How to Lease the Internet in Your Spare Time*

Nick Feamster
Georgia Tech
feamster@cc.gatech.edu

Lixin Gao
University of Massachusetts
lgao@ecs.umass.edu

Jennifer Rexford
Princeton University
jrex@cs.princeton.edu

ABSTRACT

Today's Internet Service Providers (ISPs) serve two roles: managing their network infrastructure and providing (arguably limited) services to end users. We argue that coupling these roles impedes the deployment of new protocols and architectures. Instead, the future Internet should support two separate entities: infrastructure providers (who manage the physical infrastructure) and service providers (who deploy network protocols and offer end-to-end services). We present a high-level design for Cabo, an architecture that enables this separation, and we describe challenges associated with realizing this architecture.

1. Introduction

The Internet is relatively resistant to fundamental change. The last fifteen years have offered countless "false starts" in the deployment of new services. For example, differentiated services, IP multicast, and secure routing protocols have not seen wide-scale deployment, despite offering tangible value and making significant headway through the protocol standardization process. A major impediment to deploying these services is the need for *coordination*: an Internet service provider (ISP) that deploys the service garners little benefit until other domains follow suit [18]. For example, an ISP that deploys a secure routing protocol like S-BGP [13] incurs substantial cost but still is not protected from bogus route announcements unless *other* ISPs also deploy S-BGP.

ISPs are under immense pressure to offer "value added" services, in response to both customer demands and the increasing commoditization of Internet connectivity. Building a network that has global reach requires either "building it yourself" or depending on other ISPs for connectivity. ISPs naturally adopt the latter approach to contain cost. Unfortunately, because a single ISP rarely has purview over an entire end-to-end path, new services either have been deployed only in small islands or have languished entirely. Some ISPs, hard-pressed to offer profitable services to end users, are driven to extortionary measures such as degrading service for some, while providing "better service" (though not better *end-to-end* service) for others, as evidenced by the ongoing "net neutrality" debate [7, 24].

Researchers are also under pressure to justify their work in the context of a federated network by explaining how new protocols could be deployed one network at a time, but emphasizing incremental deployability does not necessarily lead to the best architecture. In fact, focusing on incremental deployment may lead to solutions where each step along the path makes sense, but the end state is wrong. Rather, we argue that substantive improvements to the Internet architecture may require fundamental change that is *not* incrementally deployable. Unfortunately, in the context of today's Internet, ideas that are not incrementally deployable are relegated to the library of paper designs that are either never seen again, or, in rare cases, dusted off as "band aid" fixes only when crisis is imminent (as with IPv6 in the face of address depletion in IPv4).

We argue that decoupling *infrastructure providers* (who deploy and maintain network equipment) from *service providers* (who deploy network protocols and offer end-to-end services)[1] is the key to breaking this stalemate. We propose Cabo ("Concurrent Architectures are Better than One"), which exploits virtualization to allow a service provider to simultaneously run multiple end-to-end services over equipment owned by different infrastructure providers. Cabo extends network virtualization beyond its current use for supporting shared experimental facilities, such as PlanetLab [5] and GENI [10]. Rather than simply serving as an evaluation platform for selecting a single "winning" architecture, *support for virtual networks itself should be the architecture*. Cabo's design adopts the *pluralist* philosophy [4], which advocates a flexible and extensible system that supports multiple simultaneous network architectures.

Separating infrastructure providers from service providers has precedence in other industries, where unbundling the value chain has led to better service or lower cost for end customers. For example, the airline industry has airports (infrastructure providers), which allocate certain gates (and sometimes even entire terminals) to particular airlines; airlines (service providers) form relationships with multiple such airports. As infrastructure providers, airports amortize fixed costs by providing other services such as experienced personnel for fueling the planes. The airlines themselves also use code sharing and regional subcontracting to provide end-to-end service to passengers at a reasonable cost. The automobile industry is another example: In the past, automobile manufacturers created their own parts, in addition to composing the parts into cars. Today, different companies manufacture parts or build cars, leading to two separate industries with a symbiotic relationship.

