Design of Inter-Administrative Domain Routing Protocols

Lee Breslau **Deborah Estrin**

Computer Science Department
University of Southern California
Los Angeles, California 90089-0782
breslau@usc.edu estrin@usc.edu

Abstract

Policy Routing (PR) is a new area of devleopment that attempts to incorporate policy related constraints on inter-Administrative Domain (AD) communication into the route computation and forwarding of inter-AD packets.

Proposals for inter-AD routing mechansims are discussed in the context of a design space defined by three design parameters: location of routing decision (i.e., source or hop-by-hop), algorithm used (i.e., link state or distance vector), and expression of policy in topology or in link status. We conclude that an architecture based upon source routing, a link state algorithm, and policy information in the link state advertisements, is best able to address the long-term policy requirements of inter-AD routing. However, such an architecture raises several new and challenging research issues related to scaling.

1 Introduction

Internetwork size has grown rapidly as a result of several factors: proliferation of the number of networked hosts, interconnection of technically heterogeneous local area and wide area networks, and finally, interconnection of autonomous Administrative Domains (ADs). An AD is a set of resources–hosts, networks, and gateways–that is governed by a single administrative authority. Interconnection across ADs comes about through interconnection of private networks, interconnection of commercial carriers in a competitive market, and division of an internet that has grown too large to manage.

Common approaches to network interconnection create a fully connected internet out of the constituent networks. In the case of AD interconnection this is undesireable for two reasons: scale and policy. As with other types of systems, manageability is a problem for very large internets. Naming, routing, fault isolation, and security are all functions that are more easily and efficiently realized in the context of multiple, smaller, semi-autonomous regions, than in the context of a single, large, undifferentiated region. This is particularly true when the regions represent areas in which significant locality exists, e.g., ADs.

By definition, an AD represents a region that is governed by a single authority.[13, 15] Consequently, when ADs interconnect, issues of policy arise at the boundary between neighboring administrative authorities, and transitively across all the administrative authorities in the collective internet. Network access control mechanisms have been designed for use in inter-AD gateways to control access to end-systems.[8, 20] When networks are used for transit purposes, as well as for access to end systems, network access control mechanisms are not adequate. Policy Routing (PR) is a new area of development that attempts to incorporate policy related constraints on inter-AD communication into the route computation and forwarding of inter-AD packets. Several architectures have been proposed to implement policy based, inter-AD, routing. [2, 4, 10, 16, 18, 19]

In this paper, we present a model of internets for which inter-AD routing protocols must be developed. These protocols must function in the presence of a large number of ADs, and they must make routing decisions in accordance with administrative policy. Design issues relevant to these routing protocols are described, and current proposals for inter-AD routing are discussed within the context of three design issues: location of routing decision (i.e., source or hop-by-hop), algorithm used (i.e., link state or distance vector), and expression of policy in topology or in link status. We conclude that an architecture based upon source routing, a link state algorithm, and policy information in the link state ad-

This research was funded in part by the National Science Foundation, Presidential Young Investigator Award, with matching funds from AT&T

vertisements, is best able to support the policy requirements of inter-AD routing.

The remainder of the paper is organized as follows. Section 2 outlines the driving design requirements (i.e., model) for a PR architecture. Limitations of existing routing protocols are reviewed in Section 3. Design issues in inter-AD routing are presented in Section 4, and current proposals are discussed within the context of these design issues in Section 5. Section 6 concludes with a discussion of open research issues. An extended version of this paper can be found in [3].

2 Inter-AD Routing Model

Assumptions about inter-AD topology, scale and policies greatly influence the design of PR mechanisms. In this section we describe our model for the inter-AD environment.

2.1 Inter-AD Topology

The Research Internet has grown in a decentralized, evolutionary fashion. Many organizations connect to the Internet through bilateral arrangements with other organizations that already have Internet connectivity. The resulting topology is a mesh with varying degrees of connectivity at different places in the network. It now appears that the increasing availability of commercial high speed data services will lead to simpler and more hierarchical internet topologies. This hierarchical topology will consist of long haul backbone, regional, metropolitan, and campus networks. However, lateral links and other forms of bypass will persist at all levels of the hierarchy. Reasons for the persistence of these links include special technical requirement, economic incentives, and political/control incentives. For further justification see [3, 9].

The resulting topology is a hierarchy augmented with special purpose lateral links between some stub networks and between transit networks, as well as special purpose bypass links between stub networks and wide area backbone networks. Figure 1 shows an example internet with this kind of inter-AD connectivity. In this context *stub* network refers to an AD that is not used for transit by anyone outside of the AD. *Multi-homed* ADs are stub ADs that have more than one inter-AD connection but that wish to disallow any transit traffic. *Transit* network refers to an AD whose primary function is to provide transit services for many other ADs. Long haul backbone and regional networks are examples of transit networks. *Hybrid* (or limited-transit) networks are ADs that support access to end systems, as well as limited forms of transit to other ADs.

In the context of a global internet we require mechanism that allow stub, transit *and* hybrid ADs to exert control over the use of their resources. Further, the assumptions about the inter-AD topology emphasize the need for algorithms that generate loop-free routes. Inter-AD routing protocols should work efficiently for the general hierarchical case, but they must accommodate lateral and bypass links in a graceful manner. It is acceptable for there to be some performance impact, but functionally, the intergrity of the routing must be maintained in the presence on non-hierarchical structures. In return for the added complexity implied by this model, we will make some compensating assumptions about inter-AD dynamics.

