Building a Coreless Internet without Ripping out the Core

Geoffrey Goodell
goodell@eecs.harvard.edu
Harvard University

Scott Bradner
sob@harvard.edu
Harvard University

Mema Roussopoulos
mema@eecs.harvard.edu
Harvard University

Abstract

Despite an ostensible purpose of connecting networks, the Internet itself has, over the course of the past decade or more, become systemically fragmented. There are many causes of fragmentation, including middleboxes, incomplete peering, and the structure of Internet governance. While fragmentation may be desirable in certain circumstances and for various reasons, it can also be problematic, violating central Internet design principles and rendering routine tasks difficult. We motivate the need for a system designed to facilitate connectivity throughout the Internet, providing the benefits of locality, universal access, and distributed management, while interoperating with the existing infrastructure. Using this system as our context, we explore the somewhat unconventional perspective that the Internet need not have a well-defined core and envision an Internet consisting of a set of loosely-connected fragments, each with its own naming and address space. We explore what we gain and what we lose by taking this "coreless" approach.

I. Motivation

Much of the literature on Internet architecture has argued that access to resources should be a function of *who users are* rather than *how users are connected*, and to enable access, it should be universally possible to uniquely and specifically identify each resource. However, the modern Internet does not allow such an arrangement. The set of resources to which a given user has access has become a function of the location of the user's host within the Internet topology. That is, the network itself acts to restrict access to certain resources to hosts in particular locations. In this sense, we say that the Internet has become *fragmented*.

There are many causes of fragmentation, ranging from accidental (routing failures, misconfigured policies, unreliable network elements) to deliberate (content filtering, network address translation, firewalls, malicious service providers). We are interested primarily in (a) the fragmentation that results from middleboxes (e.g., NATs, firewalls, other content filters) and (b) the potential for inconsistency in naming resulting from multiple DNS roots or addresses that are not globally unique. Above all, we believe that fragmentation is inevitable: the address isolation afforded by NAT devices is commercially precious, and global agreement on Internet governance will only become increasingly difficult as the number of participants grows.

Previous approaches to overcoming fragmentation to facilitate end-to-end connectivity require extensive changes to operating systems (such as the deployment of new protocol stacks) or the explicit participation of ISPs and content providers. Our approach employs the more lightweight techniques used by peer-to-peer overlay networks to address this problem instead.

We propose Blossom, an unstructured, peer-to-peer overlay network of *forwarders* carrying TCP traffic that act as intermediaries between nodes that cannot communicate directly. Blossom does not require changes to the operating system or the protocol stack, and Blossom does not require active participation of ISPs or special configuration of in-band network-layer elements such as routers or middleboxes.

Blossom allows us to study what the world would be like with a *"coreless"* Internet, i.e., an Internet without globally assigned names or addresses. A client using the Blossom overlay can access a remote resource, provided that it can build a tunnel through the network, across fragments, to a remote forwarder that can access that remote resource. Like popular peer-to-peer filesharing networks, Blossom allows end users to participate directly, but Blossom users are sharing their *perspectives* rather than their content.

If we assume that we can build such an overlay network and that it can scale "reasonably," we find a number of interesting benefits to the deployment of such a system as well as potential red herrings. The purpose of this paper is to outline the issues and consider the tradeoffs.

II. Related Work

A plethora of existing studies focus on overcoming fragmentation in the Internet. These include:

- Indirection. In I3 [12], services are registered with the infrastructure. TRIAD [1] uses globally unique, hierarchical names to identify networks; these names are propagated throughout the system via BGP-like advertisements among TRIAD nodes. Blossom does not require registration of services, names of resources need not be globally unique, and names of Blossom forwarders are non-hierarchical.
- Anti-Censorship. Infranet [5] and Tor [4] aim to anonymize communication. While anonymity is not a direct goal of Blossom, the anonymity property provided by such systems could be leveraged to further our goal of location-independent access to network services.

