level. This is simpler and more scalable than giving each end user fine-grain control over path selection.

- **Negotiation for alternate routes:** An AS learns one route from each neighbor and negotiates to learn alternate routes as needed. This leads to a scalable solution that is backwards compatible with BGP, and it also allows policy interaction between arbitrary pairs of ASes.

- **Policy-driven export of alternate routes:** The responding AS in the negotiation has control over which alternate paths, if any, it announces in each step of the negotiation. This gives transit ASes control over the traffic entering their networks.

- **Tunnels to direct traffic on alternate paths:** After a successful negotiation, the two ASes establish the state needed to forward data traffic on the alternate route. The remaining traffic traverses the default route in the forwarding tables.

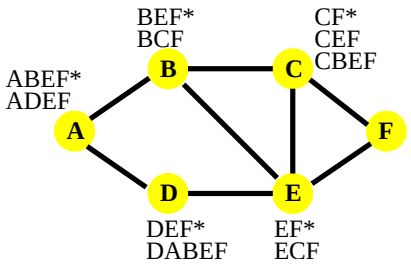
With the additional flexibility, ASes could choose paths that satisfy their special needs, for example:

- *Avoiding a specific AS for security or performance reasons:* An AS can avoid sending sensitive data through a hostile country or avoid an AS that often drops packets.

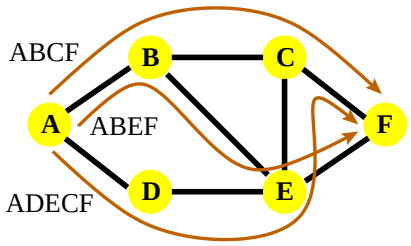
- *Achieving higher performance:* The AS can send traffic through more expensive inter-AS links that are normally not available, to achieve lower latency or higher bandwidth.

- *Load balancing for incoming traffic:* A multi-homed AS trying to balance load over multiple incoming links can request that some upstream ASes use special AS paths to direct traffic over a different incoming link[1].

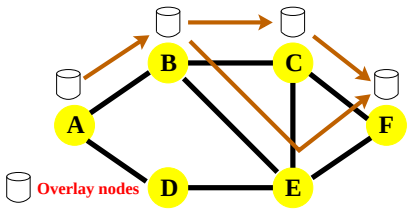
[1]Analysis of RouteViews data [8] shows that 60% of the 20,000 ASes are multi-homed and more than 2000 are announcing smaller subnets into BGP to exert control over incoming traffic. However, announcing small subnets increases routing-table size without providing precise control.



(a) current BGP protocol



(b) source routing



(c) overlay networks

Figure 2: Inter-AS routing proposals
* represents chosen route.

In designing MIRO, we separate policy and mechanism wherever possible, to support a wide range of policies for interdomain routing. Still, we present example policies and useful policy guidelines to illustrate the benefits of adopting our protocol. In the next section, we present background material on existing routing architectures. Then, Section 3 gives an overview of our main design decisions. We describe MIRO in greater detail in Section 4 and demonstrate the effectiveness and efficiency of MIRO in Section 5 using measurement data from RouteViews [8]. In Section 6 we discuss how ASes can configure flexible routing policies. Section 7 discusses additional technical issues, such as routing-protocol convergence and route aggregations. Section 8 presents related work, and the paper concludes in Section 9.

2. ROUTING ARCHITECTURES

In this section, we present an overview of the current BGP protocol, source routing, and overlay networks. To simplify the discussion, we represent each AS as a single router, as illustrated in Figure 2 where five ASes are selecting routes to a destination in AS F. In BGP, each AS selects a single best route (indicated by an asterisk) and advertises it to all neighbors. In source routing, each end host has complete knowledge of the entire topology and can choose whatever paths it wishes. In overlay networks, several overlay nodes connect to the physical network to form a virtual topology; each node can direct traffic through other overlay nodes en route to the destination.

2.1 Today's Interdomain Routing

BGP [9], the de facto interdomain routing protocol for the Internet, has several features that limit flexibility in path selection:

- *Destination-based*: BGP distributes reachability information about address blocks, and each router forwards a packet by performing a longest-prefix match on the destination address. As such, packets from different sources going through the same router would follow the same downstream path.
- *Single-path routing*: A router learns at most one BGP route from each neighbor and must select and advertise a single "best" route. This limits the number of paths advertised and poses severe restrictions on flexibility.
- *Path-vector protocol*: In contrast to link-state protocols that flood topology information, BGP is a path-vector protocol where routers learn only the AS paths advertised by their neighbors. This improves scalability at the expense of visibility into the possible paths.
- *Local-policy based*: BGP gives each AS significant flexibility in deciding which routes to select and export. However, the available routes depend on the composition of the local policies in the downstream ASes, limiting the control each AS has over path selection.

The local policies for selecting and exporting BGP routes depend on the business relationships between neighboring ASes. The most common relationships are customer-provider, peer, and sibling [10–12]. In a customer-provider relationship, the customer normally pays the provider for transit service; as such, the provider announces the routes learned from any customer to all neighboring ASes, but the customer normally only advertises the routes learned from its provider to its own customers. In a peer-peer relationship, two ASes find it mutually beneficial to carry traffic between each other's customers, often free of charge. Peering agreements often indicate that the routes learned from a peer can only be advertised to customers. Sibling ASes typically belong to the same institution, such as a large ISP, and provide transit service to each other. Upon learning routes for a prefix from multiple neighbors, an AS typically prefers to use customer-learned routes, then siblings, then peers, and finally providers, to maximize revenue. At times, though, providers deviate from these policy conventions upon customer request (e.g., to provide backup connectivity for customers). We believe that business incentives could also motivate an AS to make alternate routes available to neighbors who have special performance or security requirements.

Another problem in BGP is that an AS has limited influence over the local policies in other ASes. Each AS prefers some paths over others based on its own local goals. In some cases, an AS allows its customers to influence these preferences by "tagging" the BGP announcements. However, these techniques are usually applied only between adjacent ASes that unconditionally trust one another (e.g., a stub AS and its upstream ISP). In addition, the underlying mechanism is quite primitive—a simple tagging of routes without any kind of "back and forth" negotiation between the two ASes.

2.2 Source Routing

In the past few years, several researchers have proposed source routing as a way to provide greater flexibility in path selection [1–5]. In source routing, the end hosts or edge routers select the end-to-end paths to the destinations. The data packets carry a list of the hops in the path, or flow identifiers that indicate how intermediate routers should direct the traffic. Although source routing maximizes flexibility, several difficult challenges remain:

- *Limited control for intermediate ASes:* Under source routing, intermediate ASes have very little control over how traffic enters and leaves their networks. This makes it difficult for intermediate ASes to engineer their networks and select routes based on their own business goals, which is a barrier to the deployment of source-routing schemes.
- *Scalability:* Source routing depends on knowledge of the network topology, at some level of detail, for sources to compute the paths. The volume of topology data, and the overhead for computing paths, would be high, unless the data are aggregated; including load or performance metrics, if necessary, would further increase the overhead. In addition, the sources must receive new topology information quickly when link or router failures make the old paths invalid.
- *Efficiency and stability:* In source routing, end hosts or edge routers adapt path selection based on application requirements and feedback about the state of the network. Although source routing can generate good solutions in some cases [13], a large number of selfish sources selecting paths at the same time may lead to suboptimal outcomes, or even instability.

Even if these challenges prove to be surmountable in practice, we believe that it is valuable to consider other approaches that make different trade-offs between flexibility for the sources, control for the intermediate ASes, and scalability of the overall system.

