A Directory Service for Perspective Access Networks

Geoffrey Goodell Mema Roussopoulos Scott Bradner

Abstract

Network fragmentation occurs when the accessibility of a network-based resource to an observer is a function of how the observer is connected to the network. In the context of the Internet, network fragmentation is well-known and occurs in many situations, including an increasing preponderance of network address translation, firewalls, and virtual private networks. Recently, however, new threats to Internet consistency have received media attention. Alternative namespaces have emerged as the result of formal objections to the process by which Internet names and addresses are provisioned. In addition, various governments and service providers around the world have deployed network technology that (accidentally or intentionally) restricts access to certain Internet content. Combined with the aforementioned sources of fragmentation, these new concerns provide ample motivation for a network that allows users the ability to specify not only the network location of Internet resources they want to view but also the *perspectives* from which they want to view them. Our vision of a *Perspective Access Network* is a peer-to-peer overlay network that incorporates routing and directory services that allow network perspective-sharing and non-hierarchical organization of the Internet. In this paper, we present the design, implementation, and evaluation of a directory service for such networks. We demonstrate its feasibility and efficacy using measurements from a test deployment on PlanetLab.

I. Introduction

Network fragmentation occurs when the availability of a resource to an observer is a function of how the observer is connected to the network. In the context of the Internet, network fragmentation is well-known and occurs in many situations, including an increasing preponderance of network address translation, firewalls, and virtual private networks.

Recently, however, new threats to Internet consistency have received media attention. The issues fall into two categories: conflict concerning *naming* and the use of *geolocation* to restrict access to resources. First, a number of nations have raised formal objections to the oversight of ICANN by the United States, and a number of private organizations such as UnifiedRoot have emerged to offer alternative namespaces [29]. Global agreement on Internet governance is becoming increasingly difficult [37] which means the potential for inconsistency in naming resulting from multiple DNS roots or addresses that are not globally unique will only increase. To a significant extent, the Internet depends upon everyone having access to the same set of names. The threats, therefore, are as follows: (a) the same name does not exist in both of two locations (lack of global consistency), and (b) the same name refers to different resources in different locations (lack of global uniqueness).

Second, a perceived increase in online criminal activity has created viable business models for businesses that provide geolocation services marketed for their benefits in fraud resolution and digital rights management.[1] For example, a number of companies use these geolocation services to obtain information about how a user is connected to the Internet (such as IP address and ISP data) to determine whether the user is likely to be fraudulent. This has caused a number of legitimate online transactions to be denied when users are not connected at their usual point of attachment [20]. Finally, various governments and service providers around the world have deployed network technology that (accidentally or intentionally) restricts access to certain Internet content [26], [13].

Combined with the various well-known sources of fragmentation these new concerns provide ample motivation for the development of a technique that affords users the ability to specify not only the network location of Internet resources they want to view but also the *perspectives* from which they want to view them. In this paper, we present the design, implementation, and evaluation of a *Perspective Access Network* (PAN), an overlay infrastructure for sharing perspectives. Our PAN prototype, called *Blossom*, consists of an unstructured, peer-to-peer overlay of *forwarders* carrying TCP traffic that act as intermediaries between nodes that cannot communicate directly.

Previous work on overcoming network fragmentation to facilitate end-to-end connectivity requires extensive changes to operating systems (such as deployment of new protocol stacks), requires the explicit participation of ISPs and content providers, or imposes a global hierarchical organization of the Internet. We relax these constraints to provide *ease of deployment* and have built a system we have deployed on the Tor anonymity network [10] and on PlanetLab. Our approach does not require changes to the operating system or protocol stack, does not require the active participation of ISPs and does not require special configuration of in-band network-layer elements such as routers or middleboxes.

PAN also does not impose global hierarchical organization of the Internet [17]. Currently, both the addresses and the names used to identify resources on the Internet are allocated

[1] CyberSource, `http://www.cybersource.com/`; NatGeo `http://www.natgeo.com/`; Quova, `http://www.quova.com/`

by a collection of governance organizations that are arranged hierarchically with a single organization at the top having overall "control." Our approach allows for an Internet without hierarchically ordained names and address spaces—that is, an Internet consisting of (possibly overlapping) network fragments, each with its own local naming and addressing scheme. This scheme promotes *locality in naming*, in that multiple resources with the same name can co-exist in different local namespaces (fragments). This scheme also promotes *distributed management* of local networks, in that adding a new local network and its abundance of resources to the Internet need not require specific allocation of names, addresses, or routing from centralized authorities.

For our overlay, we assume that each forwarder need only have the ability to communicate bidirectionally with some subset of the other forwarders. A PAN client that wishes to view a resource from the perspective of a particular forwarder F that satisfies certain criteria uses the PAN distributed directory service (provided, in our prototype, by a subset of forwarders) to determine a path of connectivity through the forwarders to F. The PAN client then constructs a source-routed circuit through the forwarders on the path to F, which then performs a DNS lookup to resolve the local resource name to an IP address from its point of view and access the resource on behalf of the client. The client is connected to the resource through F and thus accesses the resource from the perspective of F.

Since we do not impose a global unique naming scheme for resources, we need a way to uniquely identify a resource. We therefore require forwarders to generate unique, self-certifying identifiers and a PAN client specifies a particular resource by concatenating the forwarder ID with the local resource name as resolved by the forwarder. This design choice, however, sacrifices a certain amount of aggregation we can perform when advertising forwarder route information within the PAN overlay.

II. Related Work

A number of existing projects that focus on overcoming Internet fragmentation propose their own directory management schemes. These projects include:

INDIRECTION. I3 [33] provides a "rendezvous-based communication abstraction" in which providers of services register with a particular location in the network, and those peers requesting services communicate with that location rather than with the provider directly. TRIAD [5] uses globally unique, hierarchical names to identify networks; these names are propagated throughout the system via BGP-like advertisements among TRIAD nodes. PAN does not require registration of services and the local names of resources need not be globally unique. While PAN forwarders do have globally unique identifiers, the structure and allocation of these (self-certifying) identifiers are not hierarchically ordained.

ANTI-CENSORSHIP. Psiphon is a single proxy application used to circumvent content filtering. A host within a country without filtering installs the Psiphon proxy software and remote hosts in countries with filtering can access blocked web sites through the proxy. Infranet [11] and Tor [10] use overlay networks to provide anonymous communication. Anonymity networks such as these can also be used for anti-censorship purposes, specifically to circumvent local restrictions on access to resources. However, since the Internet is not entirely flat, the resources to which a user of these networks (or of Psiphon) has access may vary as a function of the particular overlay node (or Psiphon host) that is used as the last-hop proxy. For example, requesting a particular web page from an anonymity network might yield content that has been tailored to the particular local network or geographic region in which the last-hop proxy resides. If anonymity is the goal, then a larger anonymity set may be worth the cost of some probabilistic variation in content reachability. PAN takes the opposite approach, choosing to use an overlay proxy network to maximize content reachability, possibly at the expense of anonymity.

DECOUPLING POLICY FROM MECHANISM. FARA [6], [7] provides a general framework for describing associations between nodes without requiring a global namespace. Platypus [31] provides a system for enforcing routing policy on the forwarding plane rather than the control plane, relying upon cooperation from intermediary ISPs. PAN aims not to require such cooperation, at least not on a technical level. However, PAN does present an argument for separating network access policy from technical decisions made at the network layer. If two PAN forwarders are both connected to the same PAN overlay, then technically speaking, each could have access to whatever the other can see, regardless of what lies between.

NON-UNIVERSAL NAMESPACES. Semantic-Free Referencing [36] stipulates that resources have globally-unique "semantic-free tags", high-entropy bit strings perhaps generated as self-certifying names by the resource provider. A client would use the semantic-free tag rather than a hostname to identify the website, and a Reference Resolution Service (RRS) would map human-readable names to semantic-free tags. The goal is to decouple the name of a resource from its content; note that this is subtly different from the *naming locality* goal of PAN.

EMBRACING HETEROGENEITY. Plutarch [9] provides access across the boundaries of fundamentally different networks. Like PAN, Plutarch does not require a well-defined Internet core or global names. Plutarch "contexts" are similar to the "fragments" that we describe. However, unlike PAN, Plutarch requires these contexts to be well-defined and non-overlapping. Plutarch also resolves names via a peer-to-peer search, which PAN avoids in favor of reducing overhead and improving connection setup time.

DNS. The Domain Name Service [21], [22] is the widely used directory service for resolution of hostnames and IP addresses in the Internet. DNS names are constructed and resolved, and updates are propagated across DNS servers in a hierarchical manner. The PAN forwarder ID space is flat because forwarders use self-generated, self-certifying identifiers. This means PAN directory servers can neither take advantage of the hierarchical approach of DNS nor can perform aggregation of forwarder identifiers as they propagate forwarder information through the directory service. The latter approach is that used by BGP [32], which aggregates prefix information to reduce

the number of entries BGP has to carry and store.

