HLP: A Next Generation Inter-domain Routing Protocol

Lakshminarayanan Subramanian *Matthew Caesar* *Cheng Tien Ee*
Mark Handley *Morley Mao* *Scott Shenker* *Ion Stoica*

Report No. UCB/CSD-03-1242

October 28, 2005

Computer Science Division (EECS)
University of California
Berkeley, California 94720

HLP: A Next Generation Inter-domain Routing Protocol

Lakshminarayanan Subramanian Matthew Caesar Cheng Tien Ee Mark Handley
Morley Mao Scott Shenker Ion Stoica

October 28, 2005

Abstract

It is well-known that BGP, the current inter-domain routing protocol, has many deficiencies. This paper describes a hybrid link-state and path-vector protocol called HLP as an alternative to BGP that has vastly better scalability, isolation and convergence properties. Using current BGP routing information, we show that HLP, in comparison to BGP, can reduce the churn-rate of route updates by a factor 400 *as well as isolate the effect of routing events to a region* 100 *times smaller than that of BGP. For a majority of Internet routes, HLP guarantees worst-case linear-time convergence. We also describe a prototype implementation of HLP on top of the XORP router platform. HLP is not intended to be a finished and final proposal for a replacement for BGP, but is instead offered as a starting point for debates about the nature of the next-generation inter-domain routing protocol.*

Categories and Subject Descriptors

C.2.6 [Communication Networks]: Internetworking

General Terms

Algorithms, Design, Experimentation, Performance.

Keywords

Inter-domain routing, BGP, scalability, convergence.

1 Introduction

Inter-domain routing presents a formidable combination of algorithmic and policy challenges. On the one hand, given the size and the rapid growth of the Internet, any inter-domain routing protocol should satisfy basic desirable algorithmic properties, such as scalability, robustness, and rapid convergence. On the other hand, for economic reasons inter-domain routing should support *policy routing*, where ISPs have the flexibility to implement a wide variety of *private* routing policies that ISPs choose not to reveal. Moreover, the routing protocol should provide sufficient information to enable ISPs to make informed policy decisions.

Designing an inter-domain protocol that satisfies both the algorithmic and policy requirements represents a very challenging task. There is an inherent conflict between the economic need for *fully-informed* and *private* routing policies and the structural need for robust routing algorithms. One could consider a spectrum of designs making different trade-offs. The Border Gateway Protocol (BGP) takes an extreme position in this design space that all routing policy must be private; no policy information is transmitted in route updates, leaving policy to be implemented entirely by local filters whose contents are kept secret. As a result, BGP suffers from inherent algorithmic problems, including poor scalability, minimal fault isolation, and slow convergence due to uninformed path exploration.[1] These problems, while mere nuisances in the Internet's early days, are becoming significantly more serious as expectations and demands placed on the Internet increase.

Although BGP does not distribute policy information, in practice it is impossible to hide certain policies because the routing protocol must distribute reachability and path information. Specifically, most provider-customer relationships are easily inferable from routing information distributed to the entire Internet [28, 9]. In addition, even though BGP provides complete path information to all ISPs, the vast majority of implemented policies do not use this information. This suggests that the extreme position taken by BGP, keeping full privacy and providing full path information, is not needed, nor perhaps even tenable.

In this paper, we explore a design point that is less extreme than BGP by proposing and evaluating a *hybrid link-state path-vector* routing protocol, called *HLP*. The design philosophy of HLP is to expose the *common case of policies* and to withhold some path information. This common case of policies exploits the assumption that a majority of Internet routes (99%) obey the structure of the Autonomous System (AS) hierarchy as imposed by provider-customer relationships. Given that this structure is largely inferable today [9, 28] and relatively stable (as we show later in this paper), HLP optimizes the routing protocol based on this structure. By analyzing the evolution of Internet routing and the growth of the Internet routing structure, we contend that this common case of policies is not merely an artifact of today's practices but is bound to stay as a common-case behavior in the future. In essence, HLP leverages the common-case policy behavior that BGP cannot hide and optimizes the protocol design for this common case. For routing policies that

[1] While some problems have been dealt with by modest incremental modifications [23, 7, 29], we contend that many of the problems are fundamental to BGP's basic architecture.

do not fit the common case behavior, HLP resorts to mechanisms resembling those of BGP to accommodate them.

The central idea used in HLP to optimize for the common case is to use *explicit information hiding* of unnecessary routing updates across provider-customer hierarchies and thereby limiting the global visibility and effect of routing events. Information hiding is fundamentally required to improve the scalability and isolation properties of inter-domain routing. If every routing event is globally visible, then the network churn grows at least linearly (if not super-linear) with the network size, which is clearly undesirable. HLP uses the provider-customer hierarchy to limit the visibility of routing information across hierarchies. Moreover, HLP's information hiding mechanism naturally fits today's routing assumptions and requires minimal modifications for deployment.

Information hiding on HLP gives substantially improved scalability, isolation, convergence and fault diagnosis properties. For the current Internet topology, the churn rate of HLP route advertisements is roughly 400 times less than with BGP. For roughly 50% of inter-AS links, HLP can isolate the effects of a fault to a region 100 times smaller than that of BGP. For most Internet routes, HLP achieves linear-time convergence by explicitly constraining the path-exploration process. HLP can support most of BGP's policies and also enables some new ones. HLP also replaces BGP's prefix-deaggregation approach to traffic engineering, which can affect route convergence and cause churn, with a cleaner approach based on cost-based traffic engineering and static prefix deaggregation. HLP also addresses many of the security and fault diagnosis problems of BGP, but we do not discuss these issues in this paper due to space constraints.

The rest of the paper is organized as follows. In Section 2, we highlight some of the pressing problems of BGP and elaborate upon the different design issues that confront the designer of any inter-domain routing protocol. In Sections 3 and 4, we describe the HLP protocol and analyze its properties. In Sections 5, we discuss traffic engineering issues in HLP and present the router level perspective of HLP in Section 6. We describe related work in Section 7 and conclude in Section 8.

2 Design Rationale

We start this section by highlighting three specific pressing deficiencies of BGP. We then describe four basic design issues and contrast the decisions taken in HLP to those in BGP.

2.1 Problems with BGP

The IRTF convened two separate working groups to define the set of requirements for a future generation inter-domain routing protocol. From their combined set of specifications [15], we selected five requirements of paramount importance, and describe the ways in which BGP fails to meet them:

Table 1: Distinctions between HLP and BGP

Design issue	BGP	HLP
Routing structure	Flat	Hierarchical
Policy structure	Support for generic policies	Optimize for common case of policies
Granularity of routing	Prefix based	AS based
Style of routing	Path vector	Hybrid routing

Scalability: Any future inter-domain routing protocol must gracefully accommodate the ongoing growth of the Internet. BGP fails this test, as its routing state and rate of churn (the rate of routing announcements received by a given router) grow linearly with the size of the network. Since 1997 the routing table has grown from 3,000 to over 17,000 Autonomous Systems (AS's) and from 50,000 to over 180,000 routing prefixes, so the issue of scaling is becoming increasingly important.

Convergence and Route Stability: To provide reliable reachability, Internet routes should be relatively stable and, when a change is necessary, they should quickly converge to their new steady-state. However, BGP is known to suffer from significant route instabilities, route oscillations and long convergence times. Nearly 25% of BGP prefixes continuously flap and a large fraction of these have convergence times on the order of hours [5]. The remaining 75% of relatively stable prefixes typically take between $2-5$ minutes to converge.

Isolation: No design can be robust and scalable if local faults within a network can have global impact. Unfortunately, BGP has very poor fault isolation properties. A simple analysis of Routeviews BGP data [32], shows that nearly 20% of the routing events are globally visible and many updates observed at a router are largely a result of events far removed from the router.

2.2 Basic Design Issues

We now contrast BGP's approach with HLP's along four design issues that face any designer of inter-domain routing protocols: routing structure, policy, routing granularity and routing style. This is not meant to be an exhaustive list, but is limited to the areas where, in our opinion, BGP is in most need of modification. For context, Table 1 summarizes the primary distinctions between HLP and BGP across these design issues.

2.2.1 Routing Structure

In order to support fully general path-based policies, BGP reveals complete path information. As a result, *local* routing events can be *globally* visible [11]. This impairs BGP's scalability, and also makes it fundamentally hard to isolate routing events [15, 11]. Moreover, the resulting interdependence between ASs makes the entire Internet vulnerable to

localized security or configuration problems; a single configuration error or compromised router can affect the rest of the network [20].

To avoid these problems, HLP hides some path information. It does so by using the natural hierarchical routing structure defined by the typical relationships between interconnected ASs — peers, customers, and providers – and *hiding* the small-scale routing dynamics in one hierarchy from nodes in another hierarchy.

2.2.2 Policy

While revealing complete path information, BGP keeps policy information private. However, this quest for policy privacy is largely futile. The vast majority of relationships between ASs can be categorized as peers, customers, or providers and, moreover, these provider-customer relationships can be accurately inferred [9, 28]. The export-rule and route preference policy settings in nearly 99% of the AS's follow two simple guidelines based on these inter-AS relationships [9, 10, 28]:

Export-rule guideline: Do not forward routes advertised by one peer or a provider to another peer or provider [10][2].

Route preference guideline: Prefer customer-routes over routes advertised by peers or providers.

While these policies dominate usage, BGP's refusal to explicitly reveal them means that BGP is unable to distinguish between a misconfigured policy and a genuine one, making BGP much harder to manage and diagnose, and more susceptible to misconfigurations and attacks. Additionally, in the absence of strict guidelines on how to set policies, policy privacy can lead to policy conflicts, poor convergence and routing instabilities [13].