2.3 Overlay Networks

In overlay networks, several end hosts form a virtual topology on top of the existing Internet [6]. When the direct path through the underlying network has performance or reliability problems, the sending node can direct traffic through an intermediate node. The traffic then travels on the path from the source to the intermediate node, followed by the path from the intermediate node to the destination. Although overlay networks are useful for circumventing problems along the direct path, they are not a panacea for supporting flexible path selection at scale, for several reasons:

- *Data-plane overhead:* Sending traffic through an intermediate host increases latency, and consumes bandwidth on the edge link in and out of that host. In addition, the data packets must be encapsulated to direct traffic through the host, which consumes extra bandwidth in the underlying network.
- *Limited control:* The overlay network has no control over the paths between the nodes, and has limited visibility into the properties of these paths. These paths depend on the underlying network topology, as well as the policies of the various ASes in the network.
- *Probing overhead:* To compensate for poor visibility into the underlying network, overlay networks normally rely on aggressive probing to infer properties of the paths between nodes. Probing has inherent inaccuracies and does not scale well to large deployments.

In contrast to source routing, overlay networks do not require support from the routers or consent from the ASes in the underlying network. Although overlays undoubtedly have an important role to play in enabling new services and adapting to application requirements, we believe the underlying network should have native support for more flexible path selection to support diverse performance and security requirements efficiently, and at scale.

3. MIRO PROTOCOL DESIGN

To provide greater flexibility in path selection, we propose extending BGP into a multi-path routing protocol, while keeping the goals of scalability, control for intermediate ASes, and backwards compatibility in mind. In this section, we present the key features of MIRO: AS-level path-vector routing for scalability, pull-based route retrieval for backwards compatibility and scalability, bilateral negotiation between ASes to contain complexity, selective export of extra routes for scalability and to give control to intermediate ASes, and tunneling in the data plane to direct packets along the chosen routes. For simplicity, we treat each AS as a single node and defer the technical details of MIRO until Section 4.

3.1 AS-Level Path-Vector Protocol

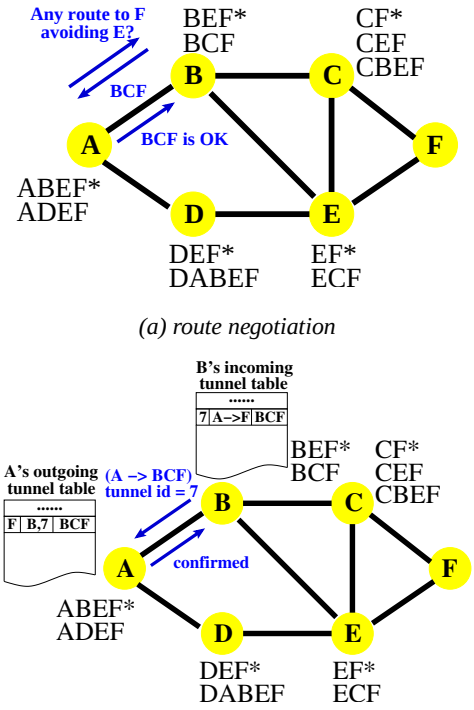
MIRO represents paths at the AS level—as in today's BGP, each AS adds its AS number to the AS-path attribute before propagating a route announcement to a neighboring domain. Although AS-level path selection seems natural for an interdomain routing protocol, other options exist. For example, some source-routing proposals suggest that all routers in the Internet be exposed to allow link-level path selection. However, we argue that link-level path selection exposes too much of the internals of intermediate ASes and limits their control over the flow of traffic. In addition, supporting link-level path selection would require the protocol to propagate a large amount of state, and to update this state when internal topology changes occur.

We argue that routing at the AS level is the right choice. First, each AS is owned and managed by a single authority, making the AS a natural entity of trust and policy specification. Second, routing at the AS level is more scalable than at the link level, and each AS can hide its internal structure and adjust the flow of traffic without affecting the AS path. Third, because business contracts are often signed by authorities rather than individual users, it is easier to verify that the performance and reliability of a route conforms to an AS-level contract. Although some recent papers consider grouping related ASes and routing packets at the AS-group level [1, 14], we advocate keeping the AS as the base granularity of path selection for simplicity. In MIRO, groups of related ASes can cooperate by revealing extra paths to other ASes inside the same group.

3.2 Pull-based Route Retrieval

Many ASes and end users are satisfied with the default routes provided by BGP. Having each AS propagate alternate routes to every neighbor would severely limit the scalability of interdomain routing, and would also force all ASes to deploy the new protocol. Instead of pushing extra routes to all neighbors, MIRO has ASes actively solicit alternate routes only when needed. For example, in Figure 3, AS A is the only AS that is unsatisfied with its default route (ABEF). As a result, AS A asks AS B to advertise alternative routes, possibly including a routing policy (e.g., "avoid routes traversing AS E") in the request. All other ASes simply use their default routes and incur no additional overhead.

ASes that have not deployed our multi-path extensions to BGP can continue to use today's version of the protocol. For example, ASes C and F do not need to run the enhanced protocol for AS A to be able to query AS B for extra routing options. Each AS can decide on its own whether to deploy the enhanced protocol so that a value-added service could be offered to others, such as its customers. In the evaluation section, we show that even a modest deployment of MIRO by a few tier-1 and tier-2 ISPs would be sufficient to expose much of the underlying path diversity in today's AS-level topology, making it possible for early adopters to enjoy significant gains. This can encourage other ISPs to deploy the protocol in order to compete effectively with these early adopters in providing value-added services to their customers.



(a) route negotiation

(b) routing tunnel establishment

Figure 3: Multi-path Routing Example

3.3 Bilateral Negotiation Between ASes

MIRO is based on bilateral negotiation between ASes, where one AS asks another to advertise alternate routes. Bilateral negotiation simplifies the protocol, and it reflects the fact that AS business relationships are often bilateral anyway. In Figure 3, negotiating with AS B would be sufficient for AS A to learn a path to F that circumvents E. In bilateral negotiations, we refer to the AS initiating the negotiation as the *requesting AS* and the other AS as the *responding AS*. The AS closer to the packet source is the *upstream AS* and the one closer to packet destination is the *downstream AS*. In the example in Figure 3, AS A is the requesting AS and the upstream AS, and AS B is the responding AS and the downstream AS.

Although we focus on bilateral negotiations, an AS can easily approximate multi-party negotiation by making requests to two ASes. In Figure 3, AS A may ask several ASes (e.g. B and D) to advertise additional paths, with the goal of discovering paths that avoid traversing AS E. Also, in responding to a request, an AS may provide additional paths obtained from another negotiation as new candidates. For example, AS B might query AS C to advertise alternate paths as part of satisfying the request from AS A, if C were not already announcing a path that avoids AS E. Still, we do not envision that multi-hop negotiation would need to take place very often, since most paths through the Internet are short, typically traversing four AS hops or less.

In the simplest case, an AS negotiates with an immediate neighbor, as in the example where AS A negotiates with B or D. Allowing negotiation with other ASes provides much greater flexibility, especially when the adjacent ASes have not deployed the new multi-path routing protocol. For example, suppose ASes B and D

have not (yet) deployed the new protocol. AS A could conceivably negotiate with C to learn the path CF, using the path ABC through B to direct packets to C, which would then direct the packets onward toward F. In directing traffic through an intermediate AS, MIRO is similar to overlay networks, though we envision the routers in the intermediate AS would support this functionality directly, rather than requiring data packets to traverse an intermediate host.

An AS can adopt flexible policies for deciding who to negotiate with. For example, an aggressive AS trying to achieve high performance might decide to query all immediate neighbors and 2-step away neighbors, another AS trying to avoid an insecure AS might consult a public Internet topology graph and exclude some ASes that will never have valid paths (e.g., those that are single-homed to the insecure AS). MIRO classifies this as a policy issue and leaves the decision to individual ASes and their configured policies.