Decoupling service providers and infrastructure providers is consistent with the new business models that have resulted

*Apologies to Staniford *et al.* [21].

[1] Throughout the paper, we use the term "service provider" as an organization that composes network services and protocols on top of physical infrastructure, and "*Internet* service provider" to refer to a status quo ISP.

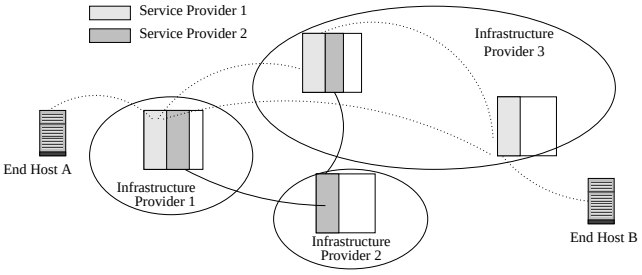

Figure 1: Cabo architecture.

from the commercialization of the Internet. The early rise of "carrier hotels" and exchange points is a perfect example. Carrier hotels reduce the cost of interconnection between ISPs by locating the physical equipment of many different ISPs in the same building. Co-location amortizes the high fixed cost of maintaining a physical footprint (*e.g.*, racks, power supplies, backup generators, switches, fiber, "hands and eyes" support, etc.) by sharing capital and operational expenditure across ISPs. Cabo pushes this amortization to its logical extreme. In the same way that connectivity providers share infrastructure like backup generators, service providers could share the network infrastructure.

Some ISPs are already pushing the trend toward decoupling service from infrastructure in interesting ways. For example, FON, a Spanish ISP, acts as third-party broker for existing 802.11 wireless access points deployed by private households [9]. Rather than deploying physical infrastructure, FON simply bundles Internet access from physical infrastructure deployed by other parties (*e.g.*, wireless access points). Cabo is motivated by a similar philosophy, and it pushes this design to its logical conclusion by allowing service providers to offer a wide range of *end-to-end* services and new network architectures, not just basic Internet access.

Realizing Cabo introduces many challenges. In Cabo, a service provider must coordinate with infrastructure providers to create virtual networks. We must demonstrate that this is easier than coordinating across ISPs to deploy new protocols and services today. Although Cabo allows each virtual network to run its own protocols, we must demonstrate that the underlying network equipment can provide such flexibility at high enough speed. Although Cabo allows each virtual network to run independently, we must demonstrate that managing multiple, simpler virtual networks running in parallel is easier than managing one more complicated network. The rest of the paper discusses these challenges in more detail, starting with an overview of Cabo and its benefits (Section 2), followed by a more detailed treatment of how Cabo works (Section 3). We discuss related work in Section 4 and conclude in Section 5.

2. Concurrent Architectures Better Than One

In this section, we present a high-level overview of Cabo and describe how Cabo enables better network services and more robust networks.

2.1 Cabo Architecture

Cabo separates the notion of conventional ISPs into two distinct entities: infrastructure providers and service providers. An *infrastructure provider* owns and maintains the network equipment (*e.g.*, routers and links) that forms an *infrastructure network*. A *service provider* establishes agreements with one or more infrastructure providers for access to a share of these router and link resources. Cabo facilitates sharing of physical resources by subdividing a physical node (*i.e.*, router) or link into many virtual nodes and virtual links. A *virtual node* controls a subset of the underlying node resources, with guarantees of isolation from other virtual nodes running on the same machine. Similarly, a *virtual link* is formed from a path through the infrastructure network and includes a portion of the resources along the path. Cabo can guarantee bandwidth or delay properties on these links using schedulers that arbitrate access to shared resources, such as CPU, memory, and bandwidth.