HLP, in contrast, explicitly publishes the provider-customer relationships and restricts the normal set of available paths to a destination to those that obey the hierarchies defined by these relationships. HLP does allow policies that do not obey these two simple rules, but it treats those as *exceptions* and provides additional mechanisms for supporting them. The result is a routing protocol that, in the common case, can recognize misconfigurations and limit the propagation of route advertisements.

2.2.3 Routing Granularity

BGP uses prefix-based routing. While the initial design of BGP promoted aggregation of prefixes to improve scalability, today's usage is dominated by the opposite phenomenon - route deaggregation for traffic engineering, multihoming and policy routing. The last 4 years' worth of BGP routing information show that nearly $11,000$ networks (including $2,800$ /24 networks) deaggregated their prefix, with a mean deaggregation factor of 8.5 (2.5 for $/24$ networks). This, in combination with the advent of many $/24$ networks, has resulted in an alarming rise in the number of distinct prefixes in a routing table; since a single routing event triggers a separate routing update for each prefix, this increase in prefixes has led to greatly increased churn.

It does not appear that this deaggregation is being fully utilized for route diversity; measurements suggest that the number of distinct paths from a vantage point to the same destination AS is less than or equal to 2 for more than 99% of ASs [6].

Given that prefix based routing results in greater churn and larger routing tables, and yet does not usually result in differing paths, we designed HLP to route at the granularity of AS's instead of prefixes. This separates routing from addressing, which had been conflated in BGP. In addition to reduced routing state and churn, routing at the AS granularity has several ancillary benefits. Because the mapping between address prefixes and locations (as identified by AS) is much more static than the topology of the network, more appropriate transport and security mechanisms can be used for the topology information and for the AS-to-prefix mapping information. This, in turn, allows for easy detection of *origin misconfigurations*, in which an AS erroneously claims ownership of the prefix owned by another AS.

2.2.4 Routing Style

BGP uses path-vector routing. Path-vector routing enables complex policies (since it enables ASs to base their policies on the entire path) and easy loop-suppression. But the worst-case convergence of a path-vector protocol grows exponentially with the length of the path [18, 19]. Path vector routing also introduces unnecessary interdependence[3] which impedes the scalability and isolation properties of the protocol.

The alternatives to path-vector (PV) are the standard distance-vector (DV) and link-state (LS) styles of routing, neither of which are good candidates for supporting policy-based routing. DV routing does not reveal any information about the path to a destination, thereby hindering policy routing. LS routing, on the other hand, may violate privacy norms of policies by revealing every activity to all destination AS's.

Apart from policies, LS and DV routing have their own protocol strengths and limitations. LS routing has fast convergence and incurs low churn, the latter because updates are for link events, not routing changes. (In PV and DV routing, one

[2] A specific variation to the export guideline which we do not consider as a violation is indirect-peering. Some ASs forward announcements from one peer to another peer either due to indirect peering (lack of direct connectivity) or due to sibling relationships (two AS's under same administration).

[3] A single routing event on a link triggers route updates to every AS that utilizes some path traversing the link thereby making a large fraction of routing events globally visible.

link event can cause many route changes.) Moreover, fault diagnosis is easy with LS protocols, because it provides complete visibility into the current state of the network. However, global visibility is antithetical to both scaling and isolation.

DV routing, in contrast, can be adapted to provide good isolation (as we show later in Section 3, nodes can hide minor cost changes to isolate the effect of routing events), but fault diagnosis is difficult.

None of these approaches are ideal solutions, but each has its own merits, and thus HLP uses a hybrid of link-state and path-vector routing. At first glance this might seem overly complex, but the hierarchical structure provides a natural way to decompose routing between the two styles; HLP uses link-state within a given hierarchy of AS's (as specified by provider-customer relationships) and uses path-vector between hierarchies. The link-state component improves convergence and reduces churn within a hierarchy, while the path-vector component preserves global scalability by hiding internal route updates across hierarchies (thereby sacrificing global visibility).

The discussion of these four design issues was intended to give a flavor of the intuition behind HLP's design. In the next section we describe how HLP actually works.

3 The HLP Routing Model

In this section, we describe the HLP routing protocol. We begin by describing the routing structure and the basic route propagation model of HLP in Sections 3.1 and Section 3.2. In Section 3.3, we explain the concept of *information hiding* which forms the key design principle of HLP that provides improved scalability and isolation properties. Later, in Sections 3.4 and 3.5, we describe how HLP handles complex AS relationships and variations to the default policy guidelines.

3.1 HLP routing structure

The design of HLP leverages the existence of a hierarchical structure in the AS topology based on provider-customer relationships. Figure 1 illustrates one such sample AS-level topology consisting of several provider-customer AS hierarchies. For ease of exposition, we assume that each hierarchy is based only on the basic provider-customer relationships and does not incorporate any complex relationships (*e.g.*, two ISPs that do not reveal their relationship or have two different relationships in different geographic locations). We will discuss how HLP handles such complex relationships later in Section 3.4.

We refer to the root AS of each such provider-customer hierarchy as a *tier-1 AS*. This deviates from the conventional terminology of tier-1 ISPs, in that, a lower-tier ISP would be classified as a tier-1 AS by our definition if it is not an explicit customer of any other AS. An AS with multiple providers (*e.g.*, multi-homed AS) can be part of more than one provider-customer hierarchy. AS's in different provider-customer hierarchies can connect using peering links and these peering links can occur at various levels in the provider-customer hierarchy. We assume that there are *no cycles* in the provider customer hierarchy.[4]

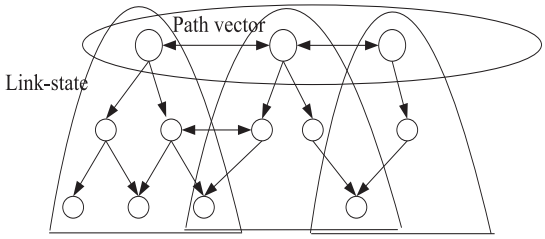

Figure 1: An AS hierarchy indicating provider-customer and peer-peer relationships. The unidirectional links represent provider-customer links and the bidirectional links represent peering links. Peering links can occur at different levels in the hierarchies.

3.2 Basic Route Propagation Model

Based on hierarchical routing structure, HLP uses a combination of *link-state* routing within a provider-customer hierarchy and *path-vector* routing across hierarchies.

Link-state aspect of HLP: Within each hierarchy, when an inter-AS routing event occurs, the other AS's in the hierarchy are notified using a link-state announcement. This link-state announcement is at the granularity of AS's and not at the granularity of routers. Every AS maintains link-state information about the inter-AS provider-customer links within its own hierarchy (inclusive of the links above it) and updates this information upon receipt of a link-state update.

Path-vector aspect of HLP: Between hierarchies, the path-vector part of HLP is similar to BGP, where an AS propagates reachability information tagged with an AS path. The primary distinction is that the HLP uses a *fragmented path vector (FPV)* that contains only a portion of the AS path to the destination, rather than the entire AS path as with BGP. The FPV omits the portion of the AS path within an AS hierarchy. As the length of the FPV path has no routing significance, every FPV advertisement also carries a cost metric.

We now describe through example the basic model of how routes are propagated within and between AS hierarchies. Each node maintains a link-state topology database and a path-vector style routing table. Nodes exchange two types of messages: link-state advertisements (LSAs) and fragmented-path vectors (FPVs) (Figure 2).

[4]The current Internet topology satisfies this property and we assume that this would obviously hold in the future. If a cycle does arise, we need to treat certain links as complex relationships (refer to Section 3.4) to explicitly break the cycle.

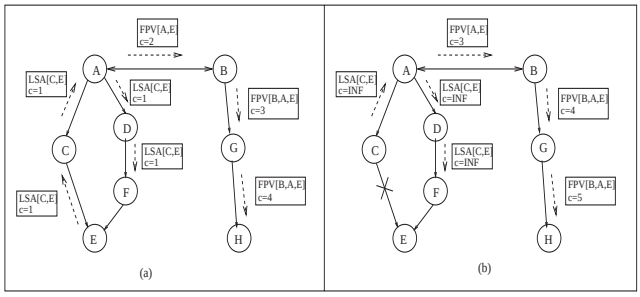

Figure 2: Basic HLP route propagation: Link failure example

Consider the example AS-topology in Figure 2(a) comprising of two provider-customer hierarchies rooted at A and B. Consider link (C, E) in this topology. Initially an LSA informs all the nodes in A's hierarchy of the existence and cost of link (C, E) (here, we consider all links to have a cost of 1). A receives the LSA, and propagates a path-vector to B, with FPV (A, E) and a cost metric of 2. The path vector is then distributed down the hierarchy to H without further modification of the path - neither the path within A's hierarchy nor the path within B's hierarchy appear in the FPV.

In a modified example illustrated in Figure 2(b), when link (C, E) subsequently fails, nodes within A's hierarchy receive an LSA to inform them of the link-failure. However, since A has an alternate path within its own hierarchy, A sends a path-vector update to B with a modified cost. This is essentially the same as a route withdrawal in BGP. In turn, B propagates the FPV down its own hierarchy to H. If however, A did not have an alternate path, A will propagate a route withdrawal to B.

FPV advertisements may be propagated across more than one peering link. Such forwarding allows HLP to express indirect peering, where an AS exports announcements from one peer to another. In such cases, the FPV path includes all the peering AS's along all the paths to avoid routing loops or the need to perform a cost count to infinity.

To summarize HLP's basic routing model:

1. All AS's maintain a link-state database of the topology in their local hierarchy.
2. The AS path in each FPV includes all AS's whose peering links were traversed, but excludes the parts of the path within the AS hierarchies.
3. All inter-AS links have a cost metric which is added to the net cost value in an FPV route advertisement.
4. HLP can model indirect peering by allowing the forwarding of route advertisements across more than one peering link.