- Decoupling Policy from Mechanism. FARA [2] provides a general framework for describing associations between nodes without requiring a global namespace. Platypus [11] provides a system for enforcing routing policy on the forwarding plane rather than the control plane, relying upon cooperation from intermediary ISPs. Blossom aims to not require such cooperation, at least not on a technical level.
- Interoperating with Middleboxes. UIP [6] and DOA [14] aim to route around middleboxes, providing efficiency and scalability in the process. Unlike Blossom, these systems create new identifiers for the transport-layer endpoints, requiring modification to the protocol stack.
- Non-Universal Namespaces. Semantic-Free Referencing [13] stipulates that resources have globally-unique self-certifying names that can be resolved by clients into semantic-free tags using third-party services that are not universal. 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 Blossom (Section IV.
- Extending NATs. IPNL [7] adds an overlay layer above IPv4 that would be routed by NATs and makes use of Fully Qualified Domain Names as end system identifiers in packets. Like Blossom, IPNL intends to provide end-to-end connectivity across NATs. Unlike Blossom, IPNL allows its routers to remain stateless. However, IPNL is site-centric, requiring special configuration and deployment of "frontdoors" that connect independently managed networks to an established core. Blossom makes no such assumptions, instead requiring only that there exists a forwarder capable of reaching the target network and that that forwarder has the ability to bidirectionally communicate with another forwarder in the Blossom overlay. Also, Blossom does not require any changes to the operating systems of end hosts.
- Embracing Heterogeneity. Plutarch [3] takes the leap of considering network fragmentation as the inevitable result of political or economic forces rather than some technical obstacle to be overcome. The authors convincingly argue that avoiding global management would promote innovation. Like Blossom, Plutarch does not require a well-defined Internet core or global names. Plutarch "contexts" are similar to the "fragments" that we describe. However, like IPNL and unlike Blossom, Plutarch requires these contexts to be well-defined and non-overlapping. Moreover, Plutarch requires special configuration of middleboxes that serve as the boundaries between contexts. Plutarch also resolves names via a peer-to-peer search, which Blossom avoids in favor of reducing overhead and improving connection setup time.

III. How Blossom Works

We give a brief overview of the Blossom architecture including a description of its components and a description of how a client uses the Blossom overlay to access a remote resource.

A. Components

The Blossom system consists of the following components:

- Resources. Resources are simply hosts that offer (possibly legacy) services to which the Blossom overlay enables access.
- Blossom Forwarders. Forwarders are the nodes that make up the peer-to-peer overlay network, working to establish virtual circuits through which TCP streams flow.
- Blossom Clients. The Blossom client consists of two components: (a) a proxy that serves as an intermediary between client applications and the overlay network, and (b) a mechanism for choosing paths and establishing circuits through the overlay network.
- Blossom Directory Servers. The directory servers obtain information about the individual forwarders. Clients contact the directory servers in turn to obtain information necessary to route traffic to the forwarders of interest.

From a high-level perspective, Blossom forms an overlay network for TCP. Applications treat the Blossom client as a generic transport-layer proxy; this proxy may use the SOCKS [8] protocol. The Blossom client receives information about the status of the Blossom network via the directory servers and passes traffic to the network of Blossom forwarders, which ultimately complete the connection to the target server.

B. Directory Servers

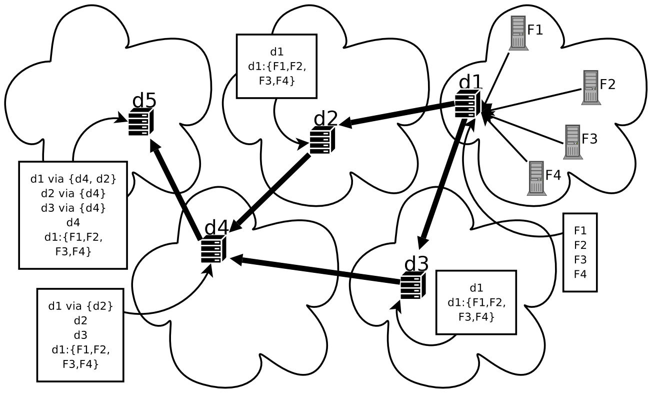

Fig. 1. Advertising Blossom Routers. Blossom directory servers use a *path-vector* algorithm to propagate contact information for Blossom forwarders. Black lines indicate the path taken by an advertisement initiated by the directory server labeled $d1$.