A *virtual network* consists of virtual nodes and links that belong to the same service provider. For example, in Figure 1, service provider 1 has a virtual network using physical resources belonging to infrastructure providers 1 and 3 to provide end-to-end services between end hosts A and B; the end hosts may run virtual machines that connect to different virtual networks, possibly run by different service providers. Service providers may install software (*e.g.*, a customized routing protocol) on their virtual components and may even program the hardware (*e.g.*, a customized packet-forwarding algorithm implemented on a network processor or FPGA). A single service provider may have multiple virtual networks tailored to specific services or topologies. or example, one virtual network may run an Interior Gateway Protocol (IGP) like OSPF and conventional longest-prefix match packet forwarding, while another virtual network may support source routing based on flat addresses.

A virtual node might even be subdivided into multiple virtual nodes, and a virtual link itself comprise multiple virtual links. Such "nesting" of virtual components might occur when one service provider offers service to another. For example, one service provider might provide end-to-end connectivity (akin to an ISP today) and sell that connectivity to another service provider that offers some other end-to-end service. Also, an infrastructure provider might offer some services beyond the basic support for virtual components. For example, to reduce the number of nodes that other service providers would need to manage, an infrastructure provider might run a virtual network of its own, with virtual links between pairs of its edge routers.

2.2 The Benefits of Cabo

In this subsection, we present examples that illustrate the benefits of Cabo. First, Cabo allows service providers to offer "value added" services by enabling end-to-end deployments and lowering the barrier to establishing a network point-of-presence. Second, Cabo simplifies network management by "outsourcing" the responsibility for the physical devices to the infrastructure providers and allowing a service provider to run several simple virtual networks in parallel.

2.2.1 *Better network services*

End-to-end network services. Some players in the "net

neutrality" debate have advocated a *tiered Internet*, where Internet service providers provide "better" service to edge networks and content providers (*e.g.*, Google) who pay more money directly to those ISPs [7, 24]. This "enhanced service" is disingenuous: a tiered Internet cannot inherently provide better service, since no single ISP controls any given end-to-end path (*e.g.*, between a home user and Google). Cabo can help reverse these troubling trends by giving a service provider the opportunity to add real value by exposing control over *end-to-end* paths. In Cabo, infrastructure providers can achieve a competitive advantage by running more efficient and robust networks, and service providers differentiate themselves by running different end-to-end services on a common physical infrastructure.

Customized protocols. Cabo allows service providers to build virtual networks with dramatically different characteristics on top of the same physical infrastructure. For example, one service provider might deploy a network based on a secure routing protocol that provides strong guarantees at the cost of complete reachability, while another offers global reachability with less security. Similarly, one service provider might perform conventional IP routing and forwarding, while another permits end hosts to perform source routing [27] on a relatively small virtual network, consisting of virtual links that span multiple hops in the infrastructure. Deploying source routing today is immensely difficult, since most ISPs disable the feature; in Cabo, a service provider could decide to offer source routing on its virtual network without having to coordinate with other ISPs.

Co-location for expanded network presence. In today's Internet, an organization that needs a global footprint must deploy physical infrastructure in a wide variety of locations; each router deployed in a new remote facility incurs a relatively high fixed cost. Today, these organizations can contract with an ISP that offers a Virtual Private Network (VPN) service, though finding a single ISP with facilities at every location may be difficult. In contrast, Cabo allows that enterprise (or its service provider) to instantiate virtual nodes and links on equipment managed by an infrastructure provider in the region. This allows the organization to runs its own virtual network or contract with a single service provider for a VPN service, without incurring the costs of deploying and managing additional equipment.

2.2.2 *More robust management and operations*

Testing and deploying new protocols. Today's router software is typically evaluated in a test lab before deployment. Large lab configurations that mimic a production network are expensive, and limiting tests to simple topologies and traffic patterns that may not give operators an accurate view of how the new software would perform "in the wild". In Cabo, new router software (including new experimental services) could be evaluated on a separate virtual network on the same underlying infrastructure; this virtual network could initially carry only test traffic or support users willing to serve as early adopters. Also, migrating a network from one protocol to another can be painstaking [11]. In Cabo, a new protocol could be deployed in its own virtual network, followed later by a cut-over of the data traffic from the old virtual network to the new one.