Although Figure 3 shows an example where the requesting AS is the upstream AS, downstream ASes may also initiate requests. For example, suppose the link EF in Figure 3(a) is overloaded with traffic sent by ASes A, B, D, and E to AS F. To reduce the load on link EF, AS F could request one of more of these ASes to divert traffic to a path that traverses the link CF. For example, AS F could negotiate with AS B to consider switching to an alternate path that traverses CF. AS B could respond by agreeing to select the path BCF instead of BEF, and advertising the path BCF to its neighbors.

3.4 Selective Export of Extra Routes

Upon receiving a request, the responding AS could conceivably propagate all known alternate routes to the requesting AS. However, announcing all of the routes might incur significant overhead. In addition, the responding AS might not view all routes as equally appealing. As such, we envision that the responding AS could apply routing policies that control which alternate routes are announced, and potentially tag these routes with preference or pricing information to influence the routing decisions in the requesting AS. For example, suppose AS C has a customer (not shown) that wants to avoid the link CF. Rather than offering both CEF and CBEF as alternate routes, AS C might announce only CEF (e.g., if sending traffic via AS B incurs a significant financial cost), or tag the CBEF route with pricing information.

We envision that the policies for exporting alternate routes would depend on the business relationships between neighboring ASes. For example, suppose an AS has selected a route learned from a customer AS but has also learned another route for the same destination from another customer. The AS may be willing to advertise all customer-learned routes but not routes learned from peers or providers. Alternatively, the AS may be willing to advertise all routes with the same (highest) local-preference value, or advertise other (less preferred) routes only to neighbors that subscribe to a premium service. These kinds of policies are readily expressed using the same kinds of "route map" constructs commonly used in BGP import and export policies today [15].

3.5 Tunnels for Forwarding Data Packets

Under multi-path routing, the routers cannot forward packets based on the destination IP address alone. Instead, routers must be able to forward the packets along the paths chosen by the upstream ASes. In MIRO, the two negotiating ASes establish a *tunnel* for carrying the data packets. The downstream AS provides a unique tunnel identifier to the upstream AS, independent of which AS initiated the negotiation. In Figure 3(b), when AS A and AS B agree on the alternate route BCF, AS B assigns a tunnel id of 7 and sends the id to AS A. In the data plane, AS A directs the packets into the tunnel and AS B removes the packets from the tunnel and forwards them across the link BC. Then, AS C forwards the packets based on the destination IP address along the default path to AS F. We consider several ways to encapsulate the data packets as they enter the tunnel, as discussed in more detail in Section 4.2.

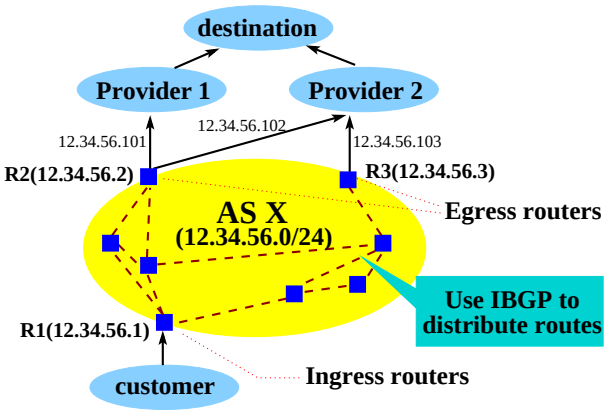


Figure 4: Intra-AS routing architecture

The upstream AS need not direct *all* packets into tunnels. Rather, the AS may apply local policies to direct some traffic along alternate paths and send the remaining packets along the default path (i.e., using conventional destination-based forwarding). In Figure 3, suppose BCF has lower latency than BEF. Then, AS A may want to direct its real-time traffic via BCF while sending best-effort traffic along BEF, especially if AS B charges for using alternate routes. The upstream AS could implement these traffic-splitting policies by installing classifiers that match packets based on header fields (e.g., IP addresses, port numbers, and type-of-service bits). An AS may also split the traffic to balance load across multiple paths. The AS could direct a fraction of the traffic along each of the paths by applying a hash function that maps each traffic flow (e.g., packets with the same addresses and port numbers) to a path, as in prior work on multi-path forwarding within an AS [16].

4. MIRO IMPLEMENTATION

Despite the conceptual appeal of viewing each AS as a single node, ASes often have multiple routers that participate in the interdomain routing protocol. In this section, we describe how to implement MIRO across a collection of routers in an AS. Then, we present several practical methods for encapsulating packets and identifying the end-points of tunnels in the data plane. Finally the control-plane design is presented.

4.1 Intra-AS Architecture

A large AS typically has multiple edge routers that exchange BGP routing information with neighboring domains, as illustrated in Figure 4. Data packets from the customer enter AS X at the ingress router R1 and traverse several internal routers before leaving the network at an egress router, such as R2 or R3. Although BGP is a single-path protocol, these routers do not necessarily select the same interdomain route to the destination (e.g., R2 and R3 might route via Provider 1 and 2, respectively). Typically, large ASes use internal BGP (iBGP) to distribute routing information to other routers; for example, R1 in Figure 4 might learn BGP routes from both R2 and R3. Even if both R2 and R3 select a BGP path through Provider 2, MIRO would allow the customer to learn the alternate route through Provider 1, upon request. AS X can provide these extra routes even if the two providers do not run MIRO.

An implementation of MIRO must install the appropriate data-plane states in both AS X and the customer network. If the customer requests and selects an alternate route, AS X needs to provide a tunnel identifier that the customer can use in encapsulating data packets and directing them through the appropriate egress point. In addition, AS X needs to ensure that the packets, upon reaching router R2, are decapsulated and forwarded via the egress link to Provider 1, even if R2 normally forwards packets via Provider 2 to reach this destination. That is, R2 needs to decapsulate the packet and still forward the packet based on the tunnel identifier[2]. The customer, in turn, must install the necessary state to ensure that packets entering the network are diverted to the appropriate tunnel. This may require the customer AS to install data-plane state at multiple ingress routers where the data packets may arrive.

Providing alternate routes to the customer requires coordination amongst the routers in AS X. By default, R2 would not announce the alternate route (learned from Provider 1) to R1 via iBGP. We envision two main ways to implement the control protocol. First, the customer may request alternate routes from R1 which, in turn, requests alternate routes from its iBGP neighbors R2 and R3. If the client selects the route, R1 would propagate the tunnel identifier and instruct R2 to install the necessary data-plane state for decapsulating and forwarding the packets as they leave the tunnel on their way to Provider 1. Second, a separate service, such as the Routing Control Platform (RCP) [18], could manage the interdomain routing information on behalf of the routers. In this approach, the RCP would exchange interdomain routing information with neighboring domains and compute BGP paths on behalf of the routers. The RCP in AS X would handle the requests from the customer's RCP for alternate routes to reach the destination. The RCP could also install the data-plane state, such as tunneling tables or packet classifiers, in the routers to direct traffic along the chosen paths.