Theorem 1: *In the absence of cycles in the provider-customer hierarchy, if every AS follows the HLP route propagation rules and every AS chooses a customer route if one exists, then the routing protocol is devoid of non-transient routing loops and the count to infinity problem.*

The proof of this theorem uses the following simple labeling of the links: Associate a label 3 with any customer-provider link that appears along a path, a label 2 to a peering link and a label 1 to a provider-customer link. The HLP propagation rules ensure that the labels of any valid routing path is always *non-increasing*. A non-transient routing loop will clearly violate this non-increasing property unless if all the links in the loop have the same label. Such a loop can comprise only of peering links (otherwise, the provider-customer hierarchy has a cycle). The FPV argument in every HLP route advertisement contains the entire path of peering links to avoid such loops. Hence, the basic route propagation model of HLP is devoid of non-transient loops. Transient loops can however occur in the middle of a route convergence process. A detailed proof of this theorem is presented in the Appendix.

3.3 Explicit information hiding using costs

The basic route propagation model described above is insufficient to achieve good scalability and isolation. To improve these two metrics, we need to perform *explicit information hiding* of routing updates. The basic philosophy is to *propagate a route update only when necessary.* We achieve this information hiding using the concept of *cost-hiding*. When an AS observes a cost increase or failure on the primary route R to a destination, it checks if it has an alternate route S with *comparable cost* to that of R. If so, it switches to the alternate route S and can potentially explicitly suppress the routing updates to neighboring AS's that this route switch may trigger. Here, we assume the cost of a route to be an additive metric and two routes are said to have a *comparable cost* if their cost difference is smaller than a cost-threshold Δ that is defined by the AS. The notion of *comparable cost* relaxes the notion of shortest path routing, and helps achieve better scalability and isolation.

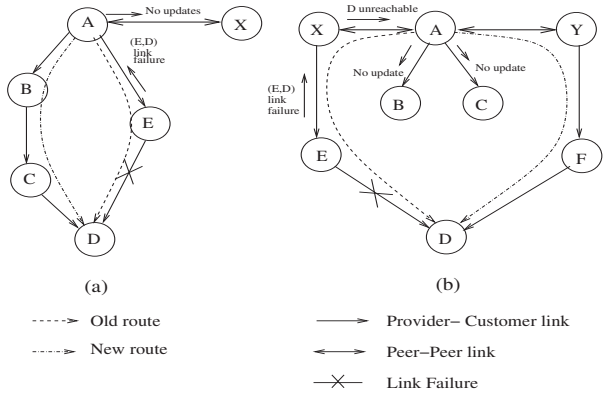

Figure 3: Two forms of cost-hiding.
(a) AS A chooses an alternate route within its own hierarchy.
(b) AS A chooses a route using an alternate peering link (A, Y) and hides the change from its customers.

One needs to be careful while using explicit information hiding for route update suppression. Whenever an AS suppresses a routing change to its neighbor (which routes through this AS), the routing state maintained by the neighboring AS becomes stale. If route suppression is not done correctly, the staleness it introduces can cause non-transient routing loops in the system. In HLP, we explicitly use the AS hierarchy and the route-preference guideline to avoid non-transient loops.

In HLP, we support three forms of cost-hiding: (a) not propagating minor cost changes (within a threshold Δ of previous advertised cost) of customer routes (previous AS in the path is a customer) across peering links; (b) not propagating minor cost changes of peer routes (previous AS in the path is a peer) to customers; (c) hiding the failure of one of multiple parallel peering links between a pair of AS's. The first two cases are illustrated in figure 3, and involve cost hiding by an AS higher up in the hierarchy that the origin of the change. In the third case, the issue is local to the two AS's, and it is entirely their own choice whether or not to advertise a cost change. We prove the following result on HLP's cost-hiding mechanism:

Theorem 2: *In the absence of a cycle in the AS hierarchy, if every AS strictly adheres to the route-preference guideline, then HLP with cost hiding is devoid of non-transient routing loops and the count to infinity problem.*

Similar to Theorem 1, the proof of this theorem relies on the non-increasing label property of HLP paths. The cost-hiding rules that we use in HLP do not introduce any loops since they preserve the non-increasing label property. We refer the reader to the Appendix for a detailed proof of this result.

3.4 Handling complex relationships

In practice, not all inter-AS relationships are purely provider-customer or peer-peer. Two examples of complex relationships between AS's are: (a) a *sibling relationship* between two AS's that are owned by the same administration; (b) two AS's intend to have different relationships for different destinations or at different geographic locations (e.g. provider-customer in Europe, peer-peer in US).

In HLP, we model all complex relationships as peer-peer links in the AS topology *i.e.,* every complex relationship is explicitly published as a peering link. The primary reason to do is, by treating these links as peering links, HLP emulates the behavior of BGP over these links thereby maintaining compatibility with what is status quo. Moreover, the AS's involved in a complex relationship need not reveal the nature of the relationship.

3.5 Handling policy variations as exceptions

The *common case of policies* in HLP assumes that the default behavior of all AS's follows the export-policy guideline and the route-preference guideline described in Section 2.2.2. An AS that intends to violate either of these two guidelines will trigger an *exception*. There are two forms of exceptions to the default behavior:

1. *Export policy exception:* An AS prefers to forward advertisements from one provider/peer to another provider/peer (except indirect peering which allows forwarding across peers).
2. *Prefer customer exception:* An AS prefers a non-customer route over a customer route.

These are the only forms of exceptions to the common case of policies as specified by the two guidelines. We will now first discuss the frequency of these exceptions before describing how we handle them in HLP.

3.5.1 Frequency of Exceptions

Policy exceptions are supposed to be *rare* events and the common case behavior of an AS should not be treated as an exception. For example, complex relationships should not be treated as an exception since they are explicitly advertised as peering links.

Type	Oct 15 2003	June 15 2003	Jan 9 2003
Prov-Prov	0.8%	0.1%	0.3%
Prov-Peer	0.5%	0.5%	0.4%
Peer-Prov	0.1%	0.1%	0.1%

Table 2: *Fraction of Internet routes under three-different types of export policy exceptions. Prov-Peer refers to the fraction of routes where an AS forwards announcements from a provider to a peer.*

We analyzed the frequency of exceptions using BGP routing table data from Routeviews [32] and RIPE [25] and measured the fraction of current BGP routes that violate the default behavior. As shown in Table 3.5.1 for three sample data-sets, we find that roughly 1% of the routes cause an export policy exception. We repeated the analysis across different time-periods and found comparable results. A recent work by Wang *et al.* [8] describes a mechanism for inferring the route preference policies of AS's. Their measurement study shows that most of the AS's prefer non-customer routes for less than 0.5% of destination prefixes. In summary, a very small fraction of Internet routes cause export-policy and prefer-customer exceptions.

3.5.2 Handling export policy exceptions

To violate the AS hierarchy and forward a route from a provider to a peer, an AS treats the provider-customer link as a peering link. In figure 4, AS D forwards routes from a provider C to a peer E. To do so, it converts the LSA into an FPV containing the path (D, C). In the general case, the FPV appears exactly as it would do if the adjacent provider-customer links (in this case only link (C, D)) had been peer-

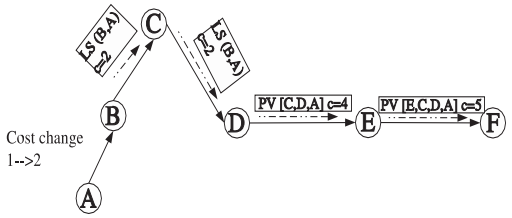

Figure 4: AS D forwards a route from provider C to peer E

ing links. This translates to the case of having an FPV traverse multiple peering links.

In a similar fashion, to forward an announcement from a peer/provider to a provider translates to treating the customer-provider link as a peering link. Hence, an FPV announcement from a peer/provider will be propagated to the provider with the path-vector in the FPV including all the three AS's involved in the exception.

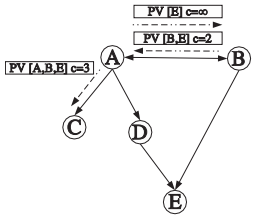

Figure 5: A wishes to choose a non-customer route to E

3.5.3 Handling prefer customer exceptions

Consider the figure 5 where AS A prefers to choose a non-customer route (using peering link (A, B)) over a customer route to destination E. To do so, A performs two operations. First, A propagates an exception to all its providers and peers withdrawing its customer route to E. Second, A propagates an FPV corresponding to the chosen non-provider customer route to its customers. In essence, these operations are equivalent to executing HLP in the case where the customer E did not exist in A's hierarchy. One example of a prefer-customer exception is the case of *backup* links to providers where an AS intends to use these links only during failure scenarios.

To summarize, HLP supports exceptions in the following manner: *any network that chooses to forward a route in violation of the constraints on a provider-customer link should model the link as a peering link (with regards to this route) and use the normal HLP propagation rules.*

4 HLP protocol analysis

In this section, we analyze the scalability, isolation and convergence properties of HLP. In this analysis, we explicitly assume that all AS's follow the default policy behavior and there are no exceptions. Based on our analysis, we show four important results. First, using *explicit information hiding* coupled with AS-level routing helps in achieving a 400 fold reduction in the churn rate incurred in HLP in comparison to BGP. Second, for routing events along 50% of inter-AS links, HLP can isolate an event to a region 100 times smaller than that of BGP (Section 4.1.2). Third, as the level of multi-homing increases, the churn and isolation factors significantly improve (Section 4.1.3). Finally, HLP significantly improves the worst-case convergence time over BGP by explicitly constraining the length of FPVs in HLP (Section 4.2).