Protection against misconfiguration. Cabo provides isolation between different network components and services, which can provide protection against misconfigurations and bugs. Network protocols are commonly misconfigured [16] and are subject to implementation bugs. Adding a new service, provisioning a new customer, or rebalancing traffic each requires an operator either to invoke certain configuration commands or to install new software; these actions may cause instability or temporary service disruptions. Cabo allows services that might interact to be compartmentalized into different virtual networks, thereby preventing configuration errors or software bugs related to one network service from interfering with others.

Accountability at every layer. In the current Internet, a single ISP manages its network from the physical infrastructure, all the way up to applications, but that ISP typically does not have purview over an entire end-to-end path. When performance or security problems arise, the ISP must initiate the arduous process of locating the source the fault (often a different ISP) and coordinating to diagnose and fix the problem. This process is inherently difficult because both monitoring and mitigation require coordination across one or more administrative boundaries. Cabo, on the other hand, allows each entity to have complete, end-to-end control over the layer it is managing. For example, when the virtual components do not behave as expected, the service provider has direct recourse (and a direct business relationship) with the infrastructure provider managing the equipment.

3. How Cabo Works

Cabo must support (1) simultaneous operation of multiple virtual networks on top of a single physical infrastructure; (2) handling requests from service providers to create and instantiate these virtual networks; and (3) enabling infrastructure providers to discover and manage the physical infrastructure. This section describes these tasks (which Figure 2 loosely illustrates) in succession, working from the top down: We start by describing the support that Cabo must provide once the virtual network is instantiated and end with the support that must be available to infrastructure providers.

3.1 Supporting Concurrent Networks

Cabo must allow several virtual networks to share the same physical infrastructure, and it must guarantee that these simultaneously running networks do not interfere with one another's operation. Allowing multiple virtual networks to share the same physical infrastructure requires the ability to *virtualize* the network components (*i.e.*, nodes and links). Virtualization implies provides both *logical* isolation of distinct virtual networks (*e.g.*, to separate namespaces) and *resource* isolation (to ensure that virtual networks cannot interfere with one another). We first discuss the requirements for virtualizing nodes (*i.e.*, routers) and links; we then describe the challenges for providing resource guarantees for both the virtual nodes and links.

Virtual routers. Router virtualization—providing multiple

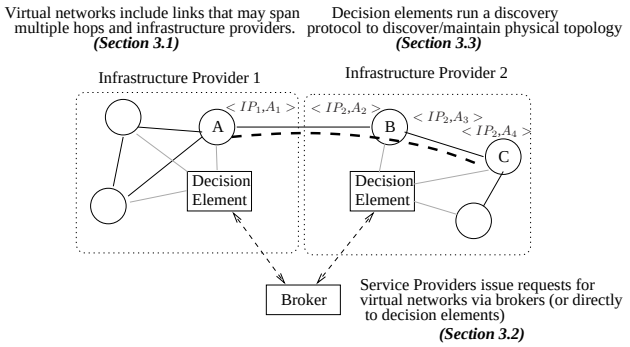

Figure 2: An infrastructure provider's decision elements manage physical infrastructure; service providers can then request virtual links, named by the physical interfaces. The thick dotted line shows a virtual link that traverses four physical interfaces.

virtual routers on a single physical router—is fundamental to Cabo because it allows multiple service providers to share the same set of physical routers. Various router vendors (*e.g.*, Cisco, Avici) provide virtual routers to simplify network design at network points-of-presence (PoPs), reduce capital expenditure, and lower the barriers to co-location [17]. (Some router vendors have gone so far as to claim that router virtualization can increase network reliability by reducing the number of interconnections within a PoP.) Indeed, router virtualization has many possible uses and has gained traction with major router vendors; Cabo can use this developing technology as a way to construct virtual nodes. To better support new protocols and forwarding algorithms, Cabo could also make use of programmable routers [14, 25], which can be simultaneously used by multiple parties.