4.2 Data Plane Packet Encapsulation

Although a variety of tunneling techniques exist, we focus our discussion on IP-in-IP encapsulation. In this approach, the response from the downstream AS includes an IP address corresponding to the egress point of the tunnel. To divert a packet into the tunnel, the upstream AS encapsulates the IP packet destined to this IP address. MIRO must ensure that the upstream AS knows how to reach this IP address, even if the downstream AS is several AS hops away. In addition, we need to determine which IP address MIRO should use, and ensure that the egress router is equipped to decapsulate the packets and direct them to the next AS in the path. We have identified two main options for which IP address the downstream AS should provide, with different advantages and disadvantages:

IP Address of the Egress Links or Egress Routers: When the IP address of egress links are used, the downstream AS first labels each egress link with a different reserved IP address, then advertises these addresses to the upstream AS. For example, in Figure 4, the links R2→provider 1, R2→provider 2, and R3→provider 2 are given IP addresses 12.34.56.101, 12.34.56.102, and 12.34.56.103, respectively, then 12.34.56.102 and 12.34.56.103 are advertised to the upstream if provider 2 is the selected next-hop AS. Since the IP address uniquely identifies the egress link, the packet does not need to carry any separate identifier for the tunnel. Alternatively, the downstream AS can advertise the IP addresses of egress routers. Because there are fewer egress routers than egress links, this would consume fewer IP addresses, but requires the data packets to carry a separate tunnel identifier so the egress router knows which egress link to use. For example, AS X in Figure 4 could advertise 12.34.56.2 and 12.34.56.3 if provider 2 is the next hop AS, and advertise 12.34.56.3 if provider 1 is selected instead. R2 would check tunnel id to see if link to provider 1 or that to provider 2 should be picked.

One Reserved IP Address for All Tunnels: The downstream AS reserves one special IP address for all routing tunnels. At each ingress router, the packet destined to this special IP address is replaced with the correct egress router IP. For example, AS X in Figure 4 chooses 12.34.56.100 as the special IP and that IP is the destination for any packet belonging to a tunnel in X. Also, each ingress router grabs a mapping table of (tunnel_id, set of egress router IP), for example, (tunnel 7, {12.34.56.2, 12.34.56.3}) will be installed on R1 if tunnel 7 uses the AS X→provider 2→destination route. Then R1 learns from the intradomain routing protocol that R2 is the closest one in the set, therefore R1 sets 12.34.56.2 as the chosen IP. When R1 sees a packet destined to 12.34.56.100, it checks the tunnel id in the packet, finds that the id is 7, then retrieves 12.34.56.2 from its lookup table. Finally R1 replaces 12.34.56.100 with 12.34.56.2 and forwards the packet to R2.

By using one IP address for all tunnels, the downstream AS does not reveal any internal topology information to the upstream. Therefore the ingress routers in the downstream AS could freely adjust which egress router or link they use. However, this method requires packet rewriting and therefore data-plane modifications at all ingress routers. In contrast, by exposing IP addresses corresponding to egress routers or egress links, the internal topology is partially exposed to the upstream, so changes in internal topology might lead to tunnel destruction or ineffective packet delivery. Moreover, it poses security challenges as anyone could send packets to these addresses. Advanced packet filters or network capabilities [19] could be used to prevent this problem.

4.3 Control Plane Tunnel Management

The control plane manages the creation and destruction of tunnels, based on negotiations between pairs of ASes. Figure 3 in Section 3 presents an example where AS A launches a request to AS B, specifying the destination prefix and (optionally) the desired properties of the alternate routes. Upon receiving the request, AS B advertises the subset of candidate routes that are consistent with its own local policy. Then, AS A selects a candidate route and performs a handshake with AS B to trigger creation of the tunnel. AS B replies with a tunnel identifier (represented by the number "7" in the figure), or the IP address of the tunnel end-point, and the ASes update tunnel tables accordingly.

A tunnel remains active until one AS tears it down, either actively or passively. AS A will tear down the tunnel if the path AB changes (e.g., to traverse an intermediate AS) or fails, and AS B will tear down the tunnel if the path BCF to the destination prefix fails. The ASes can observe these changes in the BGP update messages or session failures. However, when A can no longer reach B at all, the "active tunnel tear-down" message itself may not be able to reach AS B any longer. To avoid leaving idle tunnels in the downstream ASes, AS A and B should adopt a soft-state protocol, where they exchange "keep-alive" messages in the MIRO control plane, and destroy tunnels when the heartbeat timer expires. These "keep-alive" messages could be directed to a specialized central server (such as the RCP) in each AS; that server will monitor the health for all routing tunnels and actively tear down unused ones.

5. PERFORMANCE EVALUATION

In this section, we evaluate the effectiveness of MIRO based on an AS-level topology, annotated with the business relationships between neighboring ASes. After describing our evaluation method-

[2]This functionality, known as "penultimate hop popping," is implemented in Multi-Protocol Label Switching (MPLS) [17], a tunneling technology deployed in many backbone networks.

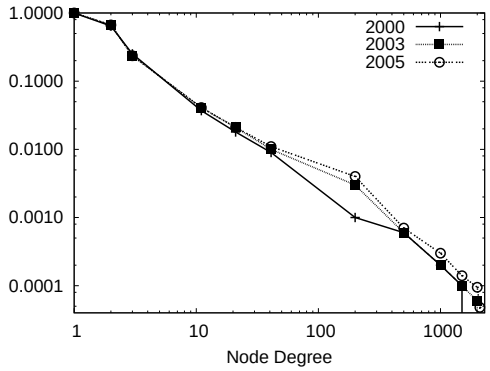


Figure 5: Node distribution

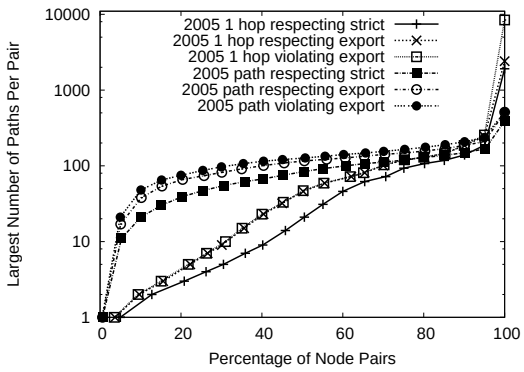


Figure 6: Number of available routes

ology, we show that MIRO could expose much of the path diversity in the AS-level topology. However, demonstrating whether MIRO provides *enough* flexibility requires evaluating the protocol with a particular policy objective in mind. We focus most of our evaluation on the scenario where the source AS wishes to avoid a particular intermediate AS for security or performance reasons. We use these experiments to demonstrate that MIRO is flexible and efficient, and offers substantial benefits to early adopters. We also briefly consider a second application where a multi-homed stub AS needs to negotiate with upstream ASes to balance load across multiple incoming links.

5.1 Evaluation Methodology

Ideally, we would evaluate MIRO by deploying the new protocol in the Internet and measuring the results. Since this is not possible, we simulate MIRO operating in an environment as close to the current Internet as possible. Evaluating on streams of BGP update messages is not sufficient, both because of the limited number of data feeds available and the need to know what routing policies to model. Instead, we evaluate MIRO on the AS-level topology, assuming that each AS selects and exports routes based on the business relationships with its neighbors [20]. The local preferences of the routes are decided solely based on AS relationships, and each AS is treated as one node.

We draw on the results of previous work on inferring AS relationships [11, 12], applied to the BGP tables provided by RouteViews [8]. Invariably, RouteViews does not provide a complete view of the AS-level topology, and even the best inference algorithms are imperfect, but we believe this is the most appropriate way to evaluate the effectiveness of MIRO under realistic configurations. Our main results depend primarily on the typical AS-path lengths and the small number of high-degree nodes, which are viewed as fundamental properties of the AS-level topology. As such, we believe our main conclusions still hold, despite the imperfections in the measurement data.

We evaluate MIRO under three instances of the AS-level topology, from 2000, 2003, and 2005, to study the effects of the increasing size and connectivity of the Internet on multi-path routing. To infer the relationships between ASes, we apply the algorithms presented by Gao [11] and Agarwal [12], but only present results for the algorithm in [11] due to space limitation; a previous study suggested that the Gao algorithm produces more accurate inference results [21]. Our experiments with the Agarwal inferences show similar trends. The key characteristics of the AS topology and business relationships are summarized in Table 1. Figure 5 plots the distribution of node degrees for the three years for the Gao algorithm. The graph is consistent with previous studies that show a wide variance in node degrees, where a small number of nodes have a large number of neighbors; these nodes correspond to the tier-1 ASes that form the core of the Internet.