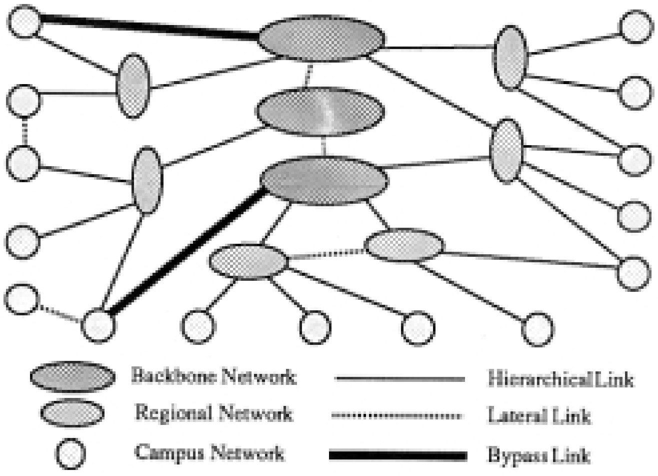

Figure 1: Example Internet Topology

2.2 Scale

We are interested in a general architecture to support the future world-wide internet of millions of networks spanning hundreds of thousands of ADs. Moreover, we are interested in the world of commercial carriers and other forms of private networks, as well as the Research Internet. Thus, the protocols should address the needs of a wide range of users.

For the sake of this evaluation we will assume that the global internet could grow to be on the order of 10^5 ADs. Many would be stub ADs but we would like an architecture that could work well for 10^4 transit ADs.[17]

In the context of this very large internetwork, it is desireable for inter-AD topology to change infrequently. Inter-AD topology changes when either an AD partitions, or the connection between two neighbor ADs fails. Since ADs are relatively large entities, it seems reasonable to make the assumption that intra-AD partitions will occur infrequently, as an AD is likely to be characterized by sufficient intra-AD network redundancy and a robust IGP. In other words, an AD must be configured to maintain relatively stable connectivity to the outside world if it is to get adequate service from the routing architecture. It is less practical to make such an assumption about inter-AD link redundancy. Consequently, the protocol must be somewhat adaptive to changes in inter-AD topology, since it is not desireable to rely on static routes.

2.3 Inter-AD Policies

We now consider the types of policies that ADs should be able to express and enforce and the implications for routing. The purpose of policy routing is to control use of network resources, not to act as end system access controls. Moreover, the security (i.e., assurance) of the control mechanisms is an orthogonal issue to the semantics of what kinds of controls can be expressed and enforced. Of course the level of assurance provided by the mechanisms will affect greatly the kind of policies that ADs express. For the sake of this paper we focus on the semantics of the kinds of policies we wish to express. Issues of security are addressed in [7].

As Clark points out in [4], the source of a packet as well as the carrier(s) of the packet should have the ability to express policy regarding its handling. We refer to policies of the carrier as transit policies and policies of the source as route selection criteria. Common source and transit policies may be based on such things as the source and destination of the traffic, the other ADs in the path, Quality of Service (QOS), time of day, User Class Identifier, authentication and security requirements, and charging and accounting policies. For a more detailed discussion of these policy requirements, see [9].

Although the purpose of PR is to enlarge the range of policies that can be enforced in internets, not all conceivable policies must be supported in a PR architecture. In all of the PR proposals that we discuss it is critical to their operation and performance that the policies be *slow to change*. Moreover, specific policies adopted by participating ADs will affect the performance of the overall internet. ADs should adopt the least restrictive policies possible and should control access at the coarsest granularity possible to maximize connectivity and enhance performance.

The policies that must be supported imply that a transit AD might make different routing decisions depending on where the packet originated, where it is destined, the path that it has traversed, the QOS requested, and the user class of the originator. In particular, *source* ADs need to be able to express policies regarding packet handling, leading to different handling for different packet sources by transit ADs. Furthermore, *transit* ADs must be able to specify policies that depend on the identity of the source AD. As a consequence there is no single spanning tree that describes the best route to a destination for all sources in the internetwork. Determining "the best", and even the availability of a route, depends upon the source of the packet, as well as on other conditions. QOS routing is characterized by a similar but far more manageable issue, namely the existence of multiple spanning trees, one for each QOS. Because any particular QOS spanning tree applies equally to all sources the potential increase in overhead is not as radical as with PR. It is this aspect of policy routing that makes the problem more difficult, but at the same time an interesting area of research in routing protocols.

The remainder of the paper addresses routing protocol design for inter-Administrative Domain routing in more detail.

3 Limitations of Traditional Routing Protocols

Thus far we have described requirements for controlled flow of traffic across AD boundaries. Network access controls based on different kinds of gateway filters have been used to control the flow of traffic into and out of stub networks.[8, 20] These filters allow an AD to filter packets not meeting certain criteria. However, transit networks must advertise their filtering policies in order to prevent routing loops and dropped packets. It is not sufficient to discover a policy by having packets dropped until a higher level timeout occurs. Rather, policy restrictions must be incorporated into the route calculation and selection processes. In this section we describe how existing routing protocols lack the functionality required by inter-AD routing. We discuss both interior and exterior gateway protocols.

Interior gateway protocols (IGPs) are designed for use within a single AD. A new generation of these IGPs have been developed, among them are IGRP[14], OSPF[21], and DEC IS-IS[5]. Independent of whether they use a link state or distance vector algorithm[1], these protocols have been refined to provide adaptive, shortest path routing with relatively low overhead (in terms of computation and information exchange) within regions of limited size.