Virtual links. In addition to virtualizing nodes, Cabo must virtualize *links* between any pair of virtual nodes. The ability to create these types of tunnels already exists in many forms, at the network layer and below (*e.g.*, at layer 2). Virtual links in Cabo might look much like the tunnels that exist in today's VPNs, which provide support for tunneling and encapsulation. Cabo provides considerably more functionality to a service provider than a VPN, which simply provide connectivity between edge sites over a single ISP backbone. First, Cabo gives service providers direct control over the protocols and services that run on the virtual nodes. Second, Cabo allows a service provider to instantiate a virtual network on infrastructure that is owned by multiple infrastructure providers.

Scheduling. To provide the semblance of a dedicated network to each service provider, Cabo must schedule access to physical resources, such as CPU, memory, disk, and link bandwidth. In particular, each service provider requires guarantees for link characteristics, ranging from best-effort service to the same loss, delay, and queuing qualities as a physical link with a pre-determined allocated bandwidth. Cabo can exploit existing support for CPU and link scheduling to guarantee resources for each virtual component. Each physical interface can maintain a separate queue for each virtual interface with a scheduler that services these queues to achieve the performance guaranteed. We may be able to exploit previous work on link-level quality of service at the network layer in the design of the scheduler [8, 28].

3.2 Instantiating Virtual Networks

Cabo must enable a service provider to request virtual network components from infrastructure providers and an infrastructure provider to fulfill a service provider's requests (if possible). This process requires an interface for service providers to make these requests, a "signaling band" over which these requests can be communicated, a mechanism that allows an infrastructure provider to calculate whether the request can be fulfilled, and algorithms for determining how to embed a virtual network in the physical infrastructure in a way that makes efficient use of the physical resources.

Interface and bootstrapping. Cabo must provide an interface for service providers, who will compose physical infrastructure from one or more infrastructure providers to construct a virtual network. Ideally, the service providers would be able to specify their requirements in terms of specific properties, such as the location of virtual nodes and the bandwidth and delay of virtual links), rather than identify specific routers and links. Cabo also requires bootstrapping capabilities that allow service providers to load software on the the virtual nodes once they have access to them. For example, an infrastructure provider could run a virtual network that provides basic reachability and might also offer services for loading software onto the nodes, collecting measurement data, and reserving node and link resources.

Signaling. To issue requests to infrastructure providers, service providers must already have network connectivity to infrastructure providers, which introduces a circularity where network connectivity (to the infrastructure providers) is required to obtain network connectivity (for the virtual network). To resolve this circularity, Cabo could ultimately provide complete connectivity (*i.e.*, the ability for any node to reach any other node) through the evolution of the service-provider market. This economy is similar to the scenario in today's Internet, which we think of providing complete connectivity even though the network does not provide any such guarantee. Until Cabo provides such connectivity, a service provider might use other means to communicate with an infrastructure provider, such as today's Internet or the phone network. This mechanism resembles the way the Internet evolved where, initially, requests for Internet connectivity were made via phone, fax, or postal mail; later, once global Internet reachability was available, these requests could be made via the Internet itself (*e.g.*, via Web sites).

Accounting and admission control. Guaranteeing quality-of-service not only requires scheduling competing traffic flows from concurrently running virtual networks (as described above); it also requires that the infrastructure does not "overbook" resources to service providers. Accordingly, the network must maintain an accurate accounting reserved resources and exercise admission control to ensure that the bandwidth allocated to the collection of virtual networks does not exceed the physical capacity of the network. This problem is similar to the accounting problems faced by traditional admission control protocols (*e.g.*, RSVP [28]), but

Cabo must perform admission control on virtual *networks*, not simply on individual links or paths.

