Overcoming the Internet Impasse through Virtualization

Larry Peterson
Princeton University

Scott Shenker
ICIR Berkeley

Jonathan Turner
Washington University

Abstract: *The current Internet is at an impasse because new architectures cannot be deployed, or even adequately evaluated. This paper urges the community to confront this impasse, and suggests a way virtualization might be used to overcome it. In the process, we discuss the nature of architecture and the debate between purists and pluralists.*

1. INTRODUCTION

The Internet's stunning success has changed the way we work, play, and learn. The Internet architecture, developed over 30 years ago, has proven its worth by the vast array of applications it now supports and the wide variety of network technologies over which it currently runs. Nonetheless, the Internet's increasing ubiquity and centrality has brought with it a number of challenges for which the current architecture is ill-suited.

While there is increasing interest in new architectures to address these challenges (*e.g.*, see [24, 22, 3, 9, 27, 23, 7, 4, 6, 5, 28, 8]), the prospects for significant change in the Internet architecture appear slim. Adopting a new architecture not only requires changes in routers and host software but, given the multiprovider nature of the Internet, it also requires that ISPs jointly agree on any architectural change. The need for consensus is doubly damning; not only is agreement among the many providers hard to reach, it also removes any competitive advantage from architectural innovation. imminent collapse of the Internet, there seems little hope for major architectural changes, where by "architectural change" we mean innovations that alter the basic aspects of the architecture rather than more minor modifications that can be deployed incrementally (*e.g.*, ECN and route flap damping).

Freezing forevermore the current architecture would be bad enough, but in fact the situation is deteriorating. The inability to adapt to new pressures and requirements has led to an increasing number of ad hoc work-arounds, many of which violate the canonical architecture (*e.g.*, middleboxes). While derided by architectural purists, these modifications have (usually) arisen to meet legitimate needs that the architecture itself could not. These architectural *barnacles*—unsightly outcroppings that have affixed themselves to an unmoving architecture— may serve a valuable short-term purpose, but significantly impair the long-term flexibility, reliability, and manageability of the Internet.

The daunting barriers to deployment of new architectures, while discouraging, do not directly hinder research on such architectures. Architectural invention (as cited above) continues without limitations, even if also without hope of adoption. However, live experimentation with new architectures is more problematic. The main avenue for live experimentation (as opposed to simulation or emulation) is through testbeds. We argue, as have others, that traditional testbeds have severe limitations [15]. Given these limitations, our ability to evaluate new architectures is quite limited.

Our goal in writing this paper is to issue a call to action. It is clear to all that the status quo is not acceptable. We (as a community) are unable to deploy, or even evaluate, new architectures. Moreover, we are not moving forward but instead are regressing as barnacles accumulate on the architecture. For too long the Internet research community has lived with the current impasse without directly confronting it; here we call on the community to cease being satisfied with paper designs that have no future, and instead return to its roots of applied architectural research with the intention of once again changing the world.

Overcoming this impasse will not be easy, and will require addressing three separate requirements. First, researchers must be able to easily experiment with new architectures on live traffic. Second, there must be a plausible deployment path where architectural ideas, once validated, can come into practice. Third, the proposed architectural solutions should not be focused on a single narrow problem, but instead should be comprehensive, capable of addressing the broad range of current architectural problems facing the Internet.

We propose to meet these three requirements by constructing a *virtual testbed.* This virtual testbed will support multiple simultaneous architectures running on PlanetLab, serving all of the communication needs of standard clients and servers. Our intent is to dramatically reduce the barrier to entry for new architectural ideas to be evaluated in practice. Further, we argue that this virtual testbed approach provides a clean path for radical new architectures to be unilaterally and globally deployed. Because this approach does not require universal architectural agreement, we think it a much more plausible deployment scenario for radical new designs that systematically tackle the complete set of problems facing the Internet today.

Central to our proposal is the idea of *virtualization*. Virtualization, as used in virtual memory, virtual machines, and elsewhere, is nothing more than a high-level abstraction that hides the underlying implementation details. Virtualization is the core principle in overlays, both allowing nodes to treat an overlay as if it were the native network, and allowing multiple overlays to simultaneously use the same underlying overlay infrastructure. Both aspects of virtualization are crucial to our virtual testbed proposal.

We begin this paper by reviewing the current approaches to experimentation, physical overlays and testbeds. The following sections present our virtual testbed approach, and various deployment strategies. We end with a discussion of whether virtualization is merely a means for evaluating and deploying new architectures, or if it should be an organizing principle for future architectures (*i.e.*, has become an end in itself).

2. PHYSICAL TESTBEDS AND OVERLAYS

Before a proposed architecture can even be considered for deployment, it must be adequately evaluated. While simulation and emulation are valuable tools for understanding new designs, there is no substitute for experimentation with live traffic. Preparing an implementation to deal with the real world forces designers to confront the many unpleasant realities that paper designs frequently avoid, such as multiple providers, legacy networks, anomalous failures and traffic conditions, unexpected and diverse application requirements, and so on. Moreover, the use with live traffic provides a much fuller picture of how an architecture will perform, and strengthens the case for the architecture actually providing the claimed benefit. Physical testbeds and overlays are the two ways in which researchers currently experiment with new architectures. In addition, overlays are also seen as a deployment path. In this section we discuss the limitations of these approaches.