4.1 Scaling and Isolation

To quantify the scaling and isolation aspects of HLP and compare them with BGP, we need a mechanism to analyze the routing dynamics of both protocols given the precise location and type of a routing event. However, given the complexity and generality of BGP policies, a precise modeling of BGP's routing dynamics is a challenging problem. We first describe our route-update emulation methodology which takes a conservative approach towards addressing this challenge. We later use this emulator to compare the scalability and isolation analysis of HLP and BGP.

4.1.1 Route-update Emulation Methodology

In our conservation approach towards modeling BGP dynamics, we assume that the policy behavior of every AS strictly adheres to the common case behavior based on the export-rule and prefer-customer guidelines described in Section 2.2. Based on this assumption on policies, we built a *route update emulator*, the goal of which is to precisely track the routing updates triggered by a single event. This emulation represents a lower bound on the churn-rate triggered in BGP since it does not model several intermediary states of path exploration in BGP. Hence, the churn improvement numbers we report (*i.e.,* HLP churn/ BGP churn) represent a lower bound on the actual churn improvement.

The input to the emulator is an AS topology and the set of inter-AS relationships. We model each AS as a single entity and for simplicity, we consider only two types of relationships in the emulator: provider-customer and peer-peer. Associated with each AS is a set of prefixes owned by the AS.

To compare the scalability and isolation of HLP and BGP, we restrict our analysis to inter-AS link failures. While several other types of events are possible, an inter-AS link failure (or a BGP session reset) triggers the maximum amount of churn in BGP since it simultaneously affects routes to several prefixes. We quantify *isolation* as the number of AS's that can potentially be affected by a routing event[5] and *churn* using the total number of updates generated by an event. Given an

[5]Any AS that receives an update due to an event can potentially be affected by the event since the AS can modify its routing information based on the update.

inter-AS link in an AS topology, we emulate the route propagation behavior of HLP and BGP for each destination and compute the number of AS's that receive an update about the event in each case. Any AS that receives an update can potentially be affected by the event. The improvement in the isolation of HLP is defined as the ratio of the number of AS's affected by an event in BGP to the number of AS's affected in HLP.

4.1.2 Cost-hiding: best-case analysis

We quantify the effect of churn/fault isolation on a real Internet AS topology as gathered from RIPE [25] and Routeviews [32] containing 16774 AS's and 37066 inter-AS links. We emulate policy-based routing in BGP and compute the AS hierarchy based on the inference methodology presented in [28] to characterize links as either provider-customer or peer-peer. We randomly sample 10,000 inter-AS links and fail these links in our analysis.

Without making any assumptions on how inter-AS link-costs are assigned, we begin by analyzing the best-case of cost-hiding where we set the threshold for cost hiding in HLP to the best case *i.e.,* allow reroutes regardless of path cost. Later in Section 4.1.4, we describe the mechanism that we use to set the cost threshold to approximate the best-case scenario.

Churn Improvement: The churn reduction in HLP is due to two factors: (a) using the AS-prefix mapping; (b) cost hiding of route updates. The number of prefixes owned by a single AS is a measure of the gain that this mapping provides in reducing the churn in BGP. The mean gain, then, is the average number of prefixes owned by each AS. Based on the (AS, prefix) mapping we collected from Routeviews and RIPE, the mean gain is 7.8. We observed this mean gain to roughly be stable around $6 - 8$ over the last 3 years. This reduction does not include the additional savings due to the presence of sub-prefixes in BGP for traffic engineering purposes.

The effects of cost-hiding on the churn rate are illustrated in Figures 6(a) and 6(c). We make the following observations. First, on an average (assuming that every inter-AS link has an equal probability of failure), HLP incurs roughly 2% of BGP's churn which represents a factor 50 reduction in the net-churn rate. Overall, the net mean churn reduction due to cost-hiding and AS-prefix mapping is $390 = 7.8 \times 50$. Second, for roughly 50% of the inter-AS links (median churn ratio in Figure 6(c)), the churn incurred by HLP is nearly 75 times smaller than that of BGP. Third, the churn reduction of HLP is dependent on the type of inter-AS link that failed. Cost-hiding provides substantial churn reduction for multi-homed customers (due to the presence of alternative paths) but provides no churn reduction for singly-homed customers (due to lack of alternative paths).

Isolation improvement: Figures 6(b) and 6(c) show the magnitude of isolation and the isolation improvement achieved in HLP. Recall that isolation is measured by the number of AS's that are affected by a single event. We make the following observations from our analysis. First, in the aggregate case, we found that for 50% of links, HLP has more than a 100 fold improvement in isolation over BGP. Second, more than 80% of the events are globally visible in BGP. In comparison, more than 40% of the events trigger updates to less than 10 AS's in HLP. This is because the level of isolation is dependent on the type of inter-AS link that underwent a failure.

Overall, in the best case scenario for cost hiding: (a) the mean churn rate of route advertisements is roughly reduced by a factor of 390 in comparison to BGP. (b) for roughly 50% of the inter-AS links, HLP is able to isolate the location of a fault to a region roughly 100 times smaller than that of BGP. We repeated the analysis over several AS topologies ranging from 2002 to 2004 and noticed similar numbers for the reduction in churn and isolation. Hence, the scale and isolation gains in HLP are substantial.

4.1.3 Effect of Multi-homing

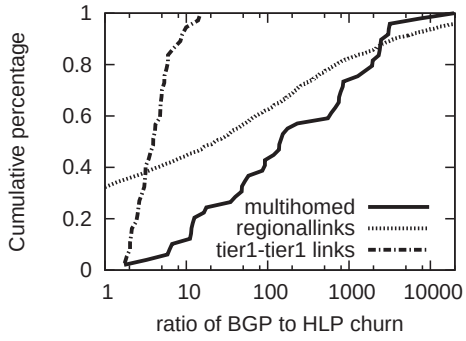

Figure 7: Churn: Comparing the churn reduction factor of HLP for different types of inter-AS link events.

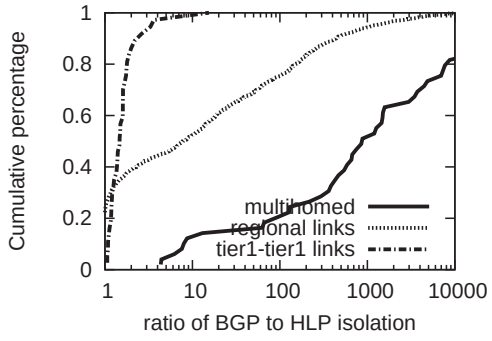

Figure 8: Isolation: Comparing the isolation improvement factor of HLP for different types of inter AS links.

As the level of multi-homing increases in the Internet, we observe the scale and isolation properties to further significantly improve in HLP. This phenomenon is illustrated in Figures 7 and 8 which show the distribution of the churn

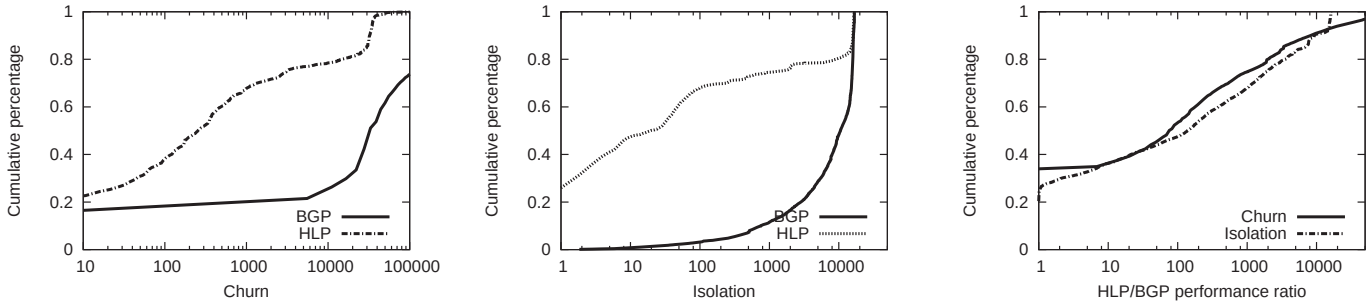

Figure 6: *(a) Churn: CDF of the number of route updates generated by a single event in HLP and BGP. (b) Isolation: CDF of the region of visibility (measured in number of AS's) of the effects of a single routing event in HLP and BGP. (c) CDF of the churn improvement ratio and isolation ratio of HLP in comparison to BGP.*

and isolation factor for events along different types of links. The median (50th percentile) churn reduction factor and isolation factor for multi-homed customer links are roughly 200 and 1000 respectively. In comparison, the isolation and churn savings on tier-1 ISPs are relatively small since these links tend to break many paths and are much harder to hide.

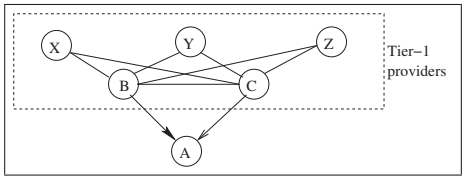

Figure 9: An example topology to illustrate the isolation properties of a multi-homed AS.

If all stub networks in the Internet were multi-homed, then we would notice substantial gains in the overall churn and isolation properties. To better explain this phenomenon, consider the simple topology in Figure 9 where AS A is multi-homed to two tier-1 providers B and C. When the link (A, C) fails, C chooses the alternative route through B of comparable cost (since it would have received an announcement from B earlier) and withdraws its previous route announcement (C, A) from all its peers B, X, Y, Z. In this case, all the peers of C automatically switch to the route through B. In this entire process, none of the AS's propagate any updates to their customers and information of the event is hidden from the rest of the Internet. Extending this simple example, many tier-1 and tier-2 AS's typically have multiple routes (of comparable cost) through different peers to a multi-homed customer. When one of these paths fails, each tier-1/tier-2 AS automatically switches to the other path without triggering any new updates. Hence, very few AS's (apart from top-tier AS's) are notified of a path failure to a multi-homed customer.