Virtual network embedding. Because a virtual link may span multiple physical hops, there may be many possible mappings for any given virtual network, especially when multiple infrastructure providers offer virtual components. Determining how to satisfy a service provider's request for a physical mapping, while making the most efficient use of the available physical resources, is important for maximizing the number of virtual networks that can share the physical infrastructure. Thus, Cabo must be able to compute such a "network embedding", but this problem is NP-hard [19]. The embedding problem becomes more complex when the resource requirements for virtual networks may change over time, or when these requests arrive dynamically. Fortunately, testbed designers and researchers have devised ways to compute efficient network embeddings, and we believe that Cabo can leverage these techniques [19, 30].

3.3 Discovering Physical Infrastructure

Before infrastructure providers can determine how to allocate physical resources to request for virtual networks, they must be able to determine the physical topology (*i.e.*, physical nodes, links, and their interconnections). The infrastructure must also provide support for notifying virtual networks about failures in the physical infrastructure.

Topology discovery. An infrastructure provider may run a discovery plane, similar to that outlined in the 4D management architecture [12]. Physical nodes could flood their identities to a decision element, and neighboring nodes could re-forward these identifiers, appending unique identifiers to the message to enable the decision elements to construct a reverse path back to the new node. The operators of the physical infrastructure must be able to reach the decision elements (*i.e.*, to instantiate virtual components and to provide an interface to service providers), either over these same network paths or via existing out-of-band mechanisms (*e.g.*, via today's IP-layer connectivity). Two adjacent infrastructure providers must be able to establish links *between their networks*. The discovery of these links involves both the *naming* of endpoints and *dissemination* of reachability information. Every physical endpoint must be uniquely named; to achieve this global uniqueness, interfaces could be named with an <infrastructure provider, local ID>, as proposed in previous work [26]. Dissemination could be achieved via flooding across inter-provider links or between the two infrastructure providers' decision elements.

Notification of topology changes. Cabo must notify virtual network components when underlying physical components have failed. The underlying physical equipment must notify each affected virtual component about the failure. If an *interface* fails, the incident physical node must notify each of the virtual nodes that have a virtual interface running on the affected physical interface. To detect physical *link* failures, virtual nodes can run a simple heartbeat protocol, as is common in today's routing protocols. Alternatively, an infrastructure provider could run an underlying link-layer protocol that automatically detects link failures (similar to how SONET can detect "Loss of Signal") and notify the affected virtual components. A physical *node* failure is similar to the failure of all of the incident physical links, except that the software running on the physical node may also need to be reinstalled. Cabo can rely on each infrastructure provider's decision elements to detect physical *node* failures and reinstall the service-provider software for each virtual node when the physical node recovers.

4. Related Work

Cabo is the first network architecture that separates service providers from infrastructure providers, but today's Internet offers several scenarios where Internet connectivity has been reconstituted to create new services. Equinix [1] and Internap [2] allow edge networks to change upstream providers on relatively short timescales, but these services can only control the *first* ISP along the path to the destination; in contrast, Cabo allows an entity to control the entire end-to-end path. Other systems, such as "Routing as a Service" [15], allow hosts to request overlay paths with certain properties. OverQoS [22] provides a mechanism for establishing overlay links with certain loss and delay guarantees; similar mechanisms may prove useful for constructing virtual links in Cabo that traverse multiple physical hops. Content distribution networks [3] and bandwidth brokers [29] also extend basic connectivity by creating paths from source to destination (or content). Cabo also allows third-parties to compose end-to-end paths and services, but does so by making the construction of virtual links a first-order primitive.

Cabo must allow many virtual networks to operate on the same physical infrastructure; some, but not all, of the functionality required by Cabo is provided by today's layer 3 virtual private networks (VPNs) [20]. Rather than building their own physically separate networks (an expensive proposition), many large multi-site enterprises opt to buy VPN service from a large ISP that runs a backbone network. VPNs allow a single ISP to support many virtual networks on a single physical infrastructure. However, these VPNs do not (in and of themselves) provide resource isolation, they cannot span multiple ISPs (and, thus, are not truly end-to-end), and they offer enterprise neither access to the physical routers nor the ability to run customized code on these routers.