Blossom directory servers publish four different kinds of entries:

- Forwarder Descriptor. Blossom directory servers provide *descriptors* that can be used by the Blossom client to establish circuits through the forwarding network. Descriptors are self-signed statements published by forwarders that contain contact information, including IP address, port, and RSA key, as well as salient information about the capabilities of the forwarder, including exit policy and bandwidth measurements.

- FORWARDER PATH. Suppose that a Blossom forwarder publishes its descriptor to some particular directory. The Blossom architecture allows forwarders to publish their descriptors in directories in locations from which those forwarders are not directly accessible. If the forwarder is not directly accessible by nodes that receive descriptors from this directory, then the forwarder must provide instructions by which some client can reach it. These instructions appear in the form of a *path*, listing a particular sequence of nodes to which to connect to establish a circuit including the target forwarder. If, in the context of Figure 1, F_1 had published to d_5 directly, then there would be a forwarder path entry for F_1 describing how to get to F_1 from the vicinity of d_5.
- DIRECTORY TABLE. Directory servers publish a list of other directory servers in the system, as accrued over time through routing advertisements. Entries for directory servers that are directly reachable are trivial, containing only the name of the server. Other entries include a path through the set of directory servers via which the remote directory service may be reached. The first four entries in the box corresponding to d_5 in Figure 1 represent directory table entries.
- DESCRIPTOR MAP. Not all Blossom directories publish descriptors for all Blossom forwarders; however, given the name of a particular Blossom forwarder, every Blossom directory must know how to find the descriptor for that forwarder. Each directory server publishes an entry corresponding to each foreign directory server, with a list of Blossom forwarders whose descriptors are published at that directory server. The last entry in the box corresponding to d_5 in Figure 1 represents a descriptor map entry.

The directory servers propagate reachability information about individual entries (both forwarders and directory servers) in their respective databases 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 its data will be routed to the requested forwarder. Figure 1 illustrates the process in which route information is propagated through the system. Entries are propagated using a BGP-like path-vector protocol, which includes a simple route selection protocol run at each of the directory servers.

C. Accessing Resources

Suppose that the forwarders have organized themselves into an overlay that can route TCP traffic. We stipulate that each forwarder independently generate a self-certifying identifier, and forwarders throughout the system refer to other forwarders using these identifiers. As long as the size of the identifier is sufficiently large and the sources of randomness are sufficiently effective, the chance of a namespace collision among these identifiers within the system will be negligible.

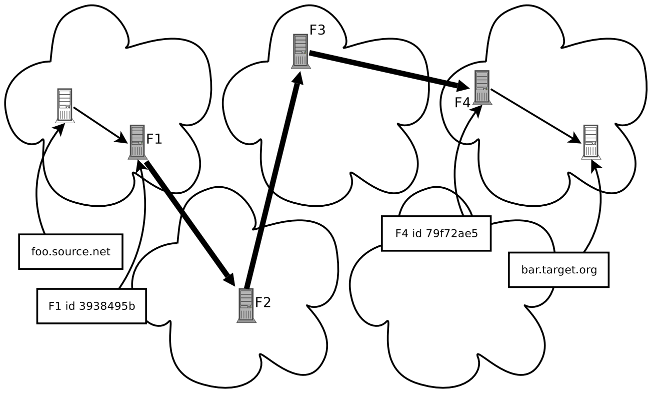

Fig. 2. ACCESSING A RESOURCE. *The source establishes a connection to* bar.target.org.79f7ł2ae5.exit. *DNS requests and TCP sessions are both tunneled through the infrastructure.*