IMPROVING INTERNET DOMAIN ROUTING. Detour [30] is a system of geographically dispersed router nodes interconnected using tunnels. Detour aims to eliminate some of the inefficiencies in routing and transport protocols that exist in the Internet. Detour nodes exchange information about latency, bandwidth, and drop rate to allow applications to use alternative paths and to route around hot spots in the Internet. Similarly, Resilient Overlay Networks [3] address certain limitations of the interdomain routing protocol BGP [28] and HLP [34] proposes a hybrid link-state path-vector routing protocol as a design alternative for a next-generation BGP. Like PAN, these systems help overcome network obstructions due to Internet routing limitations. However, their purpose is to provide a means of finding alternate routes more quickly and more effectively, rather than to create a means of accessing otherwise inaccessible resources.

VIRTUAL PRIVATE NETWORKS (VPNS). VPNs allow users to appear to be on a remote network, generally for the purpose of accessing resources only accessible to hosts on that network [14], [27]. PANs provide this, but they provide two other useful features as well: a *directory service* that allows users to specify the perspectives that they want by their characteristics, and a *routing infrastructure* that can deliver traffic to the desired perspective even if the network is fragmented in a manner that prevents the user from communicating directly with the server providing the perspective [15].

III. ARCHITECTURE

PAN consists of a pairwise-connected overlay network of *forwarders*, each of which has access to some set of Internet resources. Some resources may be available to some nodes but not to others. The overlay network that connects all of the forwarders to each other includes a *data plane* that carries tunneled DNS requests and TCP sessions, as well as a *control plane* that carries routing information.

There are two key problems with a distributed approach to assigning names in a network. First, two network components may have the same name, and second, there are performance costs associated with choosing names that do not inherently carry location information. However, we suggest that it is both possible and beneficial to sacrifice universal naming by allowing access to resources whose names are locally governed.

To address the concern about uniqueness of names used to identify forwarders, we allow each forwarder to generate a self-certifying identity (such identities may be mapped to human-readable nicknames by third-party certification authorities). Each forwarder, then, possesses two names: a *global* name, used to identify itself within the PAN network, and a *local* name, used to identify itself within its local namespace. By considering that each forwarder provides access to resources within its own local namespace, we avoid requiring that all names for all Internet resources be globally unique.

To specifically identify each Internet resource, we combine the locally meaningful name of the resource (e.g., a hostname such as `www.google.com`) with an identifier specifying the name of the forwarder from which we want to access that resource (e.g., the self-certifying name of a forwarder, like `79f72ae5`). In our prototype, Blossom, we assume that resources are named by hostname or IP address, so to access a resource listening on TCP port `80` of `bar.target.org` as seen by a forwarder named `79f72ae5`, we would represent the resource as `bar.target.org.79f72ae5.exit:80`.

In Figure 1, we show how a user would access this resource. The user's PAN client (indicated by `foo.source.net` uses the PAN directory service to obtain a source route suitable for building a circuit through the set of PAN forwarders to forwarder F4 (with ID 79f72ae5). Once the circuit from the client through F1, F2, F3, and F4 is established, F4 issues a DNS lookup of the local name `bar.target.org` to obtain an IP address from its point of view and accesses the request on behalf of the client. The client thus accesses the resource through F4.

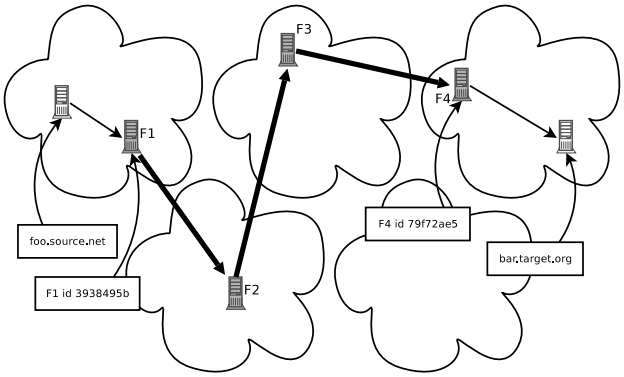

Fig. 1. ACCESSING A RESOURCE. *After making use of the PAN directory servers, a client system has a source route suitable for building a circuit through the set of forwarders to the last-hop forwarder, through which the client can access the (otherwise occluded) Internet resource.*

A. Directory Tables

Some PAN forwarders also serve as directory servers, and every PAN directory server is also a forwarder. Each directory server provides a set of *records*: (a) a *master record*, containing attributes describing itself, (b) a set of *directory records*, each containing attributes describing directory peers, and (c) a set of *forwarder records*, each containing attributes describing individual PAN forwarders. The directory server publishes these records by responding to queries in the form of HTTP-GET requests, and these attributes are periodically pushed to neighboring directories via directory updates in the form of HTTP-POST requests.

1) Master Records: A complete PAN directory server listing includes exactly one *master record*, which contains three attributes, as follows: a *header* consisting of the name of the directory server and its version, a *timestamp* indicating when this directory listing was created, and a *status* record identifying each forwarder indexed by the directory, including a bit that indicates whether the directory believes that forwarder to be active. The bit specifying whether a given forwarder is reachable is set to true when the directory server receives a

sufficiently recent record for an individual forwarder, and it is set to false when the record expires.

2) Directory Records: Each PAN directory server publishes a number of *directory records*, each containing a set of attributes that describe a specific peer directory server. A directory server accrues a set of directory records over time via directory updates from its neighbors. Unlike peer-to-peer file-sharing services such as Gnutella or BitTorrent, PAN is designed with the goal of balancing scalability with minimization of connection setup latency for clients connecting to services. Thus, clients do not request forwarder records via broadcasting or heuristic searches; instead, each directory maintains a set of directory records, each uniquely corresponding to one of its peers. Scalability dictates that each individual directory server need not know everything about the entire network, so there is no guarantee that each directory server contains a record for each other directory server in the entire network.

When a client issues a query for a forwarder record, but a directory server has no corresponding forwarder record, the directory server may refer the client to a set of directory servers that have previously indicated knowledge of forwarder records matching the request of the client. Such *referrals* are not arbitrary: clients seeking a particular forwarder record will be sequentially referred to some subset of the set of directories along the reversal of the path by which the advertisement of the forwarder propagated through the network. Each directory record describes a directory server and contains the following attributes:

- SERVICE-DESIGNATION. This field tells a client how to connect to a directory server, given that the client has already constructed a circuit to the forwarder residing on the same machine as the directory server. In our present implementation, this field is a TCP port number.
- PROPAGATION-PATH. This field contains an ordered list of directory servers (starting with the origin) through which this particular directory record has propagated before reaching the directory server upon which it presently resides. The primary purpose of this field is to avoid cycles in the propagation of directory records.
- SUMMARY. This field provides a list of PAN forwarders to which the directory server offers to forward traffic. For each forwarder in the list, this attribute also includes *one possible forwarding path* leading to that forwarder.
- COMPILED-METADATA. Every forwarder has metadata that describes the perspective it provides. This metadata could be jurisdictional location (such as country name), geolocation (latitude/longitude coordinates), network name, etc. The *Compiled-Metadata* field is a list of metadata attributes (i.e., perspective attributes) representing the union of all of the *Metadata* attributes corresponding to all the forwarders that appear in the *Summary* field of this directory record. For each *Metadata* attribute, this field also includes *one possible forwarding path* leading to a forwarder whose perspective has that attribute. Therefore, as in INS [1], clients can query for forwarder records matching some particular metadata in addition to querying for specific forwarders by name.

As an optimization, a PAN client may use the forwarder-specific or perspective-specific forwarding path information in *Summary* or *Compiled-Metadata* fields, respectively, to build circuits toward a given forwarder or perspective without querying directory servers along the path (provided that the client has access to sufficiently recent forwarder records for the constituent forwarders). This can potentially improve circuit setup latency, but there are tradeoffs. First, a client choosing this option does not receive information about possible alternative paths, thus waiving its option to choose its path from the set of advertised possibilities. Second, the path is not guaranteed to work; inconsistency resulting from slow routing convergence may allow forwarding paths that are no longer applicable to persist for some time in the *Summary* and *Compiled-Metadata* fields offered to clients by directory servers.

3) Forwarder Records: When a PAN forwarder publishes its descriptor (described below), metadata, and connection information to a directory server, the directory server in turn creates a forwarder record using that information. Each forwarder listed in a directory has exactly one corresponding forwarder record. In general, forwarder records are updated more frequently and propagated less widely than directory records; see Section III-C for details. A directory server must publish a forwarder record for itself. Each forwarder record contains some subset of the following fields:

- FORWARDER DESCRIPTOR. Descriptors are self-signed statements published by forwarders that contain contact information, including IP address and port for accepting circuit-building connections, public key, and salient information about the capabilities of the forwarder, including exit policy and bandwidth measurements.
- PROPAGATION-PATH. This field contains an ordered list of directory servers (starting with the origin) through which this particular forwarder record has propagated before reaching the directory server upon which it presently resides. The primary purpose of this field is to avoid cycles in the propagation of forwarder records. The value of this attribute may be empty, in which case the propagation path for this particular forwarder record is presumed to be the empty list (i.e., the forwarder described by this record published its information directly to the directory server upon which this record presently resides). Note that this path is not necessarily the same as that provided by the *Forwarding-Path* attribute (described next).
- FORWARDING-PATH. This field contains an ordered list of forwarders indicating the circuit that a client should construct to reach the forwarder described by this record, starting with the forwarder closest to the current directory server. Differences between this list and the list provided by the *Propagation-Path* attribute arise in two ways. First, directory servers through which a forwarder record propagates are not required to add their names to the forwarding path. Second, the PAN architecture allows forwarders to publish their descriptors to directory servers in locations from which those forwarders are not directly accessible; to address this, the forwarder may provide instructions by which clients can reach it from the perspective of the directory to which it publishes its information.