All three of these IGPs provide support for QOS routing. IGRP, a distance vector protocol, uses a vector of metrics describing topological delay, bandwidth, channel occupancy, and reliability. The formula with which these individual metrics are combined into composite metrics can be adjusted to yield metrics suitable for different Qualities of Service. In OSPF and IS-IS, two link state protocols, link state updates can contain multiple metrics corresponding to different Qualities of Service, and the basic route computation is repeated for each QOS. These mechanisms support only a limited number of Qualities of Service; they are not scalable either to a large number of QOS or to source specific policies. As more sophisticated transport and internet protocols are developed to support more demanding QOSs, more will be demanded of IGPs to support QOS routing.

Exterior gateway protocols have been used to insulate regions of the internet from one another and thereby

[1] See Section 4.3 for further discussion of link state and distance vector algorithms.

avoid the information and computational explosion that IGPs do not accommodate. Exterior gateway protocols support technical heterogeneity by allowing interconnection of regions that run different IGPs. One such protocol, EGP[24], was developed for the DARPA Internet to exchange reachability information across relatively autonomous collections of networks. EGP supports a very limited notion of policy. It allows ADs to define what portions of their connectivity database they will share, but it does not allow them to express QOS or finer grain restrictions on the use of those resources. EGP also allows an AD to manipulate the metrics assigned to different ADs as a means of favoring or disfavoring other transit ADs[2]. However, EGP does not allow the AD to explicitly advertise its policy information for incorporation into the routing decision of other ADs.

EGP allows ADs some autonomy in defining metrics, as described above. However, in order to maintain loop-free routing, EGP places a severe topology restriction on interconnected regions–there can be no cycles in the EGP graph. As noted in Section 2.1, this is an unreasonable restriction for a global internet. ADs require the flexibility to configure multiple inter-AD connections, and it is not feasible to monitor connectivity adequately to enforce the topology restrictions, even if they were acceptable.

Based on the inability of existing IGPs and EGPs to address the requirements of inter-AD routing, we consider the design of alternative architectures in the following sections.

4 General Design Issues for Inter-AD Routing Protocols

Four issues in particular affect the design of an inter-AD routing architecture. This discussion sets the stage for Section 5 where we step through the design space defined by these issues and evaluate three existing proposals for inter-AD routing.

4.1 Level of Abstraction

The first design issue effecting inter-AD routing is the level of abstraction at which inter-AD routing should be treated. Routing protocols operating inside a single AD exchange information about the status of individual gateways and networks. In inter-AD routing, however, it is advantageous to exchange information at the granularity of ADs, and to treat an inter-AD route as a sequence of ADs. This abstraction reduces the amount of information exchanged between ADs, as well as the frequency of these exchanges. Also, it allows ADs to hide internal details of their networks.

As with any abstraction or hierarchical routing, some optimality may be lost. Nonetheless the benefits of this abstraction far outweigh its costs. Therefore, throughout this paper, we consider an inter-AD route to be a sequence of ADs, and we ignore routing internal to administrative domains.

4.2 Policy in the Routing Architecture

Our model of the internet, presented in Section 2, requires that administrative domains be able to restrict or allow access to resources based on administrative policy. Therefore, policy must be reflected in the routing architecture.

One way to accomplish this is to embed policy in the internet topology. In this approach, relationships are defined between neighbor ADs so as to control the flow of routing information, and therefore data packets across inter-AD links. For example, a proposal discussed later (see Section 5.1.1) makes use of a partial ordering of nodes to constrain the flow of routing information. Effecting policy through such an ordering is problematic because it limits the combinations of policies expressible using a single partial ordering. Also, maintaining these inter-AD relationships may require the involvement of a central authority or excessive coordination among ADs. However, this method of expressing policy lends itself well to scaling, as it allows ADs to be grouped into a hierarchy without affecting the policies that are expressible.

A second approach to expressing policy in the routing architecture is to explicitly associate policy related information with routing exchanges between ADs. That is to say, link or path updates contain administrative constraints and service guarantees that apply to the resources they advertise. We refer to these constraints as *Policy Terms (PTs)*.[4]

4.3 Routing Algorithms

Routing algorithms used in computer networks can be classified as either distance vector or link state. For a general discussion of these algorithms see [6, 11, 12]. In Bellman-Ford distance vector algorithms, a node receives information about its neighbors' shortest path metrics to all destinations. The node calculates its shortest paths and distributes this information to its neighbors. Distance vector algorithms are relatively simple to implement, but they can converge slowly. In link state algorithms, each node floods the status of its adjacent links to all other nodes in the network. Each node computes its shortest paths to all destinations using this complete topological information. Link state algorithms are more complex to implement, but they do not exhibit the same convergence problems that distance vector algorithms do.

Traditional distance vector protocols hide informa-

[2] See S. Brim, *IP Routing Between U.S. Government Agency Backbones and Other Networks*, December, 1989, available from the author.

tion about paths, providing knowledge only about the first hop toward a destination. While this may be desirable in some environments, inter-AD policy routing is concerned with control of access to network resources based on administrative policy. Addressing such administrative policy may depend on more knowledge about an inter-AD path than is provided by distance vector algorithms. Link state algorithms, on the other hand, provide all nodes with global information. In an inter-AD environment, this can include information about policy constraints needed to make routing decisions consistent with administrative policy.

4.4 Location of Routing Decision

The final design issue we consider is the location of the routing decision. Source routing refers to a paradigm in which the source of traffic determines the route and includes this route in each packet. Under hop-by-hop routing, each routing entity makes an independent decision to determine the next hop towards the destination.

Under the hop-by-hop paradigm, all nodes must make consistent routing decisions based on consistent data in order to avoid routing loops. In an environment where routing decisions are made according to adminstrative policy, this implies that all ADs must be aware of all other ADs policies. Source routing, on the other hand, provides a simple mechanism for avoiding routing loops. Specifically, the source AD uses a loop-free route synthesis algorithm, and/or inspects a source route to guarantee that it contains no loops.