Max. Hop-Length difference	Cumulative Probability
0	42.9%
$<= 1$	89.7%
$<= 2$	99.6%
$<= 3$	99.9%
$<= 5$	100%

Table 3: Cumulative distribution of the maximum hop-length difference between the shortest (hop-length) primary route and a secondary route (both of them obeying the default policy behavior).

4.1.4 Determining the cost-hiding threshold

Now, we describe a simple rule of thumb for determining the cost hiding threshold (denoted by Δ) such that HLP can approximately achieve the scale and isolation properties that is achievable in the best-case scenario.

Note that, when AS's assign costs to routes there must be some cost-standard to determine meaningful cost values. The minimum requirement should be that AS's at least allocate link costs from a *common cost-range*. Without loss of generality, let us assume that all AS's use a common cost-range, say $[0 \ldots m]$ for some value m.

The ability to use cost-hiding to improve the scale and isolation properties is dependent on the presence of an alternative route of *comparable cost*. Given a common cost-range, the *cost difference* between two routes to the same destination is dependent on the *hop-length difference* between the two paths. As shown in Table 3, we find that for nearly 99.6% of destination AS's, the hop length difference between the primary and secondary routes is at most 2 and for 90% the difference is at most 1.

To preserve the scale and isolation properties for a majority of Internet routes, we need to pick a threshold Δ that can offset the sum of the cost of 2 inter-AS links. By doing so, we can approximate the scale and isolation results achieved in the best-case cost hiding (Section 4.1.2) for more than 99%

of routes. For example, in the simple case where inter-AS links are assigned uniformly in the range $[0 \ldots m]$, choosing a cost-hiding threshold of $\Delta = m$ achieves the desired result. In the general case, a simple thumb-rule is to set $\Delta = 2 \times \mu$ where μ is the mean inter-AS link-cost.

4.2 Convergence properties

We define *convergence time* as the interval of time (assuming certain propagation delays along the links) it takes the entire network to reflect a particular route change, *e.g.,*a new route becomes available, a route has disappeared, or a route has changed. To study the protocol convergence characteristics, we use the model introduced by Labovitz *et al.* [18]. The only difference is that instead of a fully-connected mesh we assume a hierarchical topology with n nodes (AS's) that reflects the topology enforced by HLP.

We make three simplifying assumptions in our analysis. First, similar to previous works [18, 23], we model an AS as a single entity though the underlying AS may comprise of several routers. This assumption holds because from the perspective of HLP, if all routers adhere to the route propagation rules, the behavior of all routers in unison presents a consistent view of the AS.[6] Second, we model the route propagation delay within an AS to be a constant value, the assumption being that the ratio of the propagation times across different AS's is a constant factor. Finally, we do not consider any form of flap dampening to be activated that may affect convergence.

We prove the following result on convergence:

Theorem 3: *For a given destination D, let $k(D)$ represent the maximum number of peering links in any HLP route advertisement to destination D. Under the assumption that every AS adheres to the HLP route propagation rules, if an event E affects destination D, then the route updates to D triggered by event E will converge within a maximum time of $O(n^{k(D)})$.*

We refer the reader to the Appendix for a detailed proof. In comparison, Labovitz *et al.* [18] showed that the worst case convergence time of BGP in a n-node fully connected graph is $O((n-1)!)$. $k(D)$ represents the maximum length of an FPV (number of peering links) in an HLP route to D. The primary reason for the convergence improvement in HLP is that HLP *explicitly constrains* the value of $k(D)$ (an AS that has a customer route) while BGP places no such restrictions. The primary take-away of this result is: *The value of $k(D)$ is at most 2 for 99% of Internet routes thereby providing a quadratic worst-case convergence time for these destinations.* In fact, $k(D) \leq 1$ for 90% of destinations [28, 9] thereby providing linear-time convergence. The maximum value of $k(D)$ we observed for any route was 4.

[6] Though two routers within the same AS may end up advertising different routes to a destination, the two routes will be consistent in that, if one route is a customer route, the other one should also be a customer route.

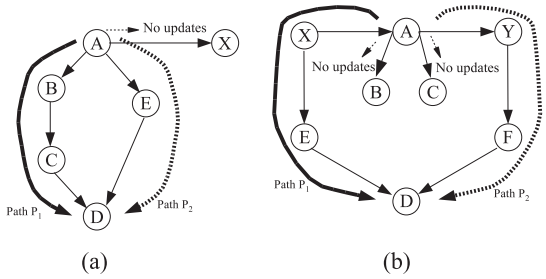

Figure 10: AS A can select two separate routes to destination D without triggering updates to its neighbors.

5 Traffic Engineering and Policy Support

Policy routing and traffic engineering (TE) are interrelated. Although BGP was never designed to do traffic engineering, it is frequently used that way. When considering any alternative routing protocol, it is important to understand the way that the traffic engineering options are affected.

HLP can in fact support most of the commonly used BGP traffic engineering practices and policies, while maintaining the basic scalability, isolation and convergence advantages. In this section, we describe a set of TE mechanisms that can be incorporated into HLP to provide: (a) AS's the flexibility to perform *prefix-level route selection* (Section 5.1); (b) AS's the ability to achieve *inbound traffic engineering* by manipulating link-costs (Section 5.2). (c) a destination AS the ability to do relatively infrequent *prefix deaggregation/aggregation* for TE purposes. Unlike BGP[7], the mechanisms we describe below trigger very few route updates. Finally, we conclude in Section 5.3 by comparing and contrasting the policies supported by BGP and HLP.

5.1 Prefix-level route selection

Prefix-level route selection is a traffic engineering mechanism commonly used with BGP, whereby an AS can independently choose different routes to different prefixes owned by the same destination AS. This is easy with BGP's prefix-based routing, but HLP does AS-based routing, so this seems at first to be problematic. However, we can use HLP's *information hiding* to support prefix-level route selection, without even requiring any addition route update messages. The example topologies in figure 10 illustrate this process. In both cases, AS A has two distinct routes to destination D which are of comparable cost. Using information hiding, AS A has the flexibility of picking either of these two routes without needing to inform its neighbors of its choice. If D advertises two distinct prefixes in the (AS, prefix) mapping table,

[7] Prefix deaggregation/aggregation can trigger several unnecessary route updates in BGP.

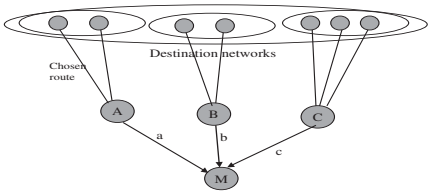

Figure 11: An example of a multi-homed customer M with three providers A,B,C and the cost-knobs (a, b, c) that M can set.

then A can choose to route the two prefixes independently even though they have the same origin AS. However, HLP's default preference for customer routes places one constraint: An AS that has a customer route to a destination can perform prefix-level route selection only within its available choice of customer routes (and cannot choose a non-customer route).

Static prefix deaggregation: A destination AS that intends to perform inbound traffic engineering for different sub-prefixes needs to explicitly deaggregate its prefixes into sub-prefixes and publish these sub-prefixes in the (AS, prefix) mapping table. Given this information, any AS that has multiple routes to this destination AS can choose a *separate route for each sub-prefix*. By maintaining this deaggregation mapping to be relatively static, HLP avoids the unnecessary routing dynamics (route withdrawals, new advertisements for sub-prefixes) triggered by prefix deaggregation in BGP.

5.2 Cost-based inbound traffic engineering

AS-prepending is another BGP traffic engineering technique, whereby a routing domain prepends its own AS number multiple times to the AS Path it advertises to a neighbor to make the link less preferred. This is a very crude technique, but very commonly used.

In HLP, this process is more explicit: an AS can manipulate the cost of its inter-AS links to achieve volume-based inbound TE. This works under the assumption that most AS's choose routes based on cost as currently happens with BGP's AS Path length. Figure 11 illustrates the process; the multi-homed stub network M has the flexibility of setting the costs (a, b, c) to its providers to influence the inbound traffic flow. The degree of control exerted by M on the incoming traffic from source AS X is then determined by the differences in cost of paths between the provider networks (A, B and C) and X. If the cost difference between these paths can be offset by suitably setting (a, b, c), M can choose the provider through which traffic from X will be routed. This cost difference between these paths is dependent on the differences in their hop-lengths. In a prior analysis (refer to Table 3 in Section 4.1.4), we showed that in nearly 90% of Internet paths the hop-length difference between the shortest and second-shortest path to a destination is at most 1 (99.6% for a difference of at most 2). Hence, we believe that M can perform fine-grained control by manipulating costs to its upstream providers. However, the level of such control is dependent on the underlying distribution of link-costs and we intend to evaluate this more completely in future work.

5.3 Policy practices: HLP vs BGP

BGP practices that carry over: Several existing TE practices in BGP can be directly carried over without much modifications to HLP. We cite three specific examples of commonly used practices today. First, since HLP is meant as a replacement to eBGP, it does not affect most of the intra-domain traffic engineering practices (*e.g.,* hot-potato routing) unless these conflict with HLP's inter-domain routing policy. Second, existing proposals to perform traffic-engineering across multiple peering links between two domains using MEDs or negotiated routing [21], can also be naturally extended to HLP. Third, HLP can also support *community attributes*, a feature used by customer AS's to signal specific policies to upstream providers.