- METADATA. This attribute provides additional perspective information (e.g., geographic region, network name, connectivity information, access to particular resources, etc.) describing the forwarder.

B. Client Interaction

Our implementation of PAN leverages the circuit-building module of Tor [10] to instruct a running Tor process to build a circuit through the overlay of PAN forwarders. (Tor provides its own directory service, but PAN does not make use of it.) To see how the various components interact, refer to Figure 2. The main PAN client process itself does not interact with client applications directly; instead, it communicates with PAN directory servers using specially-built Tor circuits, and it uses descriptors obtained from these conversations to instruct Tor to build circuits that client applications can use. To take advantage of PAN, client applications may need to interact with an application-specific proxy that assures that requests for network resources are semantically correct. For example, a proxy for a web browser might rewrite HTTP headers to excise the PAN forwarder request from the hostname fields. Similarly, the same proxy might rewrite HTML tags containing URLs to ensure that all links on a page are accessed via the same PAN directives when clicked or loaded automatically.

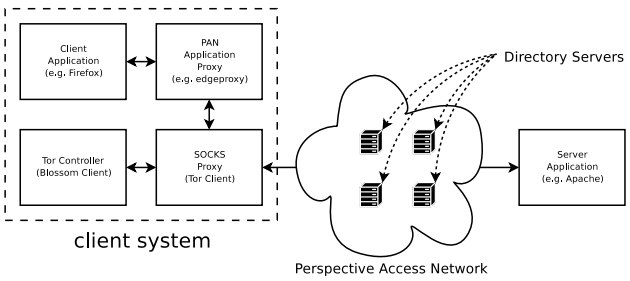

Fig. 2. CLIENT PERSPECTIVE. *Client applications communicate with PAN via a series of proxies; PAN consists of software (a program that controls a running Tor process) as well as a service (the perspective access network itself).*

1) Issuing Queries: To establish a path to a specified exit point, PAN must first determine the path to the exit point and obtain descriptors for each of the forwarders along that path, including the last one. Sufficient information necessary to learn a path to a given destination and all of the requisite descriptors may be available from the directory server to which the client speaks directly. Otherwise, the client will need to obtain the missing information via a series of queries to directory servers. Each time that a client queries a directory server A and is referred to a neighboring directory server B for more information, the client extends the circuit used to communicate with A to B, thus adding a single hop to the circuit. See Figure 3.

There are two types of queries, *explicit queries* and *perspective queries*. *Explicit queries* request a path to a particular forwarder whose name matches a given string, indicating that the client wants to build a circuit that terminates at some specific last-hop forwarder. *Perspective queries* request a path

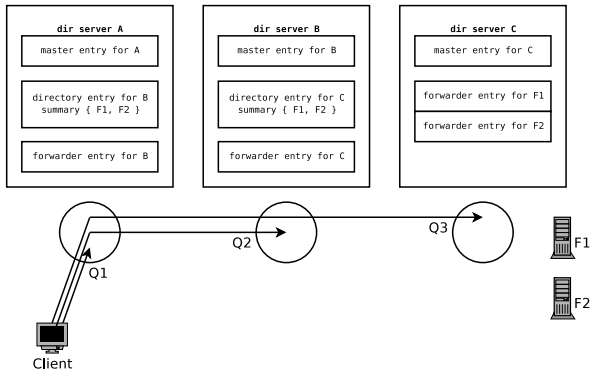

Fig. 3. ISSUING QUERIES. *Suppose that a client application requests a service as seen by forwarder F_2, and the PAN client is configured to use directory server A. The client first sends a query to A, who responds with a referral to B. The client next sends a query to B, who in turn refers it to C. Finally the client sends a query to C, who has the descriptor for F_2. The client then uses the resulting circuit through $\{A, B, C\}$ to extend the circuit to F_2 and connect to the target service via F_2.*

to a forwarder with certain attributes in its corresponding *Metadata* field, indicating that the client wants to build a circuit that terminates at any last-hop forwarder that matches some set of criteria. Note that directories control the content of *Metadata* fields within forwarder records, so, for example, a client issuing a perspective query may choose to reject a circuit to a specific forwarder if its forwarder record does not contain a metadata record matching the original request.

The contract between a directory server and a client issuing a query is as follows. If a client issues a query, then a response from the directory server must include the following:

- (a) a forwarder record for a forwarder that matches the query,
- (b) (in the event of an explicit query) a set of directory records and their corresponding forwarder records, such that each directory record contains the given forwarder name in its *Summary* field,
- (c) (in the event of a perspective query) a set of directory records and their corresponding forwarder records, such that each directory record contains an element that matches the query string in its *Compiled-Metadata* field, or
- (d) an empty list of records, indicating that the query was unsuccessful.

Finally, a directory server may be configured to interpret a query as *recursive*, meaning that the directory server may issue the query on behalf of the client just as with recursive queries in DNS. A client may specify to the directory server that it intends for its query to be non-recursive, in which case the directory should honor that request (to avoid the chance that a cached entry might be wrong). We discuss design tradeoffs of recursive queries in Section IV.

2) Building Circuits: In our prototype, once it has obtained forwarder records for the entire path to the last-hop forwarder, the PAN client will provide the necessary descriptors to Tor and then ask Tor to build a circuit using those descriptors (see Figure 1). Once the circuit has been built, PAN will inform Tor

that the TCP stream received from the client application should be attached to the newly constructed circuit. We have used our Blossom prototype implementation to confirm that the set of web pages accessible from some ISP in China differs from the set of web pages accessible from some ISP in Boston.

C. Directory Protocol

The directory servers propagate both forwarder records and directory records to other directory servers throughout the system. In this manner, any client using any of the directory servers throughout the system will have a measure of assurance that it can build a circuit to its requested forwarder, provided that directory server configuration permitted the propagation of routing information.

Directory records are stored as *long-term state* that is assumed to be up-to-date unless a *Directory Update* request from a neighboring directory server is received. The message volume involved in maintaining synchronicity of routing information can be expensive, so a directory periodically pushes the changes to its neighbors. Reliability is achieved by stipulating that if a directory server *A* fails to successfully send an update to a particular neighboring directory server *B*, then *A* will consider *B* to be offline. When a directory comes online, it requests a *burst* from each of its neighbors to bootstrap its knowledge of the records advertised by each of its neighbors. The burst contains the neighbor's master record, all of its hard state (i.e., all directory records), and its own forwarder record. After receiving the bursts, the requester applies a path-selection algorithm to determine the set of records that it should propagate, and it updates each of its neighbors with this set of records. Subsequently, the directory will only receive *directory updates* from its neighbors when individual records change. Each time the directory server receives a directory update that results in a change to its own set of records, that directory server must notify its neighbors about the change within a reasonable period of time (unless filtering and aggregation rules obviate the need to update a neighbor about the change – see Section IV-D).

Forwarder records are stored as *short-term state* that is periodically refreshed, since forwarder descriptors change frequently and individual forwarders themselves may join and leave the network frequently. Individual forwarder records must be periodically re-issued: if a forwarder record expires before it is replaced, then directory servers should discard it.

Periodically, directory neighbors send empty updates to each other even if they have no directory changes to send. Such empty updates are *keepalive* messages. If a directory has not heard from one of its neighbors for a sufficiently long period of time, it concludes that the link to the neighbor has been severed and responds by issuing a *withdrawal* message to its peers indicating that the directory record is no longer available. Withdrawal messages carry valid *Propagation-Path* attributes, and any directory server *A* that currently offers a directory record whose *Propagation-Path* attribute contains the name of a neighbor *B* about which it received a withdrawal message must propagate to its other neighbors either a message announcing the withdrawal of *B* or an ordinary directory record with a *Propagation-Path* attribute that does not contain *B*. In this manner, all directory servers that have selected the withdrawn route will be informed of the change.