In hop-by-hop routing, a source AD is constrained by choices made at transit ADs. Specifically, when distance vector algorithms are used, a source AD can only choose from among those routes selected and advertised by its neighbors. Similarly, using link state algorithms, the source has no control over the routing decision made by transit ADs. Therefore, regardless of the algorithm used, valid routes that would be preferred by the source AD may not be available to it.

Applying hop-by-hop routing to inter-AD policy routing implies that the source AD must rely on other ADs to make routing decisions in accordance with its policies. In source routing, however, the source has control over the entire inter-AD path. Therefore, the dependence on other ADs to select paths consistent with the source's policies is reduced.

In an environment with source specific policies, hop-by-hop routing also places an increased burden on transit ADs. Each transit AD may have to compute many routes to a single destination, to be used by different packet sources. Source routing relieves transit ADs of this burden; since the source specifies the next-AD hop, independent route computations by transit ADs are not required for each packet source.

5 Routing Architecture Design Space and Proposed Mechanisms

In the previous section, we identified three areas in which alternative design decisions can be made when developing an inter-AD routing architecture. The various combinations of these decisions yield a design space with eight distinct points (see Table 1). In this section, we discuss the points in this design space, presenting actual proposals when they exist and identifying architectures that are impractical.

		Decision Point		
		Hop-By-Hop Routing	Source Routing	
Algorithm	DV	5.1 (ECMA)	5.5.2	Policy in Topology
	LS	5.5.1	5.5.1	
	DV	5.2 (IDRP)	5.5.2	Policy Terms
	LS	5.3	5.4 (ORWG)	

Table 1
Design Space for Inter-AD Routing

As listed in Table 1, we begin by discussing inter-AD routing architectures using a distance vector algorithm, hop-by-hop routing, and policy embedded in the topology. Successive design points are described by altering one aspect of the design at a time. That is, we then consider an architecture with distance vector and hop-by-hop routing, but with explicit policy terms used to express policy. Next, a link state algorithm is substituted for the distance vector algorithm, and the resulting architecture (link state, hop-by-hop, policy terms) is discussed. Finally, by using source routing instead of hop-by-hop routing, an architecture employing link state, source routing, and explicit policy terms is presented.

While the bulk of this section centers on the four design points mentioned in the previous paragraph, there are four other possible design points to be addressed. We conclude this section by briefly touching upon these remaining design points.

5.1 Distance Vector, Hop-by-Hop Routing, with Policy Embedded in the Topology

The first point in the matrix of design possibilities represents architectures that employ a distance vector algorithm, use hop-by-hop routing, and express policy through topology. After outlining the features of such an architecture, a proposed protocol corresponding to this design is presented.

In this architecture, an AD exchanges routing table entries with its neighbors. An AD selects the "best" path from among those offered, and can then distribute its own routing table entries to its neighbors. Policy is reflected in the topology of the internet. That is to say, rather than explicitly including policy related information in routing updates, policy is reflected in implicit characteristics of links between neighbor ADs. For instance, in the proposal discussed below, a partial ordering is imposed on the topology, defining a relationship between neighbors.

An AD may opt not to distribute its routing table entries describing routes to other ADs. In this way, the AD acts only as a stub AD and will not carry transit traffic. Alternatively, the AD can advertise routes to a subset of destinations only, serving as a transit AD for traffic destined to this subset while refusing to carry traffic bound for other ADs.

The forementioned is an example of destination specific policies. Expression of source specific policy is more problematic in this architecture. If a partial ordering is imposed on ADs, distribution of routing table updates can be limited to only those ADs above or below an AD in the partial ordering. For instance, in the proposals discussed below, if an update is passed to a node lower in the partial order, this information can never be passed to a higher node. In this way, an AD can specify a policy with respect to a subset of sources.

This inter-AD routing architecture has several deficiencies. First, as with all distance vector algorithms, looping and speed of convergence must be addressed carefully. Second, using a single partial ordering to implement policy routing limits the specific policies that can be expressed by each AD. Third, as the route computation is distributed, ADs are constrained by decisions made elsewhere. The particular route selection made by a downstream AD may not adhere to a source ADs policy requirements, resulting in no available route when in fact a legal route exists (i.e., a route that is permitted by the policies of all transit ADs involved).

Now we turn our attention to a specific proposal for inter-AD routing that corresponds to the architecture described above. Attempts to deal with some of the problems identified with the architecture are evaluated.

5.1.1 ECMA

The National Institute of Standards and Technology (NIST) proposal, as submitted to ECMA[10][3], specifies a method for routing database distribution and for route computation based on the distance vector data. This proposal is designed for use in a topology containing cycles. The traditional looping and convergence problems are avoided through the use of a partial ordering of all clusters[4] [5], or ADs. This partial ordering must be coordinated among ADs. Consequently, change in the partial ordering must be coordinated by an authority that manages the partial ordering for all ADs effected by the change. In addition, multiple routing databases can be used for different QOS.

The partial ordering of ADs prevents looping and convergence problems in the presence of an inter-AD topology containing cycles. Every inter-AD link is labelled as an up link or down link, depending upon the relationship between the neighboring ADs in the partial ordering. Data packets are marked as to the type of links they have traversed. Once a packet traverses a down link, it cannot traverse another up link, thereby preventing loops. Routes described in distance vector updates are marked as to the types of links traversed to reach the destination, so that forwarding decisions that prevent loops can be made.