After inferring AS relationships, we apply conventional policies for selecting and exporting routes to construct routing tables, where each AS originates a single prefix. This represents the base scenario of single-path routing based on the existing BGP protocol. To evaluate MIRO, we consider three variations on how a responding AS decides which alternate routes to announce upon request:

- *Strict Policy (/s)*: The responding AS only announces alternate routes with the same local preference as the original default route. For example, if an AS originally advertises a peer-learned route to its neighbors, the AS would not announce any alternate routes learned from a provider. We assume that the AS follows conventional export policies. For example, an AS would not announce a route learned from one peer to another peer.

- *Respect Export Policy (/e)*: The responding AS announces all alternate routes that are consistent with the export policy. For example, an AS would announce all alternate routes to its customers, and all customer-learned routes to its peers and providers.

- *Most Flexible Policy (/a)*: The responding AS announces all alternate routes to any neighbor, independent of the business relationships.

The last scenario, though arguably unreasonable in practice, provides a basis for evaluating how well MIRO is able to expose the underlying path diversity in the Internet.

5.2 Exposing the Underlying Path Diversity

In our first experiment, we measure the path diversity under the three scenarios, and compare with conventional BGP and source routing. We first compute the number of candidate routes between each (source, destination) AS pair, and then sort the totals and plot the distribution in Figure 6. The graph shows results for two scenarios: (i) each source AS negotiates with any of its immediate neighbors (i.e., the "1-hop" set) and (ii) each source AS negotiates with any ASes on the default BGP path to destination (i.e., the "path" set).

Of the 300 million (source, destination) AS pairs we analyzed, only 5% have no alternate paths in the worst case (i.e., the (5%, 1) point on the "1-hop strict policy" curve). The number of paths

Name	Date	# of Nodes	# of Edges	P/C links	Peering links	Sibling links
Gao 2000	10/1/2000	8829	17793	16531	1031	231
Gao 2003	10/8/2003	16130	34231	30649	3062	520
Gao 2005	10/8/2005	20930	44998	40558	3753	687

Table 1: Attributes of the data sets

Name	Single	Multi/s	Multi/e	Multi/a	Source
2000	27.8%	65.4%	72.9%	75.3%	89.5%
2003	31.2%	67.0%	74.6%	76.6%	90.4%
2005	29.5%	67.8%	73.7%	76.0%	91.1%

Table 2: Comparing the routing policies

grows exponentially in the "path" curves, and it increases pretty quickly and stays relatively flat in the "1-hop" curves. For both sets of data, more than half of the AS pairs can find at least tens of alternate paths, and a quarter of the AS pairs have at least one hundred alternate paths. Moreover, the "respect export policy" and the "most flexible policy" curves are similar for both sets of data, meaning that we can reap most of the benefits of multipath routing without violating the export policy. The "strict policy" line is a bit more restrictive but still performs quite well.

5.3 Avoiding an AS in Default Path

Counting the number of paths is not sufficient to evaluate the effectiveness of MIRO, since many of the paths may share some nodes or edges in common. Next, we evaluate the ability of MIRO to satisfy a specific policy objective: avoiding an intermediate AS known to have security or performance problems. We calculate the success rate for every (source AS, destination AS, and AS-to-avoid) triple. We deliberately exclude cases where the AS-to-avoid is an immediate neighbor of the source AS. In these cases, avoiding the AS would require the source to select a path from another immediate AS anyway. In addition, an AS is not likely to distrust one of its own immediate neighbors.

5.3.1 Success rate of different policies

Table 2 presents the cumulative percentage of the success rate for each policy. As expected, the table shows that single-path, multi-path, and source routing policies provide increasing degrees of flexibility. In the single-path case, the source AS can only satisfy its policy objective by selecting a route announced by another immediate neighbor. In the multi-path case, we allow the source AS to use the routes announced by BGP or establish a routing tunnel with another AS. Although source routing can select any path, the source AS cannot always find a path that avoids the offending AS. If the AS-to-avoid lies on every path to the destination, then no policy can successfully circumvent the AS. We run a depth-first search algorithm on the graph to identify those nodes.

Multi-path routing performs very well for this application. Using the most strict multi-path policy, the success rate increases from around 30% in the single-path routing case to around 65%. Relaxing the policy boosts that number further to around 72%. If we allow the tunnels to traverse paths that violate conventional export policies, we can increase the success rate to around 75%. This is not all that far from the source-routing policy's success rate of 90%. Source routing achieves most of this gain by selecting paths that conflict with the business objectives for intermediate ASes. For example, source routing would allow two ISPs to communicate by directing traffic through a stub AS, which is not desirable.

Policy	Success Rate	AS#/tuple	Path#/tuple
strict/s	65.4%	2.55	15.9
export/e	72.9%	2.18	27.3
flexible/a	75.3%	2.00	71.5

a) Year 2000 data

Policy	Success Rate	AS#/tuple	Path#/tuple
strict/s	67.0%	2.83	28.7
export/e	74.6%	2.38	44.3
flexible/a	76.6%	2.22	106.8

b) Year 2003 data

Policy	Success Rate	AS#/tuple	Path#/tuple
strict/s	67.8%	2.80	36.6
export/e	73.7%	2.53	58.9
flexible/a	76.0%	2.38	139.0

c) Year 2005 data

Table 3: Comparing the intermediate states

5.3.2 Avoiding State Explosions

The next experiment quantifies the amount of state that MIRO must handle to negotiate a routing tunnel. We conduct this analysis by counting the number of ASes the source must contact, as well as the number of candidate paths received before a successful alternative is identified. For this test, we eliminate the cases where today's single-path routing would be successful, as MIRO would not need to establish tunnels on alternate paths. Table 3 lists the success rate of multi-path routing, the average number of ASes queries per (source, target, avoid) tuple, and the average number of paths obtained in each case.

For the 2005 data, when we use the flexible policy instead of the strict policy, the average number of ASes contacted decreases to 2.38 from 2.80, which seems to suggest that the source AS initiates fewer negotiations. However, by switching to flexible policy from the strict one, the average number of paths increases from 36.6 to 139, so we actually need to check more paths although there are fewer negotiations. Similar trends can be seen in other years, because the more flexible policy tends to allow more candidate routes in the responding AS. Comparing across the years, the number of paths per tuple increases with time because of the increasing connectivity of the AS topology.

5.3.3 Incremental Deployment

In the next experiment, we show that MIRO is effective even if only a few ASes adopt the enhanced protocol. In our tests, we found that a handful of highly connected Tier-1 ASes contribute to most of the path alternatives, if export policies are respected. Referring back to Figure 5, only 0.2% of the ASes has more than 200 neighbors, and less than 1% has more than 40. However, these ASes play an important role in MIRO. In Figure 7, the x-axis is the percentage of nodes that have adopted MIRO, plotted on a logarithmic scale. We assume that the source AS can only establish tunnels with one of these nodes, in order of decreasing node degree to capture the likely scenario where the nodes with higher degree adopt MIRO first. The y-axis plots the ratio of success in finding

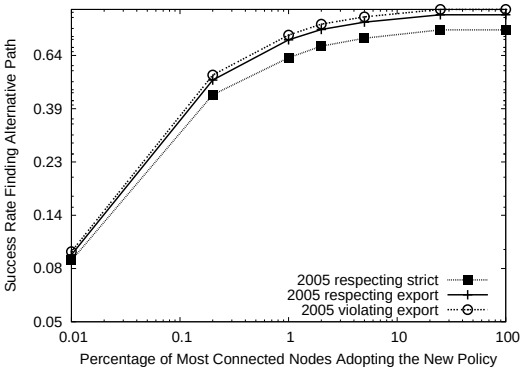


Figure 7: Incremental deployment

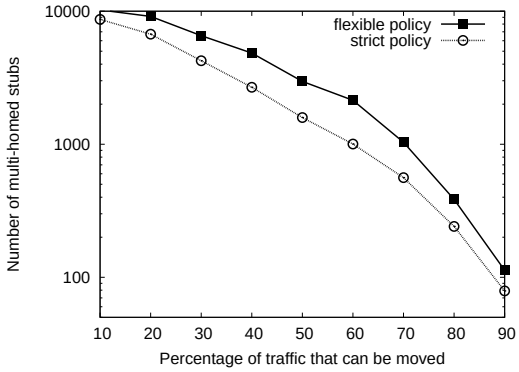


Figure 8: Multi-homed stub ASes with power nodes

a path that avoids the offending AS, using as base the numbers for ubiquitous deployment and the most flexible policy.