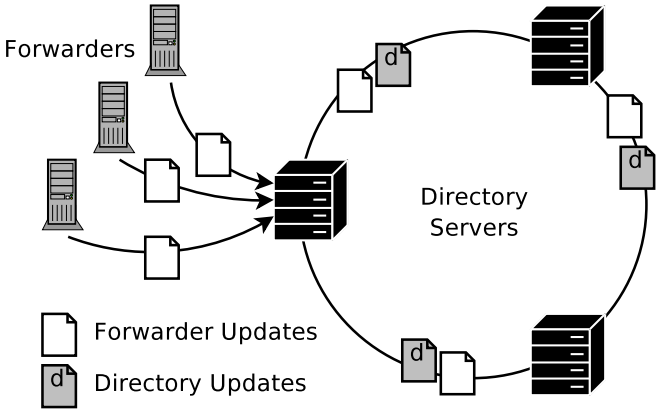

Fig. 4. DIRECTORY PROPAGATION. *Each forwarder publishes its forwarder record to some set of directory servers, and each directory server publishes its directory record to its neighboring directory servers. Directory servers propagate both kinds of records according to their individual policies.*

1) Directory Propagation: Both directory records and forwarder records are propagated using a BGP-like path-vector protocol that includes a simple route selection algorithm applied at each directory server. Figure 4 illustrates the process by which route information is propagated through the network. Each forwarder advertises its forwarder record to some set of directory servers, and directory servers propagate the forwarder record through the network as far as policy permits. Forwarders that are also directory servers advertise only to themselves. Each directory server creates a directory record for each of its neighbor directory servers and propagates the record through the network. Thus, forwarders push forwarder records to directory servers, and directory servers push both forwarder records and directory records to other directory servers.

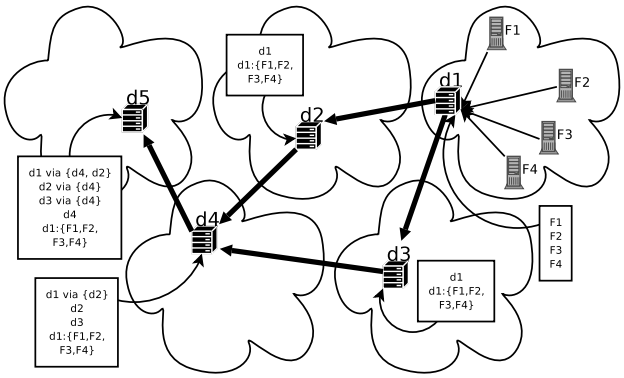

Fig. 5. ADVERTISING PAN FORWARDERS. *PAN directory servers use a path-vector algorithm to propagate contact information for PAN forwarders. Black lines indicate the path taken by an advertisement initiated by the directory server labeled d_1. The boxes represent the records stored at the various directory servers, including Propagation-Path and Summary attributes of directory records.*

If a directory server receives two conflicting forwarder

records (e.g., two records with different attributes for the same forwarder), then it chooses to propagate the one whose *Forwarding-Path* attribute has the shorter length. Figure 5 provides an overview of how forwarder information propagates in the general case. The specific configuration of individual directory servers may cause exceptions to these rules; Section IV-D discusses this in greater depth.

2) Directory Requests: Directory servers address five different kinds of requests, all issued using HTTP/1.1 [12]:

- COMPLETE LISTING. This is a request by a PAN client for the entire set of records, including its master record, all directory records, and all forwarder records. The response to this request can potentially be quite large, but query overhead for a client could be reduced substantially if most of the forwarders to which it desires to build circuits have forwarder records published on the same directory server.
- DIRECTORY BURST. This is a special request sent by a directory server when it first comes online to bootstrap its knowledge of the records advertised by each of its neighbors. A directory server responds to this request by providing a master record, all of its hard state (i.e., all directory records), and its own forwarder record.
- QUERY. This is a query from a client or directory server for a forwarder record, either explicitly (by name) or implicitly (by metadata or descriptor-derived data field).
- PUBLISH FORWARDER RECORD. This is an HTTP request from a forwarder to upload a complete forwarder record.
- DIRECTORY UPDATE. This is an HTTP request from a neighboring directory server to upload status changes (deltas) since the most recent successful update.

IV. DESIGN CONSIDERATIONS

The design of directory servers and propagation of routing information is more challenging in PAN than in BGP for several reasons:

- While BGP routing table entries consist of IPv4 prefixes, PAN routing table entries consist of *attribute sets*, a richer domain describing what can be reached using the network.
- PAN directory servers have the additional property that they provide information directly to clients as they construct source-routed circuits.
- While IPv4 prefixes are assigned by a central authority, there are no central authorities dictating the allocation of perspectives.
- Managing policies in PAN is more complex than in BGP. The PAN policy framework, described in Section IV-E, applies the techniques used to assign policy in BGP routing to this richer PAN domain.

The performance, scalability, and effectiveness of PAN largely derives from the design, implementation, and configuration of its directory servers. We consider the important issues in this section.

A. Structured versus Unstructured

Perhaps the first design question about our directory service is, considering the extensive research in distributed hash tables (DHT), whether we should implement our directory service using a structured network with $O(\log n)$ lookup operations rather than an unstructured network with fewer performance guarantees.

There are several problems with using a DHT. We list two here. First, DHTs assume a full mesh of connectivity. We want to allow an unstructured, organic growth of our network. Imagine, for example, using DHT to propagate BGP routing tables. This would be necessarily impossible, because the DHT itself requires some notion of connectivity that does not exist until the underlying network itself about which the DHT carries information is in place.

Second, one of the key characteristics of DHT systems is the use of a uniform hash function to uniformly distribute load across servers, and the hash function, which dictates which servers get which load, is essential to the $O(\log n)$ routing performance. However, the information that PAN stores is, to a large extent location-dependent, and that location-dependence is, after all, the reason for our service. It would be detrimental to scalability and deleterious to server incentives to store information haphazardly throughout the network, when it makes more sense for individual directory servers to just store the information relevant to themselves.

B. Propagating Forwarder Information

Propagating the self-certifying name of each forwarder carries the advantage that clients may explicitly specify each forwarder individually. However, this advantage comes at a cost, since self-certifying names cannot be aggregated. The result is that individual directory servers must contain at least the name of each forwarder in the entire network, so that they can appropriately respond to explicit queries requesting any individual forwarder. But, we can further relax the assumption that each directory server knows about each forwarder by allowing directory servers the option of propagating only metadata, rather than entire summary records. Naturally, metadata fields may contain the names of the forwarders themselves, but we rely upon the discretion of the individual directory servers to negotiate which information is propagated through the network. The effect of limited, policy-driven propagation may be that directory servers proximate to a given set of target forwarders may be configured to propagate their names and metadata while directory servers farther from the forwarders may be configured to propagate metadata only, to improve network scalability. The result would be that directory servers nearest to a given forwarder would contain all of the information needed by a client who desires to build a circuit to that forwarder. Directory servers somewhat farther from the forwarders might not have descriptors for the forwarders themselves, but might possess *Directory Records* containing *Summary* attributes that provide enough information for clients to issue queries for the individual forwarders by name. Finally, directory servers in regions most distant from the forwarders might not have knowledge of the names of the individual

forwarders themselves, and might only have metadata describing the forwarders collectively. Clients using directory servers in these regions would have no means of specifying those particular forwarders explicitly, but would only reach them in aggregate, by querying for attribute sets rather than explicit names.

Whether propagation of metadata is sufficient to assure reasonable scalability for PAN depends upon how PAN is used. For example, BGP scalability is limited by the number of independently propagated prefixes. Aggregation helps to some extent, since each prefix may correspond to thousands or even (theoretically) millions of individual hosts, but as we consider shorter prefixes, it becomes clear that at some level, the hierarchy ends, leaving each individual BGP listener with a table containing hundreds of thousands of distinct prefixes.

If the set of PAN forwarders were arranged such that there were exactly one per BGP prefix, with each forwarder as a directory server, and if peering relationships among directory servers topologically corresponded to peering relationships among autonomous systems, and if each client expected the ability to identify each PAN forwarder explicitly, then in theory the scalability of the Perspective Access Network would be essentially the same as that of the BGP network that exists today. However, this pattern of deployment and usage might not be what we can expect in a future PAN. Also, it is possible for multiple PANs to exist concurrently; private organizations might deploy their own PANs for their exclusive use.

For example, we might imagine that PAN would be used to link private networks, in which case we might assume that there would be one PAN forwarder in each private network. Since there are millions of private networks of this sort, an assumption that each would require the ability to identify each other explicitly could seriously constrain the scalability of PAN. However, we can resolve this by stipulating that clients who want to access specific destination forwarders know *a priori* how to reach directory servers that contain the necessary information for learning how to construct circuits that terminate at those specific forwarders. (Such instructions could be preconfigured in the PAN software at the time of distribution, for example.) PAN provides an architecture that allows communities of this sort to develop without overconstraining their structure.

Another use of PAN might be to have individual volunteers provide views of the world to be used at a high level of granularity; for example, clients might specify the names of particular countries or particular ISPs. In this situation, the exclusive propagation of metadata improves scalability considerably.

C. Responding to Queries

Suppose that a client issues a query for information that a particular directory server cannot provide but knows how to find. The directory server then has a choice. It may issue a *referral*, telling the client how to retrieve the information itself from other directory servers in the network, or it may treat the query *recursively*, forwarding the request on behalf of the client, and ultimately responding to the client with the information in the same manner that it would had it possessed the information at the time at which it received the query.