Changes in topology result in rapid convergence since the partial ordering suppresses looping. A topology change affects ADs close to the source of the change, and the effect weakens for those ADs farther away. The altered AD tells about a new link to all neighbors and they either reject it or pick it up in one computation. If the partial ordering is computed properly, and verified, the partial ordering and up-down rule prevent loops, and consequently prevent the count to infinity phenomenon common to other DV algorithms.

ECMA also has a mechanism for supporting QOS routing. Each AD can define multiple sets of Forwarding Information Bases (FIB) corresponding to multiple QOS indexes. An AD defines a separate metric for each QOS supported by at least one of its neighbors. If a particular neighbor does not advertise a particular QOS then the AD assigns an infinite metric to the neighbor for that QOS, and consequently the AD does not compute routes for that QOS through the neighbor.

As with distance vector protocols in general, ECMA supports information hiding, as well as the selection of one next hop over another. Similarly, destination specific filters can be applied to the distribution of routing updates in order to control the destinations to which transit traffic is carried. Source specific policies, however, are possible only to the extent that they can be reflected in the partial ordering of ADs. For instance,

[3] A second protocol adhering to this architecture, Border Gateway Protocol (BGP), version 1, is not discussed here.[18]

[4] It has been proposed that the same physical group of AD resources may be replicated and represented as multiple logical clusters for the sake of reflecting policy in the topology, thus allowing a wider range of policies to coexist. However, logical replication requires that the replicated region be assigned multiple network addresses in order to determine which FIB (routing table) should be applied to a particular packet.

[5] A cluster is analogous to an AD. For consistency with the remainder of this paper, we use the term AD.

if an AD distributes a routing update over a down link, this information cannot be passed up the hierarchy by a subsequent AD. In this way the AD has some control over the eventual recipients of its routing updates, and hence the traffic sources for which it carries traffic.

We have two fundamental concerns with ECMA's suitability to inter-AD routing. The first is the limitations on policies expressible by the protocol. We have already mentioned problems associated with expressing source specific policies using the partial ordering. Also, ECMA is not well-suited to express finer grained policies based on such things as User Class Identifier. The QOS mechanism does not scale well with the number of possible packet classifications (e.g., UCI, QOS, source). Finally, policies of different ADs may not be mutually satisfiable. That is to say, there may not be a single partial ordering that simultaneously expresses the policies of all ADs.

The second, and related, concern regards scaling and the practicality of maintaining the global partial ordering. It is uncertain whether this scheme is workable for a large number of ADs that have varied, non-static policies. Establishing the global partial ordering requires both computation and negotiation either by a central authority or by a set of entities each with authority over a subset of the internetwork. First, the policies of all ADs must be collected. A computation is applied that attempts to accommodate all the policies in a single partial ordering. If unresolvable conflicts arise among policies, i.e., those that can not be accommodated in a single partial ordering, then the relevant authority must negotiate with the ADs involved to revise their policies in such a way that they can be accommodated in the single partial ordering. This scheme is intended to work for a near infinite number of ADs. However, when policy changes, the partial ordering may need to be recomputed and may require another round of negotiation with affected ADs. Therefore, whereas the scheme may be feasible for a very large internet with static policies it is not appropriate for an environment of variable policies as was described in Section 2.

In summary, the ECMA approach does incorporate policy routing within an architecture that uses hop-by-hop routing and DV route computation. However, the DV approach, by definition, implies that an AD advertise a single metric per-destination per-QOS to all of its neighbors. This metric is a function of the entire path from that AD to that destination. It allows the AD to hide information about its own path to the destination, and therefore it withholds information from its neighbors that possibly is relevant to the neighbors' policies. Moreover, the DV approach allows ECMA to support only transitive policy relationships. Policies that discriminate among traffic sources in a non-transitive manner are cumbersome, and sometimes impossible, to support in ECMA.

5.2 Distance Vector Hop-by-Hop Routing with Explicit Policy Terms

The discussion of the previous design revealed difficulties imposed on inter-AD policy routing by the use of topological restrictions to express policy. We next consider another design using distance vector, hop-by-hop routing. However, in this case, policy is expressed by explicitly including policy attributes in routing updates. Two proposed protocols that reflect this design, BGP version 2 and Inter-Domain Routing Protocol, are described.[1, 19]

In traditional distance vector protocols, routing exchanges include only a destination and a metric. In this section, we describe an architecture that includes additional information, related to the policy constraints of a path, thereby allowing more flexible expression of policy. For instance, a routing update may include a list of the source ADs that are permitted to use the route described in the routing update, and/or a list of all ADs traversed along the advertised route.

When an AD receives a routing update, the update specifies a destination and policy constraints associated with the route to that destination. The AD receiving the update must then determine whether it can use the route based upon these policy constraints. If so, the AD can apply its own policy filters to determine whether or not it wants to use the route. For instance, better (e.g. less constrained) routes to the same destination may already exist, so the new route may be rejected. If the route is accepted by the AD, it then determines whether to advertise this route to its neighbors, based on its own policies. If it chooses to advertise the route, additional policy constraints can be added to it before distributing the update to its neighbors.

Traditional protocols employing distance vector hop-by-hop routing only allow nodes to advertise a single route to each destination per-QOS in order to avoid looping. When routing decisions are based on administrative policy, it may be desirable to advertise multiple routes per destination, each with different policy attributes. In this case, a set of policy attributes can be treated much like a quality of service in QOS routing. Thus, it is possible to advertise multiple routes, and still avoid looping, so long as each route and each packet can be identified with a unique set of policy attributes.