Figure 2 depicts how Blossom enables an Internet host to access resources outside its local fragment. Suppose that the source (labeled foo.source.net) wants to communicate with a host known to forwarder F_4 as bar.target.org. Suppose that the source knows how to talk to F_1, and that the self-chosen ID of F_4 is 79f72ae5.[1] Then, the source will tell F_1 to open a TCP session to bar.target.org.79f72ae5.exit on its behalf. The control plane provides F_1 with routing information indicating that F_2 is the next hop en route to F_4, so F_1 knows how to forward packets through the overlay to F_4. Next, F_1 forwards the request for bar.target.org through the overlay to F_4, who uses DNS to resolves it to an IP address. At this point, F_1 can tunnel the entire TCP session through the overlay to F_4. Note that this involves segmenting the TCP session—the conversation between the source and F_1 will have a different pair of source and destination addresses than the conversation between F_4 and the target resource. This means that Blossom will not work with end-to-end address-based security systems such as IPSec; we describe the policy implications in more detail in the following section.

Observe that the combined name bar.target.org-.79f72ae5.exit is globally unique, but the name was not apportioned by any authority of global scope. Also, there is no requirement that each resource be associated with exactly one forwarder; multiple forwarders may be able to reach the same resource, possibly using different names.

IV. WHAT WE GAIN

A. Consequences of our Design Choices

We submit that the amorphous nature of the Internet facilitates its growth, that fragmentation is part of this amorphous nature, and that designing an architecture that acknowledges fragmentation as a fundamental characteristic of the underlying network provides a number of benefits. These include:

[1] We chose four bytes to create an illustrative example; actual IDs would be longer. Also, in practice we use human-readable names, mapped to self-certifying IDs by a third party.

Locality. The existing Internet paradigm intends for there to exist a global namespace in which centralized authorities allocate names hierarchically and uniquely. Conversely, in the real world, the meaning of a name is dependent upon its context (unless there is a lot of money involved). That is, there can exist two companies named *Olympus*, each selling a different service (e.g., a global airline service and a pizza service in Boston). Some trademarks like “Xerox” prevent others from re-using the name but only because lawyers have determined it reasonable to uphold the validity and universality of the particular trademark; for many smaller organizations, name re-use is allowed and unchallenged. Why assume that all names must be unique just because a few organizations insist that their names be unique everywhere? We would rather not take a position on this; quite the contrary, we believe that technology should not get in the way of reasonable legal process. A technology that requires global uniqueness takes the courts (and thus society) out of namespace decisions.

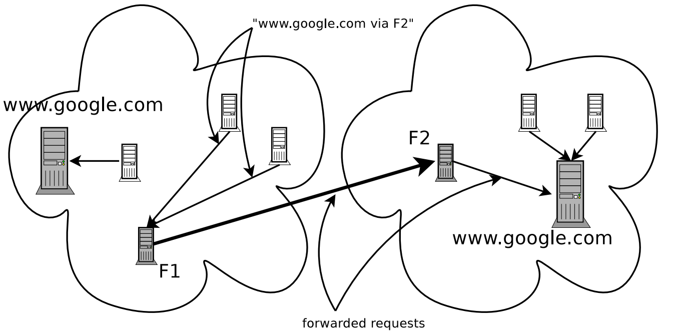

Fig. 3. LOCALITY. *Multiple services with the same name may coexist within different local namespaces.* (Meaningful names within a local space.)