The difference between recursive and non-recursive (referral) responses to queries is comparable to the difference between their analogues in DNS. Referrals have the advantage that directory servers do less work, so servers under heavy load may wish to use this method. Recursive queries have the advantage that clients do less work and directory servers may cache the results. An enterprise may want to deploy servers that support recursive queries to allow clients to take advantage of requests made earlier by other clients if available, and possibly avoid some extra network traffic in the general case.

D. Filtering and Aggregation

A number of parameters govern how individual PAN directory servers interact with forwarders, clients, and their peers. These parameters include *policy directives* that allow operators of directory servers to control aggregation, specify which information to propagate by attribute and propagation path, and manage network resources.

Recall that directory servers have control over the contents of *Metadata* and *Compiled-Metadata* attributes. *Filtering* and *aggregation* rules instruct directory servers how to adjust the values of these attributes. These rules are configured as part of the policy configuration described in Section IV-E.

Filtering rules configure a directory server to *filter* certain metadata. This may be desired if a directory server chooses not to propagate certain kinds of perspective information to certain other directory servers. Aggregation rules configure a directory server to aggregate metadata carrying perspective information to improve scalability. Two forms of aggregation are possible. The first form of aggregation involves collapsing substantively identical nodes (i.e., same attributes) into a single attribute set and advertising that attribute set. Since substantively identical nodes offer the same perspective as far as a client is concerned, no information is lost in this process. The second form of aggregation involves collapsing substantively similar, but not identical, nodes (i.e., partially matching attributes) into a more general attribute set by *single-attribute aggregation* or by *subdivision*, as discussed in Section V-C, which may be considered a special kind of aggregation.

Information is lost as directory servers decide what information to discard (i.e., the extent of filtering and aggregation) to reduce the number of distinct sets of metadata to a reasonable value. The directory server should then set a flag indicating what data has been discarded, so that downstream directory servers can continue the same aggregation if they so choose, and so that clients have a hint about what upstream directory servers have answers to more specific queries.

E. Policy Framework

The configuration of each directory server includes a *policy* that defines which routes to accept, which routes to propagate, how to assign preferences among routes, and any bandwidth caps to apply while routing traffic. We use the Routing Policy Specification Language (RPSL) as a starting point [2]. By

selecting the relevant features of RPSL, adapting them to handle perspective descriptions, and adding some features to improve incentives for deployment, we propose a Perspective Routing Policy Specification Language (PRPSL), a form of RPSL adapted for use with PAN directory servers. We refer the interested reader to a more detailed treatment [15].

V. Evaluation

To illustrate some of the design tradeoffs inherent to the PAN directory service, we performed empirical measurements using a deployment of our prototype implementation (called Blossom [2] of roughly 300 nodes on PlanetLab. In our experiments, each of the nodes serves as a forwarder in the PAN overlay, and some subset of the nodes also serve as directory servers. We refer to nodes that perform just forwarding as *standalone forwarders*.

For each of our experiments, we assigned forwarders and directory servers at random from the set of PlanetLab nodes that we had previously determined to be responsive. Our selection process assigns forwarder roles randomly, so the topologies that we chose are *conservative* in the sense that pairs of nodes that directly communicate with each other are determined without regard to the underlying network infrastructure. We suspect that pairwise communicators in most PAN networks deployed in practice would be chosen more intelligently.

A. Circuit Setup Performance

To test setup latency for circuits involving multiple hops through the forwarding network and the effect of client queries on path setup time, we generated some paths of various lengths using randomly chosen PlanetLab nodes and constructed circuits using those paths. Using these paths, we performed two experiments:

- Generic Circuit-Building Test. We tested the time taken for Tor to build a circuit for a specified path by requesting to send TCP traffic to some port on the final node in the circuit. The results of this test are represented as solid triangles in Figure 7. Each triangle represents the median observed TCP connection setup latency using predetermined circuits of that particular length over ten independent trials. We are interested in using PAN for interactive applications, and by comparison, the International Telecommunications Union recommends an average call setup delay of eight seconds for international calls via the ISDN [19]. (Observe that users of the popular Tor network tolerate substantial circuit setup latency as a matter of course.)
- Circuit-Building Test with Queries. In our second experiment, we tested the time taken for Tor to build a circuit according to a path that the PAN client determines by iteratively issuing queries to each successive directory server along the path to the final node in the circuit. The results of this test are represented as hollow circles in Figure 7. Each circle represents the median observed TCP connection setup latency using dynamically determined circuits of that particular length over ten independent trials. In each case, the number of queries performed is equal to the number of hops minus one. Note that connection setup consistently takes longer when the PAN client performs queries.

[2]The source code for Blossom is available at `http://afs.eecs.harvard.edu/~goodell/blossom/`

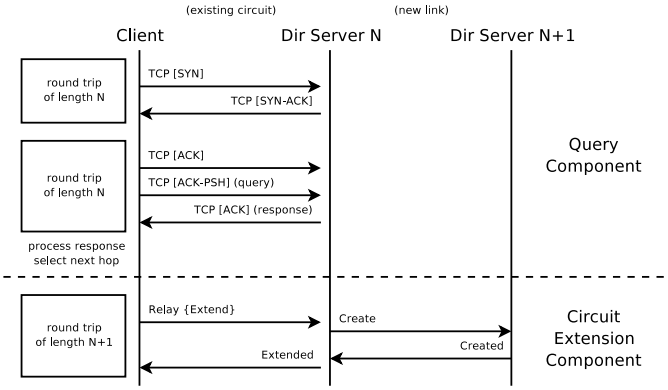

Fig. 6. Extending a Circuit (with Querying). *Clients that do not already know the next hop in the circuit must first send a query to the current directory server before instructing Tor to extend the circuit.*

Whether a client will have to perform queries or not depends upon how directory servers within the PAN network are configured. Figure 6 illustrates the interaction that takes place between a PAN client and directory servers when the client extends the circuit from length n to length $n + 1$. The top portion of the interaction, marked "Query component," only occurs when the client issues a query before extending the circuit.

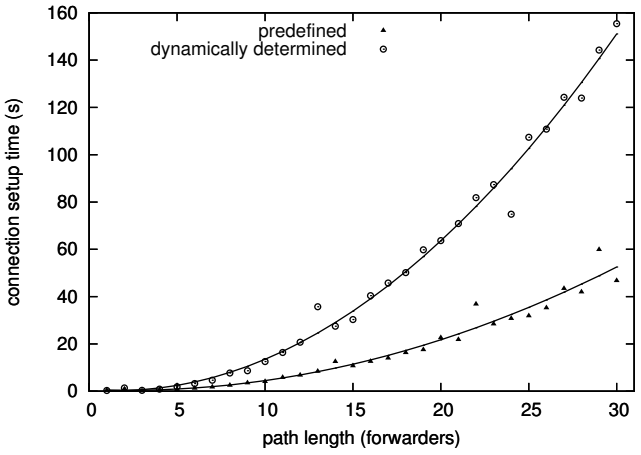

Fig. 7. Circuit Setup Latency. *Time taken to build a circuit and establish an end-to-end TCP session for circuits of varying lengths. The solid lines represent quadratic least-squares regression curves for the two experiments.*

In both cases, since the process of extending a circuit from length n to length $n + 1$ involves sending messages back and forth over the entire $O(n)$ length of the circuit, the circuit setup time scales quadratically with the length of the circuit. The two parabolic lines in Figure 7 correspond to a quadratic least-squares regression of the data from each

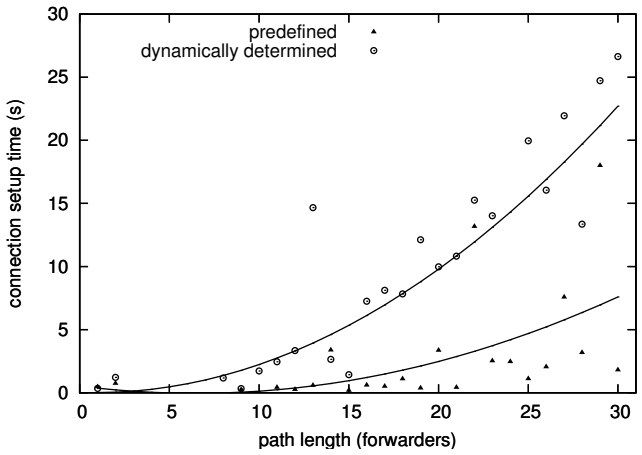

Fig. 8. CIRCUIT SETUP LATENCY, ADJUSTED FOR NETWORK DELAY. *This graph presents the same experiments as Figure 7, but adjusted to remove the round-trip times introduced by network delay. Note that the scale of the y-axis differs from Figure 7.*

of the two experiments, respectively. Observe that queries introduce noticeable additional latency, particularly as circuit length increases. Figure 8 presents the same results, except subtracting the expected network delay times between pairs of nodes (i.e., all of the round-trip times indicated in Figure 6). We obtained the pairwise network latency values from a set of measurements conducted by C. Yoshikawa.[3]

The circuit-setup experiments involved randomly-chosen PlanetLab [18] nodes. The results are thus conservative because the neighboring nodes are chosen without regard for the underlying network topology. We suspect that in actual PAN networks, administrators of PAN directory servers would arrange themselves in a less random, more advantageous topology. Observe that network latency accounts for the vast majority of delay associated with connection setup. Unfortunately, there is no way to avoid this delay; the only solution is to improve the underlying network. However, Figure 8 shows that system-internal delay accounts for some portion of the time spent during circuit setup, and this particular delay can potentially be improved by reimplementing our prototype. Note that this delay will also increase with circuit length, since establishing longer circuits involves interaction with a greater number of directory servers, which scales linearly with circuit length, and more cryptographic operations, which scale quadratically with circuit length.