While this protocol allows more general policies to be expressed by ADs, it still suffers from problems inherent in hop-by-hop routing. Transit ADs can use policy terms to compute and advertise routes with diverse policy requirements to their neighbors. However, if these policies are source specific, transit ADs might have to perform separate calculations for each possible source AD. This approach is analogous to maintaining multiple spanning trees. Moreover, in this hop-by-hop scheme, the source is dependent upon subsequent ADs to make routing decisions in accordance with the source's policy. Since source route-selection criteria are not advertised, there is no means for the source to assert its preference that particular routes be used or avoided. In summary, transit ADs may expend resources computing multiple routes per destination (many of which may never be used), and source ADs may be unable to use the routes they prefer.

5.2.1 Inter Domain Routing Protocol

Two protocols adhering to these design choices, Inter Domain Routing Protocol (IDRP) and BGP version 2, have been proposed. As the two protocols are very similar, we will focus the present discussion on IDRP, mentioning BGP only where it differs from IDRP.

IDRP attempts to solve the looping and convergence problems inherent in distance vector routing by including full AD path information in routing updates. Each routing update includes the set of ADs that must be traversed in order to reach the specified destination. In this way, routes that contain AD loops can be avoided.

IDRP updates also contain additional information relevant to policy constraints. For instance, these updates can specify what other ADs are allowed to receive the information described in the update. In this way, IDRP is able to express source specific policies.[6] The IDRP protocol also provides the structure for the addition of other types of policy related information in routing updates. For example, User Class Identifiers could also be included as policy attributes in routing updates.

Using the policy route attributes IDRP provides the framework for expressing more fine grained policy in routing decisions. However, because it uses hop-by-hop distance vector routing, it only allows a single route to each destination per-QOS to be advertised. As the policy attributes associated with routes become more fine grained, advertised routes will be applicable to fewer sources. This implies a need for multiple routes to be advertised for each destination in order to increase the probability that sources have acceptable routes available to them. This effectively replicates the routing table per forwarding entity for each QOS, UCI, source combination that might appear in a packet. Consequently, we claim that this approach does not scale well as policies become more fine grained, i.e., source or UCI specific policies.

5.3 Link State Hop-by-Hop Routing with Explicit Policy Terms

We now discuss another point in the design space by considering the use of a link state algorithm along with hop-by-hop routing and explicit policy terms in routing exchanges. Within the context of this discussion, nodes refer to ADs and links to inter-AD connections. In the design under consideration, link state updates can be augmented to include policy related attributes of the resources they advertise, such as restrictions placed on, or service guarantees provided by, their use. Such an approach was suggested in [23].

These link state updates will be flooded throughout the internet, giving each AD global knowledge of all links and their associated policy restrictions. This information permits each AD to compute routes satisfying any set of policy restrictions to all other ADs. Therefore, this architecture allows an AD to discover a valid route if one in fact exists. However, each AD along this route must repeat the same calculation to compute this route. In link state algorithms without policy routing, a node computes a single spanning tree for all possible destinations. This spanning tree is used to route packets regardless of their source. However, because we allow for the possibility of source specific policies, an AD potentially must compute a separate spanning tree for each potential source of traffic. Hence, the replicated nature of this computation may become an excessive burden for transit ADs. If each node does not compute and maintain multiple spanning trees, then limitations such as those described in the previous section exist.

Also, we note that as in the architecture outlined in the previous section, sources are dependent upon other ADs to make routing decisions that conform to their policy requirements. Even though the source has calculated an entire route that adheres to its policy, it still relies on other ADs to repeat and replicate this same computation. Further, in order to avoid loops, all ADs in the path must make the same decision as the source. This implies that all ADs in the path must be aware of policy related criteria used by the source to select from among multiple available routes. This problem, as well as the problem of computing multiple spanning trees, is addressed by the next design.

5.4 Link State Source Routing with Explicit Policy Terms

The final design that we consider in detail employs a link state algorithm with source routing and explicit policy terms. We begin with a general discussion of this design choice, and then present a specific proposal developed by D. Clark and the Internet Open Routing Working Group.[4, 16]

As with the previous design discussed, link state updates containing policy related information are flooded throughout the internet. Using complete knowledge concerning topology and policy, each node is able to discover routes (if they exist) to any destination with any combination of policy attributes.

[6] The BGP protocol, as specified in [19] does not allow for the expression of such source specific policies, but we note that it would not be difficult to add this to the protocol.

However, in this architecture source routing is employed. That is, after the source calculates a route, it includes the entire route in the packet header so that subsequent ADs in the path need only examine the header to determine the next AD in the path; subsequent hops do not make an explicit routing decision. As stated previously, we consider an inter-AD route at the abstraction of a sequence of ADs. Thus the route calculated by the source, and included in the packet header, consists of a sequence of ADs. Intra-AD routes are a matter left to local concern. This approach attempts to balance the benefits of source control with adaptive routing capabilities of hop-by-hop routing.

This design affords important advantages. First, it grants the source control over the entire route. Therefore, the source can express and enforce any combination of its own policies, and it can keep these policies private from other ADs. Moreover, this control is achieved without requiring transit ADs to compute routes that adhere to the policies of all possible source ADs. A transit AD can concentrate on assuring that routes crossing it conform to its own policies, while leaving other ADs to enforce their own policies. Finally, source routing provides an efficient mechanism for assuring loop-free routes, independent of the network topology. Therefore, multiple paths to a single destination are feasible, without replicating entire routing tables.

We now turn to a review of a specific proposal that illustrates this design. In particular, we describe how policy is expressed, and how the overhead associated with source routing (e.g., increased header length) is minimized.