BGP policies not supported by HLP: While HLP can support a variety of BGP policies, there are certain corner-case policies that it cannot. Two examples of such policies are preference rules based on generic *regular expressions* on the path-vector, and import rules using *negation-based expressions* on the path-vector. Due to information hiding, the entire routing path to a destination AS is not visible in HLP and this limits the ability to use generic regular-expression-based policies in HLP. However, in many cases, given the AS hierarchy information and the FPV to a destination, an AS can roughly construct the structure of the complete path. While tier-1 and tier-2 AS's can reconstruct paths with high accuracy, this ability decreases for AS's which are lower in the AS hierarchy. An example of a negation based policy is *avoiding other transit networks* where an AS X intends to avoid routes through a specific AS Y. Among the top-tier ISPs, the FPV in HLP routes should provide sufficient information to enforce this policy. But for AS's lower in the hierarchy, it becomes harder to enforce this policy. Our discussions with tier-1 ISPs seem to indicate that negation based policies are not very common.

HLP specific policies: HLP also enables a new set of policies. Two such policies are class based routing and cost-based routing. HLP can support different classes of routes much like Differentiated Services in the Internet - *e.g.*, customer-class, peer-class, provider-class. Unlike BGP, one can make these class definitions transitive in HLP across multiple inter-AS links. Moreover, HLP allows AS's to specify costs on links and provides the ability to set policies based on these link costs (as shown in Section 5.2).

6 HLP: A router perspective

Until now, our description of HLP focused on treating an AS as a single entity. However, given that an AS can comprise of hundreds of routers, we require an internal HLP protocol, *iHLP*, to ensure that all routers within an AS behave in a manner consistent with the HLP protocol. While a complete design of iHLP is outside the scope of this paper, we describe the basic consistency checks that iHLP should enforce within an AS (Section 6.1). Next in Section 6.2), we describe a router-level of HLP on top of the XORP software router platform [2], present micro-benchmarks on the overhead of processing updates and explain the implementation lessons we learnt.

6.1 iHLP consistency checks

The basic design of internal BGP (iBGP) is not completely compatible with our HLP protocol design. To make routers within an AS act in a manner compatible with HLP, we need an internal HLP (iHLP) protocol to enforce four consistency checks:

Maintaining a communicating group: Since routers within an AS need to act in unison, every router should maintain a *communicating group* of live routers within the AS. Unless an AS partitions, an AS should have only one communicating group. If it does partition, iHLP should ensure that routers in each communicating group act in unison; however each communicating group will act on its own.

Maintaining customer-route consistency: HLP follows the prefer-customer guideline. To implement this, iHLP needs to ensure that in the absence of exceptions, every router should choose a customer route to a destination provided one such exists. An exit-router that raises a prefer-customer exception merely withdraws its current route. In the absence of an alternate customer route, the destination is classified as an exception; otherwise, it is not.

Maintaining link-cost consistency: From an external view, every inter-AS link is associated with a specific cost. In the presence of multiple peering links with a neighboring AS, iHLP needs to ensure a common cost value across all routers for a peering link. We do not impose any restrictions on how to compute this common cost value.

Maintaining route-update consistency: All routes propagated by AS X about a destination AS D to a neighboring AS Y should satisfy: (a) All announcements about D to Y should be of one kind: LSAs or FPVs; (b) All LSAs for the same link should have the same cost.

One simple way to implement these consistency checks would be to flood HLP messages to all routers within the AS. Alternatively, one could use a centralized routing control platform (RCP) as envisioned by Caesar *et al.* [4].

6.2 HLP software router implementation

HLP has been implemented as a module that fits in the eXtensible Open Router Platform (XORP) software router [2]. Our current implementation supports all the features in the basic HLP design including many of the policy extensions such as exceptions and backup links.

Hierarchy Size	100	300	500	700	1000
LSA proc. time	0.0052	0.0153	0.0252	0.037	0.052

Table 4: *AS hierarchy size vs LSA processing time (sec).*

# of AS's	1000	5000	10000	15000	20000
FPV proc. rate	5270	3154	1989	1452	1132

Table 5: *# of AS's vs FPV processing rate (updates/sec)*

6.2.1 Overhead characteristics

Two of the most common operations in a HLP router are the processing of LSA updates and FPV updates. Tables 4 and 5 illustrate the LSA processing time and FPV throughput that can be obtained from our implementation. These measurements were performed on a 2.4 GhZ Intel processor with 1 GB memory. Though our implementation has not been optimized for performance, these measurements indicate that a naive implementation can handle today's BGP workload. First, the complexity of LSA processing due to the recomputation of shortest paths to destinations and our numbers match with those reported in prior OSPF studies [27, 4]. However, in reality, we anticipate the number of LSAs within a given second to be very small since each event (*e.g.*, link failure) is captured within a single LSA message (unlike BGP which generates many updates due to a single event). Also, we anticipate the frequency of link-cost changes to be small *e.g.*, as a comparison, a stub network that continuously deaggregates prefixes propagates at most one update every 30 seconds [5]. Second, the FPV processing rate that we can support for a hierarchy of size 20000 is at least 10 times greater than the maximum update rate observed in a BGP router today [1].

6.2.2 Implementation lessons

The experience of building a full-fledged prototype of HLP made us carefully think through the router level behavior of HLP and determine how it differs from BGP and what additional mechanisms a HLP router requires. Three implementation aspects are noteworthy. First, the base code of HLP has many similarities to BGP and reuses more than 90% of the XORP BGP implementation. This illustrates the ease of implementing HLP using existing BGP code. Second, to make HLP work at the router level, we had to revisit the HLP design and define the necessary set of consistency checks required in iHLP. Third, one had to be careful in handling ex-

ceptions at the router level especially to avoid exception inconsistencies between routers *e.g.,* one router chooses a customer route while another does not. This requires a router to keep track of the exceptions in other routers and declare an exception only in the presence of consensus.

Transition from BGP to HLP: One of the important lessons that arose from the implementation is a simple transition plan from BGP to a simplified version of HLP which merely changes the current operational practices of BGP. The idea is to setup a two level hierarchy separating transit networks and stub networks into different levels and using only the transit-stub links as the set of published provider-customer links. Using BGP routing information, every AS can independently infer this two-level hierarchy with high accuracy [28, 9, 20]. In this hierarchy, links between transit networks execute BGP (this models the FPV aspect of HLP) and all transit networks are required to install filters to allow a stub network only to originate route announcements and not act as transit networks. Stub networks can signal link costs using AS path prepending.

This simple deployment provides several benefits. First, since stub networks account for a majority of AS's and their growth rate is higher than that of transit networks, this separation does provide improved scalability and isolation properties than BGP. Second, we minimize the possibility of misconfigurations since many stub networks are largely unmanaged and we minimize the knobs at the disposal of these networks. Third, routes between transit networks are relatively stable [5] and this stability cannot be affected by stub AS's.

7 Related work

We classify related work into two categories:

New Internet architectures: Several new Internet architectures have been proposed to address pressing problems in Internet routing. The Newarch project proposed NIRA [33] which advocates better end-host control over routing. In NIRA, an end-host has the ability to choose the sequence of providers its packets traverse. Feedback based routing [34] is an alternate design for replacing Internet routing where edge network compute an approximate topology map of the Internet at an AS-level based on measurement-based feedback from the network. It then uses this state to compute the shortest path and source routes the packets by encapsulating the route in the packet. While many such routing architectures exist outside the realm of BGP, the primary motivation for our work is the search of a protocol that completely retains the operational and economic model of BGP but only alters the route propagation model to address the pressing deficiencies of BGP. In this regard, the proposed set of guidelines for next-generation routing protocols from IRTF [15] provided a good starting point in our design.

Changes to the BGP protocol: HLP aims to provide improved scalability/isolation, diagnosis support, convergence and security. There have been several works that have proposed incremental changes to BGP to achieve these ends. Afek *et al.* [3] propose grouping prefixes with similar behavior into *policy atoms*, which can be dealt with as an aggregate in an attempt to reduce overhead. BGP-RCN [24] is a proposed modification that embeds BGP updates with the location where updates are triggered; this information can be used to greatly improve both the convergence and diagnosis properties of the protocol. BGP's poor convergence properties come from a variety of causes [22]. Worse still, certain combinations of BGP policies lead to divergence [12]. The authors of [10] propose a set of policy guidelines that guarantees routing convergence. Secure-BGP [16] and Secure-origin BGP [26] are two well known proposals to improve the security of BGP. Subramanian *et al.* [29] and Chu *et al.* [14] have recently proposed two alternative mechanisms to improve the security of BGP while alleviating some of the deployment hurdles of a PKI.

8 Conclusions

Designing an inter-domain routing protocol is a very challenging task, in part because it is difficult to design routing systems that scale globally, and in part because the range of policies that needs to be supported is ill-defined. BGP, a specific point in the design space supports a wide range of policies at the expense of poor scalability, fault isolation and convergence properties. In designing HLP, we started from the observation that to do better than BGP, we needed to make use of information that BGP does not have. The only policy information that inherently does not suffer from serious privacy issues is that of provider-customer relationships, and this is simply because this information cannot be a secret for routing to function. Having this information available in the protocol itself led directly to the observation that in sizable parts of the AS hierarchy we could use link-state style algorithms, which solve many of the problems exhibited by BGP. But between these regions, link-state is not viable due to policy-privacy issues, which forces us towards a hybrid link-state/path-vector solution. Separation of prefix-binding from topology discovery is a further step towards reducing routing protocol overhead, and also towards using appropriate security solutions for the different parts of the problem. The resulting protocol has many very desirable properties, including fast convergence, good fault isolation, lower routing table churn, and inherently better security and robustness.

However, just because a protocol has good *routing* properties does not mean that it solves the problem in a way that is *economically* viable for ISPs. Unfortunately no single person knows the big picture of what an inter-domain policy routing protocol needs to do in reality. This makes evaluation especially hard. In this paper we have tried to examine not only the basic properties of convergence, fault isolation and scalability, but also examine many of the ways BGP is

used with the aim of understanding how well HLP solves the same tasks. Our present level of understanding is that HLP measures up rather well against BGP in a large range of deployment scenarios, but only by exposing the design to a wide range of ISPs and router vendors will we learn the full story.