The curves in Figure 7 confirm that the most connected nodes contribute most of the benefit. If only the 0.2% most-connected nodes (i.e., nodes with more than 200 neighbors) adopt MIRO, we could already have around 40% to 50% of the total gain. If the 1% most-connected nodes (i.e., with degree greater than 40) adopted MIRO, we can get around 50% to 75% of the benefit; these nodes include many of the tier-1 and tier-2 ISPs. For the sake of comparison, we also evaluated the effects of low-degree nodes adopting the protocol first. In this analysis, we see success rates less than 10% until 95% of the nodes adopt MIRO. Therefore, it is not very effective to deploy the new protocol at the edge first. Fortunately, it is much more likely that a small number of large ASes would adopt MIRO than a large number of small ASes. Also, when a large ISP adopts MIRO, all of its customers immediately gain more flexibility, providing a nice motivation for adopting the protocol.

5.4 Controlling Incoming Traffic

Next, we present a brief evaluation of a second application of multi-path routing. In this example, we focus on multi-homed stub ASes that want to exert control over inbound traffic to balance load over multiple incoming links. Evaluating a traffic-engineering application is difficult without a global view of the offered traffic, so our results should be viewed as a back-of-the-envelope analysis to demonstrate the role that MIRO can play in this application. In the absence of traffic measurements, we make a number of simplifying assumptions. First, we assume that each source AS generates equal amounts of traffic. This allows us to estimate the total traffic on each incoming link simply by counting the number of source ASes using this link. Second, we assume all the ASes that transit through an intermediate AS for transit would always use this AS to send traffic to the destination. This allows us to calculate the amount of traffic that a single AS could move, if asked to switch to a different route.

We call a node a "power node" if it lies on the AS path to the destination AS for many source ASes. We evaluate the benefits of the destination AS requesting the power node to switch to an alternate path that traverses a different incoming link. If that power node advertises the new default path to all its neighbors, hopefully many neighbors will also switch to the new path. We evaluate this application by showing how many stub ASes can find at least one "power node" that can potentially move designated amount of traffic using this method. In Figure 8 both the flexible policy and the strict policy are examined on the 2005 data. In total, we tested 10,383 multi-homed stub ASes. The figure shows that around 90% of those stubs have at least one power node that can move more than 10% of the incoming traffic. Also, around half of them has one power node that can move at least 40% and 25% of traffic under the flexible and strict policy respectively.

We did some further analysis on the power nodes and found that more than 90% are nodes with more than 200 neighbors—most likely tier-1 ISPs. Immediate neighbors of the destination AS constitute only 9% of the power nodes; around 68% of the power nodes are two hops away from the destination AS. Therefore, we find that MIRO's ability to send requests to non-immediate neighbors offers a significant gain, and being able to negotiate with tier-1 ISPs, in particular, is especially useful.

5.5 Summary

Our experiments show that MIRO is very effective in helping ASes achieve their policy objectives. In the avoid-an-AS application, MIRO helps increase the success rate from 30% to 76% by establishing only one tunnel for a (source, destination) pair. Although source routing can push the success rate to 90%, it requires huge changes to the routing framework and must exploit unusual paths that traverse stub ASes. In the incoming-traffic-control application, we find that 90% of the stub ASes can move around 10% of traffic and half of them can move at least a quarter of the traffic by negotiating with a single intermediate AS.

We also showed that most of the alternate routes are provided by the most-connected nodes. This conclusion may lead people to conclude that MIRO benefits the big ISPs most. Yet, MIRO is designed to expose the existing candidate paths in the Internet, so it is not surprising that the participation of the well-connected ASes would provide the most benefit. Yet, these results are quite dramatic, suggesting that even early adopters would achieve a significant gain, especially if ASes can negotiate with ASes that are not immediate neighbors.

6. ROUTING POLICIES

The policy specification language is intentionally excluded in our design because the underlying mechanisms should give users maximum flexibility in picking and expressing their own policies. However, to give the readers a concrete picture, we will present some sample policies and describe how they can be configured. We start by describing how policy configuration is done in current Internet and proceed with comparison to the multi-path case.

6.1 Policy Configuration in Current Internet

The current BGP specification [9] only describes how two BGP neighbors exchange information and the decision process, without

defining routing policy specifications. In response, various vendors have come up with their own policy specification languages and tools. BGP policies can be divided into import policies and export policies. Import policies define which routes to filter and how attributes such as local preference should be set for the remaining announcements. Export policies filter the paths advertised to each neighbor and adjust the route attributes. The BGP decision process selects the route with highest local preference. If several routes are equal on local preference, a set of steps are applied to break ties, such as comparing AS-path length and other route attributes.

Cisco designed the route-map command that can be used to configure policy routing. The operator can specify the actions to be taken when matching condition is satisfied. For example, the following route-map command specifies that any route received from 12.34.56.1 that matches the filter parameters set in AS access list 200 ("never go through AS 312") will be accepted and have its local preference set to 250.

Cisco route-map example

```
 router bgp 100
!
  neighbor 12.34.56.1 route-map FIX-LOCALPREF in
  neighbor 12.34.56.1 remote-as 1
!
route-map FIX-LOCALPREF permit 10
  match as-path 200
  set local-preference 250
!
  ip as-path access-list 200 deny _312_
```

6.2 Multi-path Routing Policies

In addition to defining how to filter and manipulate route announcements, we must also define how negotiations should be conducted. We divide the policies into two parts: negotiation rules that deal with establishing and managing negotiations, and route-selection rules that filter and rank the available alternatives. In the requesting AS, the rules should specify when to trigger negotiation and whom to negotiate with. In the responding AS, the rules should describe when and from whom new negotiations will be allowed.

- *Requesting: when to trigger negotiation* Negotiations should only be triggered if none of the current routes satisfy desired property. Therefore the conditions triggering negotiations can be checked whenever routes change.
- *Requesting: whom to negotiate with* The requesting AS has to guess which AS may have appropriate candidate routes; good guesses can greatly shorten the negotiation process. For a security policy like "avoiding AS 312," some possible candidates are the ASes on the default path between the requesting AS and AS 312 that understand the new protocol.
- *Responding: whether to allow negotiations* The responding AS could specify a limit for the total number of tunnels, a rate limit for establishing new tunnels, or a firewall where only negotiation requests from trusted peers are accepted.

The responding AS could specify filter rules to selectively export its candidate routes. The requesting AS should also set evaluation rules to determine which candidate to pick. Those rules may evaluate several factors in the decision process, like the price cost or the quality of different routes.