5.4.1 ORWG Architecture

The Internet Open Routing Working Group (ORWG) is developing a detailed architecture based on D. Clark's model for inter-AD routing, described in [4, 16].[7] ORWG represents a substantial departuare from current routing protocols and is in the early stage of prototype development.

Routes are determined by the source at the level of abstraction of ADs. The path must traverse the ordered list of ADs but the physical nodes and links traversed between and across ADs may vary.

ADs advertise Policy Terms (PTs) that can express the types of policies described in Section 2.3. Specifically, PTs can associate path constraints, QOS, User Class, authentication requirements, and other global conditions with a path across an AD. Path constraints restrict access to the path based on source AD, destination AD, previous AD, or next AD in the path. An AD can use a Policy Term to traverse another AD only if it meets the conditions specified in the Policy Term. ADs also advertise their connectivity to other ADs.[8] A Route Server in each AD computes Policy Routes based on the advertised policy and topology information. Packets to a particular destination travel via the route specified in the Policy Route.

If a packet is traveling to a destination for which there is no currently valid policy route in use then the first packet sent must carry enough information in it to allow each AD on the path to validate the path as legal. The AD's border gateways, referred to as policy gateways (PGs), execute the validation for the AD. In effect, one can view the PGs as containing routing tables that are filled on demand. The combination of possible routes is so large that it is not practical to hold the entire set of possible routing choices. At the same time, the overhead of carrying and processing complete information for each packet is prohibitive. Thus, the first packet that travels to a destination under a certain set of conditions acts as a policy route setup packet. This packet carries the full policy route (list of ADs) and a Policy Term from each AD that the source AD believes will allow it to use this route. A policy gateway for each AD along the route checks the information and validates that the policy route is in accordance with the local policy terms of that AD. If it is, the setup information is cached and the setup packet is forwarded.

To avoid the latency of the Policy Route setup process and the header-length overhead of the source route in the Policy Route packet header, data packets that travel down an already-established policy route do not carry the same information as the Policy Route setup packet. Instead, a handle is assigned at the time that the Policy Route is set up and successive data packets use that handle. PGs use the handle ID as a key into the cache to allow for some per-packet validation (e.g., is it coming from the AD specified in the cached PT setup information). It is essential for the operation of this protocol that policy and topology change much more slowly than the time required for route setup.

This setup process has some similarities with a traditional virtual circuit model. However, there are no assumptions made about guaranteed and sequenced delivery. Packets may be delivered out of order by taking different routes within an AD. Sequencing and reliability are left to the transport layer to do as required by the application. Moreover, PRs may have a long lifetime and are not intended to correspond one to one with transport level sessions. Thus, a single policy route can support multiple pairs of hosts in the source and destination ADs.

This scheme allows for very general policies to be

[7] We will refer to the architecture as ORWG because where the two models differ we describe the ORWG variation. However, many of the general concepts were first described by Clark. We also leave out many of the protocol details described in [4, 16, 22].

[8] ORWG refers to the point of connection between ADs as virtual gateways. A virtual gateway may be comprised of multiple PGs in the interest of reliability and performance.

expressed by source and transit ADs. Sources are given control over route selection, and transit ADs can express a wide range of policies in the policy terms they advertise. However, route computation presents a significant concern.

Route computation complexity is a function of internet size, dynamics, and the granularity of the policies expressed by transit regions. Given that route computation is a computationally intensive task, it is not practical to recompute routes frequently. If policy terms are highly dynamic, PRs will frequently be out of date. Therefore, PTs should change slowly.

For similar reasons, policies should not be very granular. Although the protocol allows for host specific policies, the implication of such policies is many more PTs and an increase in the route synthesis overhead. The ORWG architecture is intended primarily for network resource control. It is not a replacement for end-system and network access controls for sensitive environments.[7]

Even with coarse grained policies that change slowly, route synthesis for the ORWG architecture presents a challenge. Precomputation of all policy routes in a large internet is computationally intractable, while on demand computation may introduce excessive latency at setup time. Consequently, a combination of precomputation and on-demand computation should be used. For example, precomputation could use heuristics to prune the search and limit it to commonly used routes. On-demand computation could then be used in those cases where a requested route was not discovered during the precomputation phase. Adapting route synthesis to an internet of global scale is the subject of ongoing research.

5.5 Other Designs

The matrix of design possibilities that we presented contained eight elements. Thus far, we have reviewed four of these designs, for which proposals already exist or for which reasonable proposals could be developed. In this section we address the four remaining design possibilities, indicating why we have excluded them from more detailed coverage.

5.5.1 Link State and Policy in the Topology

Two of the designs neglected thus far include those using link state algorithms and topology to express policy. Link state algorithms depend upon flooding of link status to all nodes in a network. Policy routing based on topology, on the other hand, uses relationships among nodes to constrain the flow of routing information. For this reason, we see these two design choices as presenting no particular advantages over those schemes already described.

5.5.2 Distance Vector and Source Routing

The two remaining designs are those that include both distance vector algorithms and source routing. We do not view these choices as mutually incompatible. One could imagine, for instance, a protocol like BGP in which the source uses the full AD path information it receives in routing updates to create a source route. Such a protocol could address some of the deficiencies identified with distance vector, hop-by-hop designs. However, we opt against further discussion of such a protocol because there is little advantage in using source routing without also using a link state scheme. The power of source routing, in the context of inter-AD policy routing, is in giving the source control over the entire route. This goal cannot be realized fully without giving the source complete information for, and control of, the route computation itself–such as a link state algorithm provides.