We believe that a system that facilitates communication across network fragments should also allow for the development of distinct local namespaces, in which names have local meaning, while also allowing access to objects in other namespaces that happen to bear the same name. Thus, we abandon global uniqueness of names in favor of flexibility. For example, in Figure 3, there are two resources named `www.google.com` in the left and right fragments. The service provided by each resource should not be required to be the same. Instead, a host in the left fragment should be able to access the `www.google.com` resource in the right fragment via the Blossom forwarder $F2$. This could potentially afford businesses the opportunity to protect their trademarks, avert some Internet namespace arbitrage, and generally lead to relaxation of an unnatural constraint on naming.

Plausibly Universal Access. Sometimes, communication between networks is compromised for architectural convenience rather than policy reasons. In such cases, we would like to provide an architecture that facilitates the use of intermediaries to allow communication between entities that cannot communicate directly. In Figure 4, hosts on the right-hand side requesting resources located in the private network on the left-hand side should be able to access the resources, provided forwarders F_1 and F_2 can communicate and maintain a persistent connection to each other.

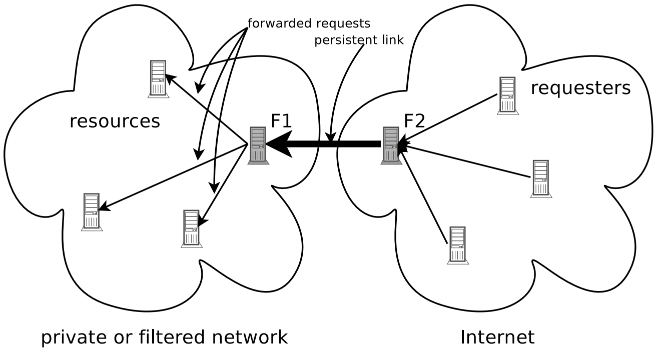

Fig. 4. PLAUSIBLY UNIVERSAL ACCESS. *If two hosts can both access forwarders within the same forwarding infrastructure, then those two hosts can use the infrastructure to communicate.* (Circumvent technical barriers.)

Distributed Management. Contrary to popular belief, the Internet is not entirely a distributed network. While its management is somewhat decentralized, some key aspects of its structure and governance are hierarchical. Autonomous systems engage in peering relationships in a manner that promotes the set of “tiers” that characterize the organization of Internet service providers today. Both the addresses and the names used to identify resources are allocated by a collection of governance organizations, arranged hierarchically. Such an arrangement is contrary to the underlying relationships among organizations interested in using the Internet to communicate. We would like to provide a means by which the Internet can grow without requiring the consent of far-removed third parties. In Figure 5, a new network fragment on the left is set up to deploy a Blossom forwarder called F. Adding this fragment to the existing Blossom infrastructure requires only that a persistent connection be established with an existing Blossom forwarder. In this case, forwarder F_1 might be chosen initially, but if F_3 becomes reachable or more convenient later, then forwarder F can set up a persistent connection with F_3 instead.

Deployability. Any complex system of sufficiently large scale that cannot be deployed incrementally will never amass enough interest to overcome the economic hurdles to deployment. Our system must provide substantial benefit even if its rate of adoption is quite limited. So, we require that our system can coexist with existing Internet infrastructure. In particular, both clients and servers should be able to easily use both our system and the underlying Internet architecture. To this end, we have developed a prototype that leverages the Tor overlay network [4] and is immediately usable by any client with no changes required to the operating system running on the host. An interesting consequence of running this prototype is that we can detect subtle differences in the service provided by some resources (such as Google), depending upon our choice of last-hop forwarder.