- *Route filtering* The filtering rules can draw on existing route attributes, e.g., only advertise routes that have a local preference of more than 100. In practice, an ISP often assigns all customer routes the same preference value, all peer routes with lower values, and all provider routes with even smaller values. Therefore we can easily specify the selective export rules described in Section 3.4 based on local preference.
- *Route preference and cost* The routes more preferred to the requesting AS may be those less desired to the responding AS. For example, the requesting AS wants to select a low latency route in the responding AS which goes through an expensive provider link. In this case, we could introduce a price system so that the responding AS is compensated accordingly. Any notion of price would work as long as both parties agree on it. With a price tag attached to each route, innovative business models could be enabled. For example, the responding AS could sell all customer routes for a lower price and all peer routes for a higher price. The requesting AS then picks routes based on both local preference and cost.

Optionally, the requesting ASes could specify simple requirements to avoid sending useless candidate routes. For example, the requesting AS could explicitly request "only give me paths without AS 312". The responding AS adds the requirement to candidate filtering before responding with final answers.

7. DISCUSSIONS

7.1 Route Convergence

Since MIRO changes how ASes select interdomain routes, we need to consider the possible effects on BGP convergence. Previous work has shown that certain combinations of routing policies can cause BGP to oscillate [22]. Follow-up work showed that convergence is guaranteed if ASes select and export routes based on the conventional business relationships [20]. However, since MIRO provides ways for ASes to violate these guidelines, convergence problems could potentially arise.

MIRO is guaranteed to converge in a restricted, yet important, scenario. If the upstream AS does not advertise the tunneled path to any other AS, MIRO converges whenever the underlying BGP converges. For example, the many stub ASes in the Internet do not export routes learned from one upstream provider to another and, as such, would never export a tunneled path "back into BGP." In reality, a requesting AS often needs just one tunnel to satisfy its path-selection goals. The diameter of the AS graph is small, and MIRO enables an AS to negotiate with non-neighboring ASes. As such, we envision that an end-to-end path would typically include at most one tunnel. In summary, we think this conservative requirement would not be too restrictive for the following reasons:

- Most ASes are stub ASes. In the 2005 topology generated by the Gao algorithm, 17,347 out of 20,930 ASes are stubs.
- The observed average AS path length is only 4, therefore tunnel concatenations are likely to be very rare—so rare we could preclude them.
- We allow negotiations between non-adjacent ASes, so instead of establishing a chain of tunnels, the source AS can directly contact the other end of the chain.

As ongoing work, we are creating a formal model of multi-path routing to establish these convergence properties. We have found several ways to relax the "just one tunnel" requirement that we are exploring in more detail.

7.2 Routing Loops

BGP takes great care to ensure that paths do not contain loops. As each router forwards packets solely based on destination IP address, loops in BGP paths can lead to lost packets. However, in overlay networks, packets can physically traverse an AS more than once. For example, if an overlay node is located in an AS *X* single-homed to its ISP *Y*, all packets forwarded by this node will traverse the network of *Y* twice. But this will not lead to lost packets, as packets in different tunnels bear different destination IP addresses. Similarly, traversing an AS more than once is not a problem in MIRO, as long as all the tunnel endpoints and the default path between tunnel endpoints form a loop-free path—a property that could be easily checked during the negotiation process.

However, traversing an AS more than once may be inefficient, so ASes in MIRO can also enforce a stricter kind of loop detection. Both negotiating parties know the path the packets will take when they leave the tunnel; moreover, the upstream AS knows the path traversed by the tunnel itself. Therefore, the upstream AS should concatenate both parts and reject negotiations if any AS appears more than once in the resulting end-to-end path.

7.3 Route Aggregation and Security

Like many studies of interdomain routing, we implicitly assume that the AS path in the BGP announcements identifies the actual sequence of ASes the data packets would traverse. However, route filtering and route aggregation may violate this assumption. A downstream AS may have a BGP route for a more-specific prefix, which would deflect data packets to a different path. Similar path inconsistencies can arise if an adversary has control over the data plane and deflects packets to a different path. Packet deflection is a general problem that can complicate BGP routing. Ultimately, a secure and robust interdomain routing infrastructure may require compromising on support for route aggregation (e.g., by routing all traffic at the AS level, rather than at the prefix level). Effective support for multi-path routing makes that possible, since ASes could still achieve their load balancing, performance, and security goals without needing to announce separate routes for each destination prefix. We plan to explore these issues in more detail in our ongoing work.

8. RELATED WORK

Previous work has considered other approaches to flexible Internet routing. Source-routing proposals [1–5] can provide multiple routes for every source-destination pair, and several of them [1, 5] explicitly suggest routing at the AS level rather than at the router level, as we do in MIRO. However, source routing does not give intermediate ASes much control over path selection. Some work considers receiver policies [4], but primarily to filter traffic coming from suspicious routes. MIRO bears some similarities to overlay networks [6], in terms of establishing tunnels that encapsulate and decapsulate packets. However, MIRO selects paths on the underlay with the cooperation of the routers in intermediate ASes, rather than directing packets over virtual links to intermediate hosts.

Several papers propose new ways to disseminate reachability information. Nimrod [23] uses clusters to hide the internal topology of a network, revealing additional details only upon request. However, the members of a Nimrod cluster must be contiguous, while MIRO's negotiations can happen between arbitrary pairs of ASes. Also, the Nimrod work does not present the technical details of how clusters and the request-response protocol should be realized. In contrast, MIRO can be deployed incrementally as an extension to today's BGP protocol. The recent HLP [14] proposal uses a hybrid of link-state and path-vector routing. Multiple ASes with provider/customer relationships form a group and use link-state routing to compute paths; the groups use a path-vector protocol to exchange routing information with each other. In contrast, MIRO uses BGP route announcements by default and supports negotiation between arbitrary pairs of ASes.

Other routing architectures consider the role of cost and incentives in making interdomain routing decisions. Nexit [24] enables pairs of neighboring ASes to cooperate in selecting egress points for exchanging traffic, to avoid the inherent inefficiency of hot-potato routing and conventional traffic engineering practices [25]. In contrast to MIRO, the negotiation in Nexit focuses specifically on selecting between the existing BGP-learned routes at multiple egress points rather than discovering new interdomain routes. In that sense, the two proposals are complementary and could conceivably be part of a larger framework for using negotiation to improve interdomain routing. Another recent study [26] proposes a routing system that advertises multiple AS paths, with pricing information attached to each announcement. However, the paper does not present a concrete design and evaluation of a protocol, making it difficult to compare to MIRO directly.

Multi-path routing has been explored in the context of *intra*domain routing. Equal Cost Multi-Path (ECMP) allows routers to split traffic over multiple shortest paths in intradomain routing protocols such as OSPF and IS-IS. Some proposals have considered ways to relax the requirement that all of the paths between two nodes have the same (lowest) cost [27]. In addition, recent work on TeXCP [16] has explored how to split traffic over multiple intradomain paths for more effective traffic engineering. In TeXCP, ingress nodes dynamically adapt the splitting of traffic over multiple paths, which are computed in advance. TeXCP and MIRO are complementary, in that MIRO focuses on identifying and selecting paths, whereas TeXCP focuses on how to adjust the proportions of traffic that traverse the paths.

Techniques for selecting multiple paths within an AS do not extend directly to interdomain routing. Within an AS, routers can share topology information and have a common objective. In contrast, in interdomain routing, ASes have limited information about the network topology and may have different (or even conflicting) path-selection goals. Some recent work has proposed extensions to BGP to propagate QoS metrics [28]. However, this approach is problematic in practice because its requires extensive deployment and cooperation among ASes, and may introduce scalability challenges if the QoS information changes frequently.