6 Conclusion

We presented a model of internets for which inter-AD routing protocols must be developed. These protocols will be required to function in the presence of a large number of ADs, and they must make routing decisions that adhere to administrative policy. Existing protocols have either been designed for use inside a single administrative domain or they do not support a wide range of policies. Three current proposals for inter-AD routing mechanisms were discussed in the context of an eight element design space defined by routing algorithm, routing decision location, and policy definition. We concluded that an architecture including source routing and a link state algorithm with policy terms is best able to solve the long-term requirements of inter-AD routing.

In the context of such an inter-AD routing architecture there remain many unanswered research questions. We conclude with a brief discussion of several outstanding issues:

- Policy Route computation is probably the most difficult aspect of this approach. Heuristics for pruning precomputations and for focusing on-demand computations must be developed. Simulation of route synthesis for realistic internets should be conducted to explore tradeoffs in synthesis strategies and effects of internet topology and policies.

- Within this routing architecture, it will be the job of local administrators to specify policies for their ADs. Given the interaction between local policies and the policies of other ADs, it will be possible to specify local policies that will result in poor service, both in terms of route computation overhead and the resulting inter-AD connectivity. Thus, it will be imperative for these administrators to have available network management tools to assist them in predicting the impact of their policies on the service received from the routing architecture.

- Several issues related to scaling demand further exploration. Two examples are database distribution strategies to provide the needed information for route computation while minimizing routing-data distribution overhead, and policy gateway state management and limitations.

7 Acknowledgements

Members of the Open Routing Working Group have contibuted to the ideas presented here. Yakov Rekhter provided many detailed comments and corrections to this and previous versions. Tassos Nakassis, Yakov Rekhter and Paul Tsuchiya provided many clarifications and explanations of the ECMA, BGP and IDRP proposals. Also, we appreciate Abhijit Khale, Martha Steenstrup, Gene Tsudik, Lixia Zhang, and the anonymous reviewers for their comments on previous drafts.

References

[1] ANSI, *Intermediate System to Intermediate System Inter-domain Routeing Information Exchange Protocol*, **Document Number X3S3.3/90-132**, June 1990.

[2] H. Braun, *Models of Policy Routing*, **RFC 1104, SRI Network Information Center**, June 1989.

[3] L. Breslau and D. Estrin, *Design of Inter-Administrative Domain Routing Protocols*, **University of Southern California TR 90 08**, March 1990.

[4] D. Clark, *Policy Routing in Internet Protocols*, **RFC 1102, SRI Network Information Center**, May 1989.

[5] Digital Equipment Corporation, *Intermediate System to Intermediate System Intra-Domain Routeing Exchange Protocol for Use in Conjunction with the Protocol for Providing the Connectionless-Mode Network Service*, October 1989.

[6] E. W. Dijkstra, *A Note on Two Problems in Connection with Graphs*, **Numerische Mathematik**, 1959, pp 269-271.

[7] D. Estrin and G. Tsudik, *Security Issues in Policy Routing*, **Proceedings of 1989 IEEE Symposium on Security and Privacy**, May 1989.

[8] D. Estrin, J. Mogul, G. Tsudik, *Visa Protocols for Controlling Inter-Organizational Datagram Flow*, **IEEE Journal on Selected Areas in Communications**, May 1989.

[9] D. Estrin, *Policy Requirements for Inter Administrative Domain Routing*, **RFC 1125, SRI Network Information Center**, November 1989.

[10] European Computer Manufacturers Association, *Inter-Domain Intermediate Systems Routing*, **Technical Report ECMA/TC32-TG10/89/56**, May 1989.

[11] L. R. Ford and D. R. Fulkerson. *Flows in Networks*, Princeton University Press 1962.

[12] J. J. Garcia-Luna-Aceves, *A Unified Approach to Loop-Free Routing Using Distance Vectors or Link States*, **ACM Sigcomm**, 1989.

[13] S. Hares, D. Katz, *Administrative Domains and Routing Domains, a Model for Routing in the Internet*, **RFC 1136, SRI Network Information Center**, December 1989.

[14] C. L. Hedrick *An Introduction to IGRP*, **Technical Report, The State University of New Jersey Center for Computers and Information Services**, October 1989.

[15] ISO, *OSI Routeing Framework*, **ISO/TF 9575**, 1989.

[16] M. Lepp and M. Steenstrup. *An Architecture for Inter-Domain Policy Routing* **DRAFT RFC**, January, 1990.

[17] M. Little, *Goals and Functional Requirements for Inter-Autonomous System Routing*, **RFC 1126, SRI Network Information Center**, October 1989.

[18] K. Lougheed and Y. Rekhter, *Border Gateway Protocol*, **RFC 1105, SRI Network Information Center**, June 1989.

[19] K. Lougheed and Y. Rekhter, *Border Gateway Protocol*, **RFC 1163, SRI Network Information Center**, June 1990.

[20] J. Mogul, *Simple and Flexible Datagram Access Controls for Unix-based Gateways*, **Proceedings of Summer 1989 USENIX Technical Conference**, August 1989.

[21] J. Moy, *The* Open Shortest Path First *(OSPF) Specification*, **RFC 1131, SRI Network Information Center**, October 1989.

[22] Open Routing Working Group, *Inter-Domain Policy Routing Protocol Specification and Usage: Version 1*, **DRAFT RFC**, April 1990.

[23] R. Perlman, *Incorporation of Service Classes into a Network Architecture*, **Proceedings of the Seventh Data Communications Symposium**, October 1981.

[24] E. Rosen, *Exterior Gateway Protocol (EGP)*, **RFC 827, SRI Network Information Center**, October 1982.