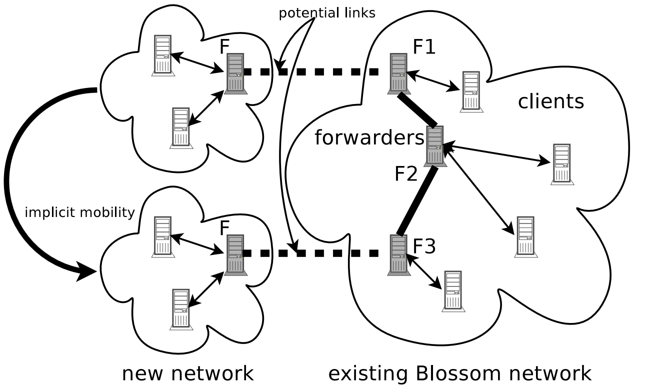

Fig. 5. DISTRIBUTED MANAGEMENT. *Adding a network and its abundance of resources to the system need not require specific allocation of names, addresses, or routing from centralized authorities.*

A "Coreless" Internet. Many previous approaches to providing end-to-end connectivity across middleboxes assume a core to which all forwarders are attached [6], [7], [14] or recognize that fragments can have their own address space allocation, but assume a globally unique DNS-like name for resources [1]. Like Plutarch [3], Blossom achieves truly separate naming and addressing in different fragments. However, unlike Plutarch, Blossom does not require the boundaries between fragments to be well-defined.

B. Remarks

The Blossom design achieves its seemingly conflicting goals of locality and universal access at the expense of universal naming. Indeed, in the core of the Internet today, names used to identify resources are universal: they depend only upon the resource and are not defined by the name, physical location, or logical location of the entity requesting the resource. We argue that universal naming is not indispensable, and we believe that by relaxing this constraint we can achieve a considerably more flexible network.

The benefit of Blossom is its separation of access policy from network-layer mechanisms. Consider an organization whose core IT staff makes network policy decisions regarding external access to internal resources or internal access to external resources. Without Blossom, specific managers and groups have three choices:

- Convince the core to make specific provisions for access policy changes affecting services in their area,
- Convince the core to work with them to deploy special infrastructure allowing partial delegation of the management of network access privileges, creating added complexity, or
- Break network access mechanisms (e.g., punch holes in firewalls, use additional ISPs to provide network uplinks to the core network), potentially undermining the goals of the core administrators.

Blossom provides organizations the opportunity to delegate responsibility for network access policy to a broader set of managers capable of making policy decisions. By providing a mechanism that can be managed locally but verified centrally, we alleviate some technical barriers to defining policy. Ultimately, technology should be used to facilitate management decisions, not encumber them. Individual managers can make executive decisions about whether allowing access to a particular resource is consistent with the stated objectives of the organization or not. We seek not to answer the question of whether such empowerment is appropriate in each individual case, but only to ensure that the requirements of particular network technologies do not prevent such questions from being asked.

Many enterprises use end-to-end authentication for some services, but there are a number of popular services that rely upon the assumption that the only hosts that have access to the service are physically on the same LAN or have particular network-layer addresses. For example, the market for secure fileservers is small. We suspect that this means that most distributed filesystems used by most businesses base their security upon assumptions about how clients are connected. We do not seek to create new risks for organizations that rely upon firewalls; we seek to provide a means by which firewalls need not unnecessarily constrain access to services. This is a problem that bridges the gap between IT and management, and our solution must respect the interests of both sides:

- For secure services, we simply configure our Blossom forwarder to exit to the corresponding IP address(es) and port(s).
- For services that are insecure because they potentially send or receive sensitive data in the clear, we may derive benefit from running the Blossom forwarder on the same machine with the service. Using onion routing, we get an end-to-end encrypted tunnel from the client to the machine with the service at no additional cost.
- For services that are insecure because they do not provide authentication, we must provide the authentication on the side with the Blossom exit forwarder. We can have the Blossom forwarder exit to a particular port (on an authentication server, which may be the Blossom forwarder itself) that provides secure authentication (e.g. via SSH or SSL), and we use the resulting secure channel to open a secure tunnel from the client to a SOCKS proxy running on the authentication server. The client can then use SOCKS to communicate with arbitrary TCP services in the network containing the Blossom forwarder.

V. WHAT WE LOSE

To achieve our various goals, we make a number of tradeoffs, all of which have associated costs. Among them:

New Namespace Constraints. Do we really need globally unique identifiers across all components that want to talk with the outside world, or merely a way to uniquely identify resources?