Some research infrastructures use virtualization to support multiple experiments at the same time. PlanetLab supports virtualization of network nodes [5], but not complete networks. The proposed GENI facility [10] and our recent work on VINI [6] focus directly on network virtualization and programmability, but these projects focus on support for experiments and do not revisit the roles of service providers and infrastructure owners. In addition, these facilities can rely on the existing Internet to reach the physical nodes, whereas Cabo must grapple with topology discovery and bootstrapping of the physical infrastructure. Still, we hope to use VINI as an environment for evaluating a prototype of Cabo and quantifying the benefits of running multiple virtual networks in parallel.

In supporting programmable routers, Cabo resembles active networks, which allow end users to install code in routers. Previous research on active networking focused

on issues with mobile code (and the resulting language and security issues) and providing control to end users [23]. In contrast, Cabo focuses on providing service providers (rather than users) with their own virtual networks, with a fairly general programming environment on the virtual nodes. In fact, a service provider could run an active-network architecture within one of its virtual networks.

5. Conclusion

This paper has made the case for Cabo, an architecture that catalyzes the deployment of network protocols and services by separating service providers from infrastructure providers. This separation gives service providers the ability to to deploy an end-to-end network protocol or service by "leasing" physical network infrastructure from one or more infrastructure providers and sidesteps the need for coordination among many independently operated ISPs.

Cabo lowers the barrier for deploying new network protocols and services, but how would Cabo itself be deployed? In particular, would the owners of the current Internet infrastructure have the right incentives to grant other service providers access to their equipment? Concerning the technical hurdles, support for virtualization in commercial routers can help enable many of Cabo's functions. Initially, ISPs could begin to offer some of the services that Cabo enables, such as establishing geographical footprints by leasing a virtual router in other ISPs and offering multi-provider VPNs to large enterprises. With regard to incentives, we note that Cabo does not prevent a single commercial entity from acting as both an infrastructure provider and a service provider; thus, although ISPs gain new capabilities from Cabo (as described in Section 2.2), but they do not lose any functions provided by today's Internet.

In the search for a single "right" future Internet architecture, Cabo offers food for thought: perhaps the right future network architecture is not an end state comprised of a collection of addressing, routing, and forwarding paradigms, but rather a platform that allows these functions to evolve as demands on communication networks change. Indeed, the designers of IP aimed for generality, recognizing that they could not predict what networked applications would ultimately run on top of the network. Continual rapid advances in communication technologies and the sheer difficulty of predicting future requirements of the network suggest that the network architecture itself should also be sufficiently general to enable support for network protocols, services, and architectures that we cannot even imagine today.

REFERENCES

[1] Equinix Direct. `http://www.equinix.com/prod_serv/network/ed.htm`.

[2] Internap Route Control Solutions. `http://www.internap.com/solutions/routecontrol/page1980.html`.

[3] Akamai. `http://www.akamai.com/`, 2006.

[4] T. Anderson, L. Peterson, S. Shenker, and J. Turner. Overcoming the Internet impasse through virtualization. *IEEE Computer*, 38(4):34–41, Apr. 2005.

[5] A. Bavier, M. Bowman, D. Culler, B. Chun, S. Karlin, S. Muir, L. Peterson, T. Roscoe, T. Spalink, and M. Wawrzoniak. Operating System Support for Planetary-Scale Network Services. Mar. 2004.

[6] A. Bavier, N. Feamster, M. Huang, L. Peterson, and J. Rexford. In VINI Veritas: Realistic and controlled network experimentation. In *Proc. ACM SIGCOMM*, Sept. 2006.

[7] BusinessWeek. At SBC, It's All About "Scale and Scope". `http://www.businessweek.com/@@n34h*IUQu7KtOwgA/magazine/content/05_45/b3958092.htm`, Oct. 2005.

[8] D. Clark, S. Shenker, and L. Zhang. Supporting real-time applications in an integrated services packet network: Architecture and mechanisms. In *Proc. ACM SIGCOMM*, Aug. 1992.