As described in Section III-A.2, when forwarder records providing access to the desired perspective do not exist, PAN clients **may** build a path based upon forwarder-specific or directory-specific forwarding path information contained in the *Summary* or *Compiled-Metadata* fields. Our experiments expose the following tradeoff: if a client tries to explicitly build a path based upon forwarding path information, it sacrifices some measure of control over the path as well as some confidence that the forwarding path information is accurate, but the process of querying all directory servers along the forwarding path degrades circuit setup performance.

[3]PlanetLab: All Sites Pings, `http://ping.ececs.uc.edu/ping/`

Overall, if we accept the ITU eight-second call setup delay recommendation for the PAN circuit construction process, our experiments illustrate that for sufficiently short circuit lengths (up to eight hops for dynamically determined circuits, up to twelve hops for predefined circuits), circuit setup latency is reasonable for human users. Ultimately, the circuit length is largely influenced by the type of application for which the PAN is used. We describe applications of PAN next.

B. Essential Applications

To assess the most important applications of PAN, we focus on five essential uses of PAN.

- CIRCUMVENT POLITICAL FILTERING. PAN provides a tool that can be used to promote human rights. Authoritarian regimes and network access providers sometimes monitor or restrict access to Internet content for political reasons. Parties interested in providing access to restricted content to dissidents and others can deploy PAN infrastructure so that people whose attachment points to the Internet ordinarily subject them to such monitoring or filtering can access Internet content as if they were in other parts of the world. For example, in China, access to resources varies widely among ISPs, since there is no consistent policy that is applied centrally throughout China's backbone, but a set of guidelines instead [25]. Thus, an organization like Open Net Initiative[4] can use a PAN to conduct clinical filtering tests. To do so, it would probably want to include jurisdictional location (e.g., country name) attributes in its PAN queries.
- ENTERPRISE. Organizations with multiple separate networks can use PAN to selectively extend the trust envelope to allow access across network boundaries. In particular, an enterprise may want to allow users to access an internal network segment in one branch office from another branch office.
- GEOGRAPHY-BASED PERSONALIZATION. Since we know that the Internet is not consistent, there may be a market for Internet perspectives. For example, website internationalization or targeted advertising are sometimes a function of geolocation. Travelers far from home may be willing to pay to view the Internet as if they were home, so that they can have some assurance that the content they find is relevant to their interests. Similarly, a user may want access to targeted advertising and customized searches available in a location to which that user is planning a trip.
- DISTORTION OR PROJECTION OF LOCATION. A user may have an interest in appearing to be somewhere else for the purpose of determining what is accessible from a remote perspective. This can be useful for performing security audits, as it provides a means of appearing to be on the other side of firewalls and other policy-enforcing boundaries. This use can also be humanitarian; for example, Open Net Initiative periodically publishes a series of reports cataloging the extent and scope of Internet

[4]Open Net Initiative, `http://www.opennetinitiative.net/`

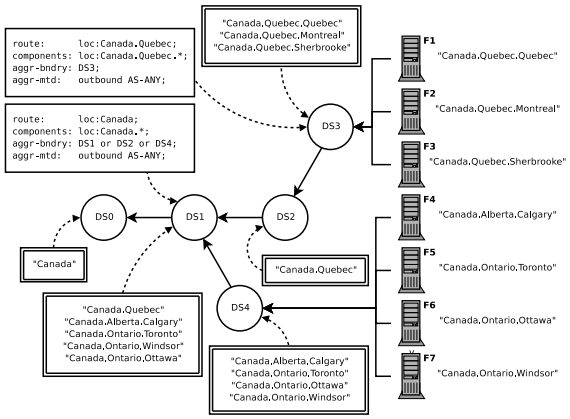

Fig. 9. PERSPECTIVE AGGREGATION. *Certain metadata, such as jurisdictional location are hierarchical and thus by definition aggregatable. Boxes with double-lined borders indicate the advertised attributes received from other directories and forwarders. Boxes with single-line borders indicate the aggregation policy w.r.t. an attribute. For example, DS3's policy states that any perspective within Quebec should be propagated as "loc:Canada.Quebec". Solid lines indicate propagation paths taken by advertisements.*

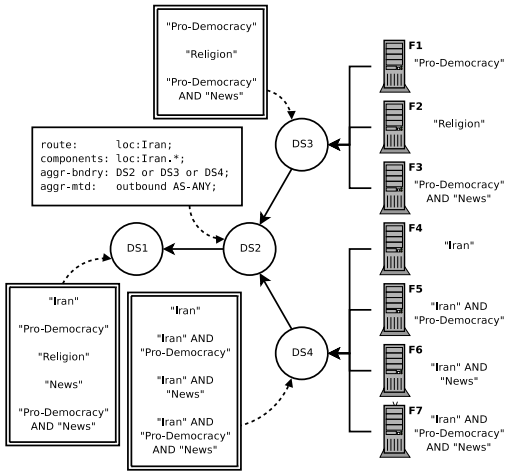

Fig. 10. SUBDIVISION OF PERSPECTIVES (1). *If a directory server receives a preponderance of perspectives with different combinations of some set of attributes, it can reduce the number of perspectives that it advertises by advertising the attributes separately. Boxes with double-line borders indicate the advertised attribute combinations that a directory server hears from other directory servers and forwarders.*

filtering in a number of nations [26]. Such cataloging requires perspectives from which to observe the filtering.

- TOPOLOGY-INDEPENDENT DMZ. An organization may want to externally provide some view of an internal part of its network, for example to provide access to some walled garden to the public or to industry collaborators. PAN provides the ability to provide an "internal DMZ" with all of the flexibility of remote access to a DMZ at the edge of the network but none of the topological constraints.

We have demonstrated the efficacy of PAN for two of the above applications: circumvention of political filtering [15] and geography-based personalization [16].

C. Aggregation Strategies

Aggregation promotes scalability; one reason not to aggregate when possible is to reduce the time required for clients to find the perspectives they seek. Small PAN networks do not benefit from aggregation enough to offset the cost of increased setup latency. As PAN networks expand in size, aggregation will become necessary to deal with the scaling issues. PAN provides the tools to perform aggregation where it is necessary for scaling, though for some semantic attribute categories, aggregation is not possible. Hierarchically-organized categories (e.g., jurisdictional location and network name) can by definition be aggregated. Flat categories, such as those describing filtering policy and functional capability, cannot.

Configuring directory server policy to aggregate hierarchical fields is straightforward. Refer to Figure 9. Observe that directory server DS3 receives perspectives located in various cities and then aggregates them all into a single announcement of `Canada.Quebec`. DS1 receives the aggregated perspective via DS2 as well as additional perspectives from DS4. DS1 subsequently aggregates all perspectives from Canada into a single perspective.

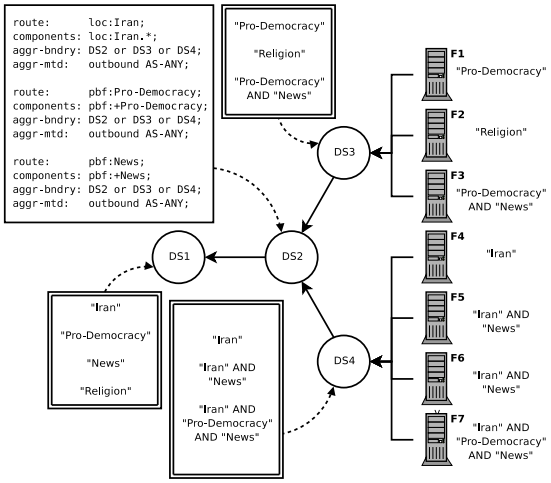

Fig. 11. SUBDIVISION OF PERSPECTIVES (2). *Advertising attributes separately may dramatically reduce the number of perspectives to advertise. Note that DS2 has no aggregation policy for* `Religion`*; by default, directory servers do not perform aggregation.*

In a PAN, individual perspectives may contain some number of attributes in each category and a user may ask for some particular combination of attributes. While we do not aggregate across fields to create the cross-product, we do allow individual directory servers to decide whether to *subdivide* a perspective that provides a particular combination of attributes, advertising the constituent attributes individually or in smaller sets. For example, a perspective that is located in Saudi Arabia and provides access to news stories might be advertised as two perspectives, one that is located in Saudi Arabia and one that provides access to news stories. Directory servers may use a *dynamic learning* procedure to determine which combinations of attributes are most popular as a basis for determining which sets to subdivide [15].