New Scalability Constraints. By giving up a global unique namespace for resources, we need some way to uniquely identify a resource. For this reason, we require forwarders to generate unique, self-certifying identifiers and concatenate these identifiers with the local names of resources to uniquely identify the resources, and these identifiers of forwarders must be propagated with directory entries through the Blossom overlay. Also, there seems to be an inherent tradeoff between the ratio of forwarders to directory servers and the frequency of updates for particular directory entries.

Regarding "reasonable" scalability, consider that there are serious limits to the theoretical scalability of BGP4 [10], the de facto protocol for interdomain routing, and nonetheless this system is quite functional and useful on a global scale. The propagation of routing updates through Blossom follows a similar pattern. Note also that one clear alternative to propagating routing updates is performing queries (and possibly caching results); this approach introduces a different set of scalability concerns and also complicates connection setup.

New Discovery Constraints. With Blossom, we will need a way to find the forwarder that can access the remote resource that we want. Ultimately, we need the global name of the forwarder plus the local name of the resource to be able to access the resource. Should we build a global distributed directory service? This sounds a lot like DNS, even if, unlike DNS, it is not explicitly hierarchical.

VI. Some Neat Open Questions

Ultimately, the success of Blossom depends upon the answers to the following questions, each of which could lead to important research.

- Do we really need a universally accepted DNS? (Consider the recent WSIS disputes over control of DNS [15].) If we have several, do we need to provide another directory service to bridge them?
- How can we perform a Google-type search across fragments with disparate name services? Furthermore, how might the locality feature afforded by Blossom be used to improve searches?
- How can a system like Blossom co-exist peacefully with reasonable causes of fragmentation? Sometimes, walls are erected for good reasons, and Blossom helps people get around these walls. What can we do about this?
- How can a system like Blossom co-exist peacefully (or perhaps, quietly) with unreasonable causes of fragmentation, such as oppressive governments? Might it be possible, if we assume the existence of practical steganography, to build a system that could effectively provide access to blocked resources even under such circumstances?
- How can a system like Blossom co-exist peacefully with a reasonable desire for placing a wall but unreasonable execution of the wall placement? That is, for organizations wanting to protect a small set of services dependent on network-layer authentication to provide access, is it possible to use Blossom to achieve centrally-ordained security policy that is successfully executed in a decentralized manner?

VII. Conclusion

Blossom provides a convenient means of bridging to external and private networks as a means of providing end-to-end connectivity to pairs of Internet nodes that are not directly connected to each other. However, Blossom is not just a means of sustaining some recondite network design principle; it has practical uses in isolating policy decisions from in-band network technology decisions. We have yet to determine whether such a system could scale to Internet-sizes of the future, and we have yet to explore whether multiple large-scale independent Blossom networks could reasonably coexist. Nonetheless, 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.

VIII. Acknowledgements

We would like to thank the following individuals whose insightful comments have greatly benefited our project: H.T. Kung, David Parkes, Matt Welsh, Margo Seltzer, Roger Dingledine, Nick Mathewson, and Kevin Lai.

References

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

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

[3] 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.

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

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

[6] B. Ford. Unmanaged Internet Protocol. In *Proceedings of the Second Workshop on Hot Topics in Networks*, 2003.

[7] P. Francis and R. Gummadi. IPNL: A NAT-extended internet architecture. In *Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 69–80, 2001.

[8] M. Leech, M. Ganis, Y. Lee, R. Kuris, D. Koblas, and L. Jones. SOCKS Protocol Version 5. Internet Engineering Task Force: RFC 1928, 1996.

[9] 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*, 2001.

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

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

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

[13] 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*, 2004.

[14] M. Walfish, J. Stribling, M. Krohn, H. Balakrishnan, R. Morris, and S. Shenker. Middleboxes no longer considered harmful. OSDI, 2004.

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