[9] FON: WiFi everywhere! `http://en.fon.com/`, 2006.

[10] GENI: Global Environment for Network Innovations. `http://www.geni.net/`.

[11] V. Gill and J. Mitchell. AOL Backbone OSPF-ISIS Migration. In *NANOG 29*, Oct. 2003.

[12] A. Greenberg, G. Hjalmtysson, D. A. Maltz, A. Meyers, J. Rexford, G. Xie, H. Yan, J. Zhan, and H. Zhang. A clean slate 4D approach to network control and management. *ACM Computer Communications Review*, Oct. 2005.

[13] S. Kent, C. Lynn, J. Mikkelson, and K. Seo. Secure border gateway protocol (S-BGP) - real world performance and deployment issues. In *Proc. NDSS*, 2000.

[14] E. Kohler, R. Morris, B. Chen, J. Jannotti, and M. F. Kaashoek. The Click modular router. *ACM Transactions on Computer Systems*, 18(3):263–297, Aug. 2000.

[15] K. Lakshminarayanan, I. Stoica, and S. Shenker. Routing as a Service. Technical Report UCB-CS-04-1327, UC Berkeley, 2004.

[16] R. Mahajan, D. Wetherall, and T. Anderson. Understanding BGP Misconfiguration. In *Proc. ACM SIGCOMM*, pages 3–17, Aug. 2002.

[17] D. McPherson et al. Core Network Design and Vendor Prophecies. In *NANOG 25*, June 2003.

[18] S. Ratnasamy, S. Shenker, and S. McCanne. Towards an evolvable Internet architecture. In *Proc. ACM SIGCOMM*, Aug. 2005.

[19] R. Ricci, C. Alfeld, and J. Lepreau. A Solver for the Network Testbed Mapping Problem. *ACM Computer Communications Review*, 33(2):65–81, Apr. 2003.

[20] E. Rosen and Y. Rekhter. *BGP/MPLS VPNs*. Internet Engineering Task Force, Mar. 1999. RFC 2547.

[21] S. Staniford, V. Paxson, and N. Weaver. How to 0wn the Internet in your spare time. In *Proc. 11th USENIX Security Symposium*, Aug. 2002.

[22] L. Subramanian, I. Stoica, H. Balakrishnan, and R. Katz. OverQoS: An overlay based architecture for enhancing Internet QoS. In *Proc. Networked Systems Design and Implementation*, pages 71–84, Mar. 2004.

[23] D. L. Tennenhouse and D. J. Wetherall. Towards an active network architecture. *ACM Computer Communications Review*, 26(2), Apr. 1996.

[24] Towards a two-tier Internet. `http://news.bbc.co.uk/1/hi/technology/4552138.stm`, 2006.

[25] J. Turner. A proposed architecture for the GENI backbone platform. Technical Report WUCSE-2006-14, Washington University in St. Louis, Mar. 2006.

[26] M. Vutukuru, N. Feamster, M. Walfish, H. Balakrishnan, and S. Shenker. Revisiting Internet address: Back to the future! Technical Report MIT-CSAIL-TR-2006-025, MIT, Feb. 2006.

[27] X. Yang. NIRA: A new Internet routing architecture. In *Proc. ACM SIGCOMM Workshop on Future Directions in Network Architecture*, pages 301–312, Aug. 2003.

[28] L. Zhang, S. Deering, D. Estrin, S. Shenker, and D. Zappala. RSVP: A New Resource reSerVation Protocol. *IEEE Network*, 7:8–18, Sept. 1993.

[29] Z.-L. Zhang, Z. Duan, L. Gao, and T. Hou. Decoupling QoS control from core routers: A novel bandwidth broker architecture for scalable support of guaranteed services. In *Proc. ACM SIGCOMM*, Sept. 2000.

[30] Y. Zhu and M. Ammar. Algorithms for Assigning Substrate Network Resources to Virtual Network Components. In *Proc. IEEE INFOCOM*, Mar. 2006.