We are under no illusion that HLP is poised to replace BGP any time soon, but only by putting a stake in the ground can we hope to stimulate informed debate about both the requirements for and the design of future inter-domain routing.

References

[1] `http://ipmon.sprint.com`.

[2] The eXtensible Open Router Platform (xorp). `http://www.xorp.org`.

[3] AFEK, Y., BEN-SHALOM, O., AND BREMLER-BARR, A. On the structure and application of BGP policy Atoms. In *IMW* (2002).

[4] CAESAR, M., CALDWELL, D., FEAMSTER, N., REXFORD, J., SHAIKH, A., AND VAN DER MERWE, J. Design and implementation of a routing control platform. *ACM/USENIX NSDI* (2005).

[5] CAESAR, M., SUBRAMANIAN, L., AND KATZ, R. H. Root cause analysis of Internet routing dynamics. Tech. rep., U.C. Berkeley UCB/CSD-04-1302, 2003.

[6] CHANG, D.-F., GOVINDAN, R., AND HEIDEMANN, J. The temporal and topological characteristics of BGP path changes. In *Proc. International Conference on Network Protocols* (2003).

[7] FEAMSTER, N., BORKENHAGEN, J., AND REXFORD, J. Guidelines for interdomain traffic engineering. *ACM Computer Communication Review* (2003).

[8] F.WANG, AND GAO, L. Inferring and Characterizing Internet Routing Policies. In *Proceedings of ACM IMC 2003* (2003).

[9] GAO, L. On inferring automonous system relationships in the internet. *IEEE/ACM Trans. Networking* (to appear 2004).

[10] GAO, L., AND REXFORD, J. Stable Internet routing without global coordination. In *Proc. ACM SIGMETRICS* (2001).

[11] GRIFFIN, T. What is the Sound of One Route Flapping?, 2002. IPAM talk.

[12] GRIFFIN, T. G., SHEPHERD, F. B., AND WILFONG, G. Policy disputes in path vector protocols. In *Proc. International Conference on Network Protocols* (1999).

[13] GRIFFIN, T. G., AND WILFONG, G. An analysis of BGP convergence properties. In *Proc. ACM SIGCOMM* (1999).

[14] HU, Y. C., SIRBU, M., AND PERRIG, A. Spv: Secure path vector routing for securing BGP. In *Proc. ACM SIGCOMM* (2004).

[15] IRTF ROUTING RESEARCH GROUP. http://www.irtf.org/rrg/.

[16] KENT, S., LYNN, C., AND SEO, K. Secure Border Gateway Protocol (Secure--BGP). *IEEE Journal on Selected Areas of Communications 18*, 4 (April 2000), 582–592.

[17] S. Kent, C. Lynn, J. Mikkelson, and K. Seo. Secure Border Gateway Protocol (S-BGP) – Real World Performance and Deployment Issues. In *Proc. of Network and Distributed System Security Symposium, (San Diego, California)*, 2000.

[18] LABOVITZ, C., AHUJA, A., BOSE, A., AND JAHANIAN, F. Delayed Internet Routing Convergence. In *Proc. ACM SIGCOMM* (2000).

[19] LABOVITZ, C., MALAN, R., AND JAHANIAN, F. Origins of Internet routing instability. In *Proc. IEEE INFOCOM* (1999).

[20] MAHAJAN, R., WETHERALL, D., AND ANDERSON, T. Understanding BGP Misconfigurations. In *Proceedings of ACM SIGCOMM* (2002).

[21] MAHAJAN, R., WETHERALL, D., AND ANDERSON, T. Negotiation-based routing between neighboring domains. In *Proceedings of ACM/USENIX NSDI* (2005).

[22] MAO, Z. M., BUSH, R., GRIFFIN, T. G., AND ROUGHAN, M. BGP beacons. In *Proc. ACM Internet Measurement Conference* (2003).

[23] MAO, Z. M., GOVINDAN, R., VARGHESE, G., AND KATZ, R. Route flap damping exacerbates Internet routing convergence. In *Proc. ACM SIGCOMM* (2002).

[24] PEI, D., AZUMA, M., NGUYEN, N., MASSEY, J. C. D., AND ZHANG, L. Bgp-rcn: Improving bgp convergence through root cause notification. *Technical Report, UCLA CSD TR-030047* (2003).

[25] RIPE'S ROUTING INFORMATION SERVICE RAW DATA PAGE. http://data.ris.ripe.net/.

[26] SECURE ORIGIN BGP (SOBGP). ftp://ftp-eng.cisco.com/sobgp.

[27] SHAIKH, A., AND GREENBERG, A. OSPF Monitoring: Architecture, Design and Deployment Experience. In *Proceedings of ACM/USENIX NSDI* (2004).

[28] SUBRAMANIAN, L., AGARWAL, S., REXFORD, J., AND KATZ, R. H. Characterizing the Internet Hierarchy from Multiple Vantage Points. In *Proceedings of IEEE INFOCOM* (2002).

[29] SUBRAMANIAN, L., ROTH, V., STOICA, I., SHENKER, S., AND KATZ, R. Listen and Whisper: Security Mechanisms in BGP. In *Proceedings of ACM/USENIX NSDI* (2004).

[30] R. Thomas. `http://www.cmyru.com`.

[31] Trends in dos attack technology. `http://www.cert.org/archive/pdf/DoS_trends.pdf`.

[32] UNIVERSITY OF OREGON ROUTEVIEWS PROJECT. http://www.routeviews.org/.

[33] YANG, X. Nira: a new internet routing architecture. *ACM SIGCOMM FDNA workshop* (2003).

[34] ZHU, D., GRITTER, M., AND CHERITON, D. Feedback based routing. *ACM Hotnets Workshop* (2002).

A Proof of Theorem 1

This proof makes the obvious assumption that there is no loop in the provider-customer hierarchy. Consider the following labeling process: Associate a label 3 with any customer-provider link that appears along a path, a label 2 to a peering link and a label 1 to a provider-customer link. Any routing path from a source AS S to destination D will be associated with a string of labels depending on the type of links traversed. The HLP propagation rules ensure that the labels of any valid routing path is always *non-increasing* (*i.e.*, once a path traverses a peering link, one can only use peering links or provider-customer links).

We prove the non-existence of a loop by contradiction. Assume that a non-transient routing loop $L = (X_1, X_2, \ldots X_k)$ exists where for some destination AS D, all the AS's X_i's set their next hop to X_{i+1} (X_n sets next hop to X_1). Hence, for any $1 \leq i \leq n$, the loop-path $L_i = (X_i, X_{i+1}, \ldots X_n, X_1, \ldots X_i)$ should be a valid HLP path. The non-increasing label property of HLP paths implies that the label of all the edges of the loop has to be the same. If any of the links in the loop is a customer-provider link or a provider-customer link, then it implies the existence of a loop in the provider-customer hierarchy which is a contradiction. The only remaining case is when all links in the loop are peering links. In this scenario, any route advertisement

that traverses the loop should contain the entire set of peering links to be appended in the FPV. A loop in this case is infeasible since the FPV propagation rules of HLP does not permit a path-vector announcement to traverse a loop. Hence by contradiction, no valid loop L exists.

Extending the argument, a count to infinity problem arises only in the presence of a non transient loop in the route propagation process. Therefore, the basic protocol is void of the count of infinity problem.

B Proof of Theorem 2

Clearly, hiding the failure of one of multiple peering links between a pair of AS's cannot cause routing loops. To analyze the other two forms of cost-hiding, we revert back to the link-labeling process used in the proof of Observation 1. Given any valid AS path P_{XD} from an AS X to a destination D, the reverse path P_{XD}^{-1} represents the path through which the route update propagated from D to X. For any AS path P_{XD}, we define the *class* of a routing path to be one the minimum label of a inter-AS link along the reverse path P_{XD}^{-1} (*e.g.*, Any route advertisement that traverses only customer-provider links has class 3, any route advertisement that traverses at least one peering link or down the AS hierarchy has class 2, 1 respectively). Now, the cost-hiding rules of HLP, satisfies the property that a node only hides "higher" class routes across lower-class links. The first cost-hiding rule can be translated as: any route update that is hidden across a peering link is a class 3 route advertisement. Similarly, any route update hidden from a customer is at least a class 2 route.

In other words, HLP with the cost hiding rules still maintains the *non-increasing* label property of routing paths, the only difference being that the actual path may be hidden. Hence, by reusing the argument in Observation 1, HLP with cost hiding is void of non-transient loops.

C Convergence Analysis: Proof of Theorem 3

We define *convergence time* as the interval of time (assuming certain propagation delays along the links) it takes the entire network to reflect a particular route change, *e.g.*,a new route becomes available, a route has disappeared, or a route has changed. To study the protocol convergence, we use the model introduced by Labovitz *et al.* [18]. The only difference is that instead of a fully-connected mesh we assume a hierarchical topology with n nodes (ASes) that reflects the topology enforced by HLP.

We make three simplifying assumptions in our analysis. First, similar to previous works [18, 23], we model an AS as a single entity though the underlying AS may comprise of several routers. This assumption holds because from the perspective of HLP, if all routers adhere to the route propagation rules, the behavior of all routers in unison presents a consistent view of the AS.[8] Second, we model the route propagation delay within an AS to be a constant value, the assumption being that the ratio of the propagation times across different ASes is a constant factor; however, the overall time to convergence varies with the size of the network (depends on n). Finally, we do not consider any form of flap dampening to be activated that may affect convergence.