Figures 10 and 11 present a scenario in which a series of

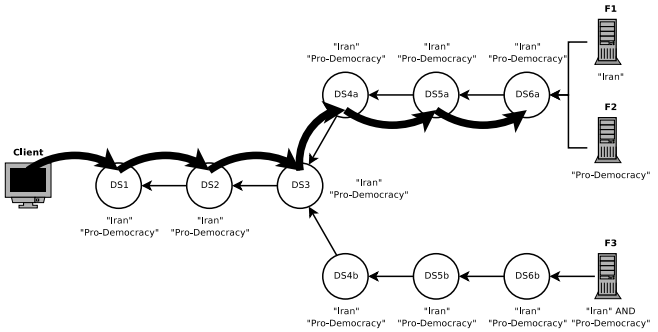

Fig. 12. CHOOSING AN UNCERTAIN PATH. *A client seeking a perspective containing a combination of attributes may issue queries along an incorrect path.*

forwarders advertise perspectives with various combinations of attributes denoting location ("Iran") and filtering policies that indicate they provide access to certain content types (e.g., "Pro-Democracy", "Religion", "News"). In Figure 10, directory server DS2 advertises "Iran" separately but still propagates advertisements of other attribute combinations. In Figure 11, directory server DS2 uses a policy such that it advertises each attribute separately.

The tradeoff resulting from aggregation or subdivision is that clients are not guaranteed to get the perspective that they want in one querying pass through the network. Figure 12 shows a client seeking a perspective located in Iran that provides access to "Pro-Democracy" content. While DS6a and DS6b both advertise that they provide both "Iran" and "Pro-Democracy" perspectives, only DS6b actually has knowledge of a perspective that provides both. When the client is in the process of learning the path, it is faced with a choice when it reaches DS3; suppose that it chooses DS4a as its next hop. Then, when the client reaches DS6a, it determines that the branch of the path following the decision point at DS3 is invalid. The client then must backtrack to DS3 and choose the other path which will lead it to a perspective that matches its query. We presume that after some number of unsuccessful attempts, the PAN client will abort and return an error condition to the application.

Observe that the client incurred a penalty for choosing the wrong path. Consider the following simple model that quantifies the penalty. Consider the directory server at which a client is faced with a choice among possible successive directory servers as the *decision point* (shown by DS3 in Figure 12), and consider the directory server at which a client learns with certainty the correctness (or incorrectness) of its circuit-building decision as the *aggregation point* (shown by DS6a in Figure 12). Suppose that there are n_d hops between the client and the decision point and $n_a(i)$ hops between the client and the aggregation point i. Let β denote the expected number of times that the client will have to backtrack before finding an acceptable circuit, and let n_a^* denote the average number of hops between the client and the aggregation point.

Next, suppose that A and B represent attributes, and a client wants a perspective with both attributes, but attribute A is not provided in a single advertisement with attribute B because of aggregation or subdivision. Suppose that the client finds a sequence of directory servers that advertise attribute A. Let $p(X)$ represent the probability of a given perspective having attribute X. Directory servers have knowledge of the number of entries with perspectives A and $B \cap A$, so:

$$\beta \approx \frac{1}{p(B|A)} = \frac{p(A)}{p(B \cap A)}. \tag{1}$$

Next, define $\tau(n)$ as the expected time required to build a circuit of length n. The value of $\tau(n)$ can be approximated by the quadratic regression curve depicted in Figure 7. For simplicity, we assume that all aggregation points are at the same distance from the client. Note that the client need not backtrack all the way to the start of the circuit, but only to the decision point, so backtracking requires expected time $\left[\sum_{i=1}^{\beta} \tau(n_a(i))\right] - \tau(n_d)$. Therefore, the expected time t that a client can expect to spend constructing a circuit to a perspective containing both attributes A and B is given by:

$$t = (1-\beta)\tau(n_d) + \sum_{i=1}^{\beta} \tau(n_a(i)) \approx \tau(n_d) + \frac{p(A)(\tau(n_a^*) - \tau(n_d))}{p(B \cap A)}. \tag{2}$$

Whether aggregation is sufficiently desirable to outweigh the performance penalty is determined by system usage over time and by the extent to which the impact on client performance outweighs the impact on directory service performance. In addition, it is possible for clients to improve upon the circuit setup time given in Equation 2 by considering multiple paths in parallel, but this improvement carries the potential for a substantial cost to directory servers and forwarders that must respond to unnecessary queries and build unnecessary circuits.

Finally, improvement over time in the technology of the directory servers themselves will continue to change the degree of aggregation that is required for scaling.

D. Security Considerations

Perspective Access Networks provide some security benefits. For example, the circuit-based design sacrifices stateless forwarding in favor of path authentication and resistance against man-in-the-middle attacks. In addition, perspectives can continue to exist even if an adversary filters access to some proportion of the PAN forwarders: in theory, as long as a path exists from the client to the desired perspective, PAN should be able to find a way to deliver the circuit.

However, the PAN infrastructure introduces some security vulnerabilities as well. For example, providing additional infrastructure components within the network introduces new services that can be attacked. Adversaries may choose to operate rogue forwarders or compromise existing exit forwarders. With control of an exit forwarder, an attacker could potentially monitor or modify the traffic between the exit forwarder and the application server. Adversaries may also attack directory servers for the purpose of returning invalid or misleading query results, injecting bogus route announcements, or discerning and cataloging which users are requesting which perspectives.

Another concern is that a determined adversary can systemically filter access to forwarders or directory servers within

a PAN. This means that if a repressive regime decided to block access to PANs by determining the set of PAN forwarders, it could do so; nonetheless, there are important reasons for designing PANs such that the network locations of the forwarders and directory servers are public. Furthermore, if a repressive regime were sufficiently paranoid, it could block all encrypted or unapproved traffic, relegating the use of PANs to steganography or covert channels. While some projects aspire to provide covert channels, PAN itself does not. Fortunately, case studies have demonstrated that Internet filtering is inconsistent [25], [26], suggesting that countries are either incapable or unwilling to systemically filter all access to circumvention technologies. For example, as of July 2006, the set of hosts not generally filtered by China includes most of the Tor network.

Considering the preponderance of incomplete attempts to filter access to Internet resources by category, we identify a set of useful countermeasures for dealing with a limited adversary. Consider an adversary that controls a network that traffic from PAN users in a particular region of the Internet must traverse. One countermeasure is to reveal the network locations of PAN forwarders sparingly, perhaps configuring directory servers in some regions of the Internet to only provide a limited number of forwarder descriptors per unit time. The challenge is that providing public access to a circumvention system means providing access to adversaries as well, and if adversaries know how to reach parts of the network, then adversaries can block the network. Releasing network locations incompletely and slowly over time creates a race between adversaries and regular users of the system. The optimistic vision is that while the set of nodes providing gateway access to the system may change, the fact that users continue to have access will not.

A second countermeasure is to "multiplex" Perspective Access Network directory servers with servers that provide other, "innocuous" content that a network infrastructure provider cannot afford to deny to its users. Specifically, a popular website could offer access to a PAN as an indistinguishable part of its service, forcing adversaries to choose between denying their users access to this website and denying access to the PAN.

A third countermeasure is to use the latest techniques for establishing covert channels as a generic platform, and send PAN traffic over the covert channels. Perspective Access Networks do not create covert channels, but this is not to say that they cannot interoperate with covert channels. Ultimately, PAN is not a complete solution for dealing with powerful adversaries seeking to deny access to circumvention technologies. However, it does provide a generic technique for describing which perspectives to access and constructing circuits to access these perspectives; this technique may have greater value to users subject to the whims of powerful network-controlling adversaries once better covert communication techniques have evolved. In the meantime, we believe that the three countermeasures will provide significant benefits.

For a more extensive security discussion, please refer to the thesis [15].

VI. Conclusion

Large-scale, public deployment of Perspective Access Networks could potentially have significant legal and economic effects. Additionally, PAN may have value in promoting end-to-end security models within both enterprises and the Internet at large.

However, there are also risks, commercial factors, and chilling effects that could potentially cause influential parties to discourage large-scale deployment and use of PAN. For example, many service providers actually intend to use network location as a means of differentiating and categorizing users, and deployment of Perspective Access Networks has the potential to confound their efforts. Of course, open proxies can be used to circumvent geography-based access restrictions today, but the proxies themselves are generally considered illegitimate because they usually run on compromised or misconfigured hosts. PAN could potentially bring circumvention into the mainstream, and once this happens there could be calls for ISPs to implement policies that disallow the operation of PAN forwarders.

Perhaps the most serious threat to network neutrality involves the possibility that ISPs might filter or restrict access to Internet content for commercial reasons. Indeed, Edward Whitacre, the CEO of SBC, has even suggested the possibility that both providers of content (e.g., Disney) and providers of services (e.g., Skype) ought to compensate the ISPs of their target audiences [24], [4] as part of a business model reminiscent of the cable television industry in the US. Clearly, the idea that ISPs should have the power to arbitrate which subset of the Internet to provide to its customers is very much alive. In fact, research has indicated that it is in the best interests of network providers to use compensation from content providers as a basis for discrimination among content providers, providing customers with inferior access or even no access to sites hosting particular content [35]. While network neutrality regulations have certain costs, there is little else to prevent ISPs from selectively discriminating.