Recent work at the IETF proposes extensions to BGP to enable a BGP speaker to announce multiple routes for the same prefix [29], without describing how these routes are selected, exported, or installed in the data plane. An implementation of MIRO could adopt the protocol extensions as a way to identify the advertised routes. Another related IETF activity is the Path Computation Element (PCE) working group [30] that is defining an architecture that allows special computational components to select paths on behalf of the routers. PCE is meant to support constraint-based path computation both within and across ASes, with an emphasis on satisfying traffic-engineering goals by establishing MPLS label-switched paths. In contrast, MIRO was designed as an incrementally deployable extension to BGP to support multipath routing. Still, the two schemes share similar requirements for ASes to cooperate in selecting paths while hiding topology details from each other.

9. CONCLUSION

In this paper, we have presented a multi-path interdomain routing protocol, called MIRO. MIRO defaults to the single-path routing provided by conventional BGP but allows ASes to negotiate alter-

nate paths as needed. This provides flexibility where needed while remaining backwards compatible with BGP. Compared to source routing, MIRO gives transit ASes more control over the flow of traffic in their networks. An evaluation on realistic AS-level topologies shows that MIRO exposes much of the underlying path diversity in the Internet, even when only the major ISPs have deployed the enhanced protocol. We also find that significant path diversity is available even if ASes adhere to conventional practices for exporting routes based on their business relationships.

A natural next step is to flesh out the implementation and build a prototype system, to quantify the overheads for encapsulating and decapsulating packets, as well as maintaining the tunnel tables. We can also evaluate the overhead for distributing the tunnel tables, as a function of network topology. Another interesting direction for study is the security implications of MIRO. Without any security measures, adversaries could spoof the tunnel identifiers to direct their traffic onto better paths or launch a denial-of-service attack on the downstream AS. A trust system should be in place so spoofed tunnel identifiers could be detected as early as possible.

Efficient support for multi-path routing enables a variety of techniques for ASes to balance load and optimize performance, beyond the load-balancing schemes today's multi-homed ASes can employ. However, the flexibility to split traffic over multiple paths introduces the possibility of oscillation, where each AS adjusts its division of traffic in response to congestion introduced by another AS. Divising a decentralized load-balancing scheme that prevents oscillation is an interesting avenue for future work. In addition, by allowing ASes to negotiate for alternate routes, MIRO opens up many interesting questions about how to incorporate pricing, load, and performance information into the path-selection process.

10. ACKNOWLEDGMENTS

This work was supported by HSARPA grant 1756303, and a URP grant from Cisco. We thank Ming Zhang for his valuable feedback in the early stages of this work. We are also grateful to Ioannis Avramopoulos, Elliott Karpilovsky, Dan Wendlandt, Yaping Zhu, and the anonymous reviewers for their comments and suggestions.

11. REFERENCES

[1] D. Zhu, M. Gritter, and D. Cheriton, "Feedback based routing," in *Proc. SIGCOMM Workshop on Hot Topics in Networking*, October 2002.

[2] H. T. Kaur, S. Kalyanaraman, A. Weiss, S. Kanwar, and A. Gandhi, "BANANAS: An evolutionary framework for explicit and multipath routing in the Internet," in *Proc. Future Directions in Network Architecture*, 2003.

[3] B. Raghavan and A. C. Snoeren, "A system for authenticated policy-compliant routing," in *Proc. ACM SIGCOMM*, pp. 167–178, 2004.

[4] K. Argyraki and D. R. Cheriton, "Loose source routing as a mechanism for traffic policies," in *Proc. Future Directions in Network Architecture*, 2004.

[5] X. Yang, "NIRA: A new Internet routing architecture," in *Proc. Future Directions in Network Architecture*, 2003.

[6] D. Andersen, H. Balakrishnan, F. Kaashoek, and R. Morris, "Resilient overlay networks," in *Proc. SOSP*, 2001.

[7] B. Quoitin, S. Uhlig, C. Pelsser, L. Swinnen, and O. Bonaventure, "Interdomain traffic engineering with BGP," *IEEE Communication Magazine*, 2003.

[8] "University of Oregon Route Views Project." http://www.routeviews.org.

[9] Y. Rekhter, T. Li, and S. Hares, "A Border Gateway Protocol 4 (BGP-4)." RFC 4271, January 2006.

[10] G. Huston, "Interconnection, peering, and settlements," in *Proc. INET*, June 1999.

[11] L. Gao, "On inferring Autonomous System relationships in the Internet," *IEEE/ACM Trans. Networking*, vol. 9, no. 6, pp. 733–745, 2001.

[12] L. Subramanian, S. Agarwal, J. Rexford, and R. H. Katz, "Characterizing the Internet hierarchy from multiple vantage points," in *Proc. IEEE INFOCOM*, June 2002.

[13] L. Qiu, Y. R. Yang, Y. Zhang, and S. Shenker, "On selfish routing in Internet-like environments," in *Proc. ACM SIGCOMM*, pp. 151–162, 2003.

[14] L. Subramanian, M. Caesar, C. T. Ee, M. Handley, M. Mao, S. Shenker, and I. Stoica, "HLP: A next generation inter-domain routing protocol," in *Proc. ACM SIGCOMM*, pp. 13–24, 2005.

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

[16] S. Kandula, D. Katabi, B. Davie, and A. Charny, "Walking the tightrope: Responsive yet stable traffic engineering," *Proc. ACM SIGCOMM*, vol. 35, no. 4, pp. 253–264, 2005.

[17] E. Rosen, A. Viswanathan, and R. Callon, "Multiprotocol Label Switching Architecture." RFC 3031, January 2001.

[18] M. Caesar, D. Caldwell, N. Feamster, J. Rexford, A. Shaikh, and J. van der Merwe, "Design and implementation of a routing control platform," in *Proc. NSDI*, May 2005.

[19] X. Yang, D. Wetherall, and T. Anderson, "A DoS-limiting network architecture," *ACM SIGCOMM Computer Communication Review*, vol. 35, no. 4, pp. 241–252, 2005.

[20] L. Gao and J. Rexford, "Stable Internet routing without global coordination," *IEEE/ACM Trans. Networking*, vol. 9, no. 6, pp. 681–692, 2001.

[21] Z. M. Mao, L. Qiu, J. Wang, and Y. Zhang, "On AS-level path inference," in *Proc. ACM SIGMETRICS*, 2005.

[22] T. G. Griffin, F. B. Shepherd, and G. Wilfong, "The stable paths problem and interdomain routing," *IEEE/ACM Trans. Networking*, vol. 10, no. 2, pp. 232–243, 2002.

[23] I. Castineyra, N. Chiappa, and M. Steenstrup, "The Nimrod Routing Architecture." RFC 1992, August 1996.

[24] R. Mahajan, D. Wetherall, and T. Anderson, "Negotiation-based routing between neighboring ISPs," in *Proc. NSDI*, 2005.

[25] R. Johari and J. Tsitsiklis, "Routing and peering in a competitive Internet." LIDS Publication 2570, 2003.

[26] M. Afergan and J. Wroclawski, "On the benefits and feasibility of incentive based routing infrastructure," in *Proc. Workshop on Practice and Theory of Incentives in Networked Systems*, pp. 197–204, September 2004.

[27] J. Chen, P. Druschel, and D. Subramanian, "An efficient multipath forwarding method," in *Proc. IEEE INFOCOM*, pp. 1418–1425, March 1998.

[28] L. Xiao, K. Lui, J. Wang, and K. Nahrstedt, "QoS extension to BGP," in *Proc. International Conference on Network Protocols*, 2002.

[29] D. Walton, A. Retana, and E. Chen, "Advertisement of multiple paths in BGP." Internet Draft, draft-walton-bgp-add-paths-05.txt, Expires August 2006.

[30] "Path computation element charter." `http://www.ietf.org/html.charters/pce-charter.html`.