Labovitz *et al.* [18] show that the worst case convergence time of BGP in a n-node fully connected graph is $O((n-1)!)$, as there exists $O((n-1)!)$ distinct paths to reach a destination. In comparison, we demonstrate the following result for HLP:

HLP convergence result: For a given destination D, let $k(D)$ represent the maximum number of peering links in any HLP route advertisement to destination D. Under the assumption that every AS adheres to the HLP route propagation rules, if an event E affects destination D, then the route updates to D triggered by event E will converge within a maximum time of $O(n^{k(D)})$.

$k(D)$ represents the maximum length of an FPV (number of peering links) in an HLP route to D. The primary reason for the convergence improvement in HLP is that HLP *explicitly constrains* the value of $k(D)$ (an AS that has a customer route) while BGP places no such restrictions. In a fully connected mesh of peers, HLP does not explore paths of varying lengths if the destination exists in the AS hierarchy of direct peers while BGP performs path exploration. We note that for nearly 90% of the Internet routes, the value of $k(D) = 1$ [28, 9], implying *linear convergence* time of HLP for a majority of routes. For more than 99% of the routes, the value of $k(D)$ is at most 2 which implies *quadratic* convergence. Based on the analysis of BGP updates using inferred provider-customer relationships [28, 9], the maximum value of $k(D)$ we observed across all destinations is 4. In general, we anticipate the value of $k(D)$ to remain constant as the Internet evolves in the future.

C.0.3 Proof of HLP convergence

We separate our analysis into two cases: (a) single peering link ($k(D) = 1$); (b) multiple peering links ($k(D) > 1$).

Single peering link: When $k(D) = 1$ all HLP routes traverse only a single peering link. The analysis of convergence time for $k(D) = 1$ uses three key ideas: (a) link state announcements within a hierarchy converge in linear time; (b) propagation of route changes from a provider downstream within a hierarchy takes linear time; (c) the FPV convergence process takes linear time since an AS that chooses a peering link needs to determine which direct peer to choose from for

[8] Though two routers within the same AS may end up advertising different routes to a destination, the two routes will be consistent in that, if one route is a customer route, the other one should also be a customer route.

a given destination.

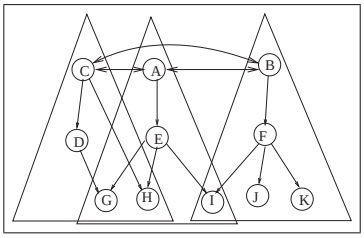

Figure 12: Example: Single peering link analysis.

Consider an example topology consisting of different hierarchies as illustrated in Figure 12 (since every path strictly belongs to within two hierarchies with a single peering link). In this topology, the primary type of event we consider is a cost change along any of the links (link failures can be modeled as a cost change). We consider cost changes along two types of links: (a) provider-customer links; (b) peering link. In case (a), when a link within A's hierarchy, say (A, E) undergoes a cost change, three steps of actions are triggered. First, all ASes within A's hierarchy are propagated a route advertisement using an LSA. Second, along each peering link, A propagates a separate advertisement for each destination within its hierarchy affected by the cost change. Finally, each peer B searches for alternate routes (through other peers or providers) with lower cost to every affected destinations (E, G, H) and propagates a cost change announcement within its hierarchy. Each of these steps converges in linear time. In case (b), when the peering link (A, B) fails (or incurs a cost change), the last two steps of case (a) (except the LSA propagation) are repeated, Hence, in both cases, the convergence time is linear.

Multiple peering links: To analyze the case when $k(D) > 1$, consider a simple topology consisting of m root-nodes (each node is a root of the AS hierarchy) which are connected *only* by peering links at a certain level in the AS hierarchy. The FPV routing part of HLP between the m nodes, is equivalent to the case where these nodes route between themselves using purely a path-vector protocol like BGP under the constraint that no path has length more than $k(D)$. Reusing Labovitz's path exploration argument, the maximum FPV routing convergence time of the system is given by $\prod_{j=0}^{j=k(D)-1}(m-j)$ which can be approximated by $O(m^{k(D)})$ when m is high and $k(D)$ is small. Once we have a bound on the convergence time of FPV, we can create an artificial AS-hierarchy where we replace the indirect peering link (through at most $k(D)$ hops) using a direct peering link. Now the resulting topology is singly peered and using the previous argument, the convergence time is linear (which is much smaller than FPV convergence time). Hence, the overall convergence of the system is bounded by $O(m^{k(D)})$. For a general topology with J levels in the AS hierarchy with h_j nodes at level j, where h_{max} represents the maximum value of h_j for all $1 \leq j \leq J$. The maximum convergence time for this topology is bounded by $O((h_{max})^{k(D)})$ which can be bounded by $O(n^{k(D)})$. □

On a final note, while this analysis used n to be the total number of ASes, the actual value of h_{max} is much smaller. Indirect peering typically occurs at the higher levels of the AS hierarchy. If we consider indirect peering only among tier-1 and tier-2 ISPs (which is typically the case), the value of h_{max} is only about 200 (in comparison to $n = 17,000$).

D Security issues in HLP

Routing security is of critical importance. We believe that HLP's architecture makes it simpler to secure than BGP, partly because a large volume of slowly changing data is separated out, which permits offline signature checking, and partly because the hierarchical nature of the protocol reduces the degrees of freedom of an attacker. We have investigated two security frameworks, one involving a public key hierarchy, and one which is more decentralized – combinations of the two are also possible.

D.1 Threat Model

Threats to a routing protocol come both from misconfiguration and from deliberate attack, where the principle concern is compromised routers. Routers are surprisingly vulnerable; some have *default passwords* [31, 30] and others use insecure interfaces such as telnet. Attacks can take the form of an *isolated adversary* (i.e., a single compromised router) or *colluding adversaries* (i.e., a set of compromised routers). The latter is particularly hard to defend against. For these purposes, misconfiguration can be considered as a special case of an isolated adversary.

The principle vectors for attacking HLP are the propagation of invalid route announcements, and the spoofing of explicitly published information. Invalid route announcements fall into three classes:

- Propagating bogus LSAs within the hierarchy
- Propagating bogus FPVs between hierarchies
- Lowering the cost metric in a forwarded advertisement to entice traffic by propagating smaller costs.

There are two types of information which is explicitly published:

- The mapping between address prefixes and ASes.
- The list of (provider,customer) relationships.

Propagating a bogus (AS, prefix) mapping is an *address space hijack*. Propagating incorrect provider-customer relationships affects the correctness of HLP itself.

D.2 PKI-based security

The principal objection to using public-key cryptography for routing is the deployment of a PKI. Secure-BGP [17] suffers from both problems, and has been deployed only in very small pockets after nearly 5 years. The public-key hierarchy

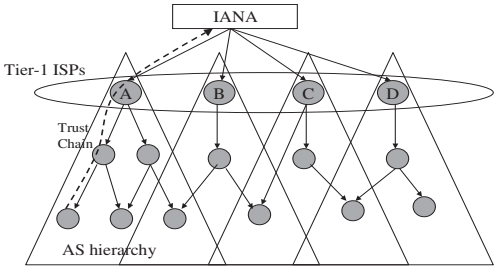

Figure 13: An Example PKI hierarchy rooted at IANA with the Tier-1 ISPs acting as the primary roots from the operational perspective.

needed by HLP follows the pre-existing provider-customer business relationships (as shown in Figure 13), greatly easing deployment. Previous work [28] on the Internet AS hierarchy has shown that the number of tier-1 ASes is small (roughly 20-25) and has been relatively stable over long timescales. Thus we envisage each tier-1 as a root of a public-key hierarchy covering its customers, with a virtual super-root at IANA. Given their small number, key management between the tier-1 providers should feasible.

An additional concern is the cost of public-key cryptography. In HLP, much of the data is in the form of relatively static mappings, which are amenable to offline validation and caching (thereby reducing the necessity of online validation). HLP can also leverage recent optimizations using Merkel hash trees [14] to minimize the cost of public-key operations.

Using a PKI, solves most of the security threats against HLP. First, digital signatures of the AS-to-address-prefix mapping and the provider-customer mapping ensure that these cannot be tampered with. Verifying the correctness of the LS and FPV announcements is straightforward since the initiator of an LSA and ASes along the FPV path append their digital signatures to the advertisement, which can be easily verified. Two issues are worth mentioning. First, HLP makes it possible to cache digital signatures to avoid the need to individually sign or check every announcement. However this might permit replay attacks. To prevent this, an AS may include a *nonce or time-stamp* in signed route updates. Second, the use of a PKI cannot prevent two colluding adversaries from tunneling advertisements to fake a link between them. Here again HLP's hierarchy limits exposure – faking a peering link between two stub networks won't affect anyone else. Only nodes high in the hierarchy can significantly affect traffic between a large number of networks.

D.3 Decentralized Security Approach

Our decentralized approach focuses on *triggering alarms* rather than preventing the origination of bad data. In this approach a router cannot verify a single route advertisement in isolation but can compare *two routes to the same destination* for consistency based on the Whisper protocol constructions [29].

The security guarantees of the decentralized approach are weaker than the PKI-based approach but it is more incrementally deployable. The Whisper signatures provides two benefits in the presence of misconfigurations and independent adversaries: (a) Any misconfigured or malicious router propagating an invalid route (either modifying the LSA or FPV or lowering the cost) will always a trigger an alarm; (b) A single malicious router advertising more than a few invalid routes will be detected and the effects of these spurious routes will be contained. To ensure the consistency of the mapping information, we propagate the (AS,prefix) and (provider,customer) mappings in a path-vector fashion (much like BGP) along with Whisper signatures. This ensures that anyone attempting to publish incorrect mapping entries or hijacking an address space will always trigger an alarm. Also, the effects of any single entity propagating several inconsistencies will be isolated and contained. Finally, the protection that decentralized security mechanisms can offer against colluding adversaries can be limited - the fundamental problem being only the ability to trigger an alarm as opposed to identify the attacker.