Clark et al. suggest that a tool that allows Internet users to circumvent both provider-selected routing could be influential in shifting the balance of power [8]. Indeed, a Perspective Access Network can be used as such a tool, though it could potentially thwart useful price or service discrimination.

Since Perspective Access Networks may allow a user to select the most relevant geolocation, they may provide an opportunity to improve advertising efficiency, offering advertisers an incentive to support the proliferation of PANs. However, advertisers may have reason to oppose deployment of PANs if such deployment means the loss of ability to dominate a local market, and they may also opt to oppose deployment of PANs simply because they do not fully understand the business implications.

Perspective Access Networks provide a convenient means of providing access to otherwise restricted networks and providing end-to-end connectivity to pairs of Internet nodes that are not directly connected to each other. Moreover, with recent new threats to Internet consistency (governance disputes, geolocation services, DNS root disputes, and accidental

or deliberate censorship of resources), it is worth considering the design and implications of a radically different vision of the Internet—one without a well-defined core, consisting of fragments whose names and address spaces are not ordained hierarchically. Our work in building a PAN is a step in this direction, and our directory service architecture represents the core of this effort.

VII. Acknowledgments

We thank H.T. Kung, David Parkes, Roger Dingledine, Nick Mathewson, and the anonymous reviewers for their insightful comments on drafts of this paper. We also thank the National Science Foundation (CAREER Grant 0446522).

References

[1] W. Adjie-Winoto, E. Schwartz, H. Balakrishnan, and J. Lilley. The Design and Implementation of an Intentional Naming System. In *Proceedings of the ACM Symposium on Operating Systems Principles*, pages 186–201, December 1999.

[2] C. Alaettinoglu, C. Villamizar, E. Gerich, D. Kessens, D. Meyer, T. Bates, D. Karrenberg, and M. Terpstra. Routing Policy Specification Language (RPSL). Internet Engineering Task Force: RFC 2622, June 1999.

[3] D. Andersen, H. Balakrishnan, F. Kaashoek, and R. Morris. Resilient Overlay Networks. In *Proceedings of the Eighteenth ACM Symposium on Operating Systems Principles*, pages 131–145, Chateau Lake Louise, Banff, AB, Canada, October 2001.

[4] S. Bradner. Just When You Think Telecom Legislation Can't Get Any Worse, It Does. *Network World Weekly*, 21 November 2005.

[5] D. R. Cheriton and M. Gritter. TRIAD: A New Next-Generation Internet Architecture. `http://www-dsg.stanford.edu/triad/`, July 2000.

[6] D. Clark, R. Braden, A. Falk, and V. Pingali. FARA: Reorganizing the Addressing Architecture. *ACM SIGCOMM Computer Communication Review*, pages 313–321, 2003.

[7] D. Clark, K. Sollins, J. Wroclawski, and T. Faber. Addressing Reality: An Architectural Response to Real-World Demands on the Evolving Internet. In *Proceedings of ACM SIGCOMM 2003 FDNA Workshop*, August 2003.

[8] D. Clark, J. Wroclawski, K. Sollins, and R. Braden. Tussle in Cyberspace: Defining Tomorrow's Internet. In *Proceedings of ACM SIGCOMM*, August 2002.

[9] J. Crowcroft, S. Hand, R. Mortier, T. Roscoe, and A. Warfield. Plutarch: an Argument for Network Pluralism. *ACM SIGCOMM Computer Communication Review*, 33(4):258–266, 2003.

[10] R. Dingledine, N. Mathewson, and P. Syverson. Tor: The Second-Generation Onion Router. In *Proceedings of the Seventh USENIX Security Symposium*, August 2004.

[11] N. Feamster, M. Balazinska, G. Harfst, H. Balakrishnan, and D. Karger. Infranet: Circumventing Censorship and Surveillance. In *Proceedings of the 11th USENIX Security Symposium*, August 2002.

[12] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee. Hypertext Transfer Protocol: HTTP/1.1. Internet Engineering Task Force: RFC 2616, June 1999.

[13] H. W. French. Despite Web Crackdown, Prevailing Winds Are Free. *The New York Times*, 9 February 2006.

[14] B. Gleeson, A. Lin, J. Heinanen, G. Armitage, and A. Malis. A Framework for IP Based Virtual Private Networks. Internet Engineering Task Force: RFC 2764, February 2000.

[15] G. Goodell. *Perspective Access Networks*. PhD thesis, 2006.

[16] G. Goodell, S. Bradner, and M. Roussopoulos. Blossom: A Decentralized Approach to Overcoming Systemic Internet Fragmentation. Harvard Technical Report No. TR-10-05, May 2005.

[17] G. Goodell, S. Bradner, and M. Roussopoulos. Building a Coreless Internet without Ripping out the Core. In *Proceedings of the Fourth Workshop on Hot Topics in Networks*, November 2005.

[18] Intel Corporation. The Evolution of the Next-Generation Internet. `http://www.planet-lab.org/`, 2003.

[19] International Telecommunication Union. Network Grade of Service Parameters and Target Values for Circuit-Switched Services in the Evolving ISDN. Recommendation E.721, Telecommunication Standardization Sector of ITU, Geneva, Switzerland, May 1999.

[20] H. Lewis. Online Fraud Catchers: Protecting You but Maybe Also Getting Your Card Turned Down. `http://www.intelligentbanking.com/brm/news/ob/20000915.asp`, December 2002.

[21] P. Mockapetris. Domain Names: Concepts and Facilities. Internet Engineering Task Force: RFC 1034, November 1987.

[22] P. Mockapetris and K. Dunlap. Development of the Domain Name System. In *Proceedings of ACM SIGCOMM*, 1987.

[23] T. S. E. Ng, I. Stoica, and H. Zhang. A Waypoint Service Approach to Connect Heterogeneous Internet Address Spaces. In *Proceedings of the USENIX Annual Technical Conference 2001*, June 2001.

[24] P. O'Connell. At SBC, It's All About "Scale and Scope". BusinessWeek Online, 7 November 2005.

[25] Open Net Initiative. Internet Filtering in China 2004-2005. Open Net Initiative Case Study, June 2005.

[26] Open Net Initiative. Internet Filtering in Iran 2004-2005. Open Net Initiative Case Study, June 2005.

[27] H. Ould-Brahim, E. Rosen, and Y. Rekhter. Using BGP as an Auto-Discovery Mechanism for VR-Based Layer-3 VPNs. IETF Internet Draft, April 2006.

[28] Y. Rekhter and T. Li. A Border Gateway Protocol 4 (BGP 4). Internet Engineering Task Force: RFC 1771, March 1995.

[29] C. Rhoads. Endangered Domain. *The Wall Street Journal*, 19 January 2006.

[30] S. Savage, T. Anderson, A. Aggarwal, D. Becker, N. Cardwell, A. Collins, E. Hoffman, J. Snell, A. Vahdat, G. Voelker, and J. Zahorjan. Detour: Informed Internet Routing and Transport. *IEEE Micro*, 19(1):50–59, January 1999.

[31] A. C. Snoeren and B. Raghavan. Decoupling Policy from Mechanism in Internet Routing. In *Proceedings of the Second Workshop on Hot Topics in Networks*, November 2003.

[32] J. Stewart. BGP4: Interdomain Routing in the Internet. Addison-Wesley, 1998.

[33] I. Stoica, D. Adkins, S. Zhuang, S. Shenker, and S. Surana. Internet Indirection Infrastructure. In *Proceedings of ACM SIGCOMM*, August 2002.

[34] L. Subramanian, M. Caesar, C. Ee, M. Handley, Z. Mao, S. Shenker, and I. Stoica. HLP: A Next-Generation Interdomain Routing Protocol. In *Proceedings of ACM SIGCOMM*, August 2005.

[35] B. Van Schewick. Towards an Economic Framework for Network Neutrality Regulation. In *Proceedings of the Telecommunications Policy Research Conference*, September 2005.

[36] M. Walfish, H. Balakrishnan, and S. Shenker. Untangling the Web from DNS. In *Proceedings of the USENIX/ACM Symposium on Networked Systems Design and Implementation*, March 2004.

[37] T. Wright. EU Tries to Unblock Internet Impasse. *The New York Times*, 30 September 2005.

Geoff Goodell is a recent PhD graduate of Harvard University. His interests include networking, security, Internet governance, economics, etc. Geoff currently works on Wall Street.

Mema Roussopoulos is an Assistant Professor of Computer Science on the Gordon McKay Endowment at Harvard University. Her interests are in the areas of distributed systems, networking, mobile computing, and digital preservation.

Scott Bradner is the University Technology Security Officer in the Harvard University Office of the Provost. He helps the University community deal with technology-related privacy and security issues.