2.1 Physical Testbeds

The traditional platform for live experimentation has been physical testbeds: leased lines connecting a limited set of locations. Testbeds can be roughly categorized as *production*-oriented or *research*-oriented. Production testbeds, such as Internet2, support real traffic from real users, often in large volume and across many sites. As such, they provide valuable information about the operational behavior of an architecture. However, the users of such a production testbed have no choice about whether or not to participate in the testbed and usually do not even realize that their traffic is part of an experiment. They thus expect the performance and reliability to be no worse than the standard Internet. Production testbeds must therefore be extremely conservative in their experimentation, using well-honed implementations of incremental changes.

Research testbeds (such as DETER) do not carry traffic from a wide variety of real users but instead are typically driven by synthetically generated traffic and/or a small collection of intrepid users. This allows them to be much more adventurous, capable of running first-cut implementations of radically new designs. Unfortunately, this lack of real traffic also renders the results much less indicative of real operational viability.

As a result neither kind of testbed—production or research—produces the data needed to adequately evaluate new architectures. It is therefore difficult to make a compelling case for new architectural designs based on a testbed evaluation.

In addition, because they utilize dedicated transmission links, both categories of testbeds involve substantial cost, and so are prohibitively expensive to operate at very large scale. Thus, they are typically of small geographic extent and arise only with substantial funding support. Given their limitations mentioned above, traditional testbeds offer far too little bang for their buck, and clearly cannot lead us into the future.

2.2 Overlays

Overlays have recently gained more widespread use, both as an experimental platform and as a deployment path [17, 2, 20]. They are not limited geographically (users can access them from anywhere) and usage is voluntary (users can decide whether or not to participate in an overlay). Moreover, overlays typically do not involve significant expenditures. Thus, overlays avoid many of the problems plaguing traditional testbeds. However, they do have limitations of their own.

Overlays have, until the advent of Planetlab [16], required substantial effort to create and maintain. As a result, only a very few of the many recent architectural proposals have been tested on overlays. Overlays remain an underutilized tool for architectural experimentation due to this previously high barrier to entry.

This barrier poses less of an obstacle for deployment because presumably serious deployment efforts have more resources at their disposal than most experimental research efforts. However, as a deployment path for radical architectural innovation, standard overlays falter in at least two areas.

First, overlays have largely been seen as a way of deploying narrow fixes to specific problems in the Internet architecture, whether for performance [17], availability [2], denial-of-service [11, 1], content distribution [12], or multicast [19]. Each of these are seen as providing an isolated function, and there has been very little work on how any of these solutions might work together. More importantly, there has been little thought devoted to how a set of such overlays might ultimately come to replace the underlying Internet architecture.

Second, overlays have been architecturally tame. Because the emphasis has been on deployment in today's Internet rather than on architectural innovation leading to tomorrow's Internet, most current overlays typically assume IP (or a close cousin) as the architecture inside the overlay itself (that is, for the inter overlay-node protocol). As such, they have not been the source of dramatic architectural advancement.

Thus, on their current trajectory, overlays are likely to become just a better way of attaching yet another barnacle, rather than an agent of fundamental change. *What is needed is not so much a technical change in how overlays are built, but rather a philosophical change in how they are used.* Our proposal of virtual testbeds, therefore, is less of a technical advancement than a focal point for a new attitude towards overlays.

3. VIRTUAL TESTBED

To address the problems mentioned above, and to therefore provide an attractive platform for experimentation and possible deployment, we propose a *virtual testbed* (VT) approach. Virtual testbeds have two basic components. First, there is an overlay *substrate*, which is a set of dedicated but multiplexed overlay nodes. The multiplexing of overlay nodes, as first advocated in PlanetLab citeplanetlab, allows multiple experiments to be run simultaneously on the same infrastructure. The effort of instantiating and maintaining the overlay is amortized across the many concurrently running experiments, drastically lowering the barrier-to-entry faced by any individual researcher.

Second, there is a general client-proxy mechanism that allows any host to opt-in to a particular experiment (*i.e.*, a specific overlay running on the substrate). This proxy mechanism treats a nearby overlay node as the host's first-hop router, and does so in a way that does not impose any limitations on the experimental architecture (*e.g.*, it does not require that IP addressing is used). It also supports opt-in at a fine granularity (*e.g.*, routing local traffic directly, or determining participation on a per-application basis).

These two features, multiplexed overlay substrate and general proxy, resolve the technical problems overlays had with barrier-to-entry and architectural limitations. To encourage the use of overlays for more radical architectures, we plan to deploy a prototype of this approach on Planetlab. It will be relatively primitive in its original incarnation, which we hope to make available within the next three months. PlanetLab currently includes over 440 nodes that span 207 sites and 25 countries, and peers with nearly 6000 autonomous systems. We estimate that a PlanetLab node is within a LAN-hop of over one million users. The PlanetLab software architecture multiplexes multiple *slices*, each running a different network service, application, or architecture. When running an architecture, each slice can be viewed as consisting of a set of virtual routers connected (by tunnels) into whatever topology the architecture selects.

Most of the technologies we leverage are straightforward, but there are still some issues to explore. For instance, it is challenging to achieve sufficiently high throughput rates on Planetlab nodes. For example, stock PlanetLab nodes are able to forward packets at 60Mbps, and although we expect to achieve 100Mbps rates with modest optimizations, it is clearly not possible for Planetlab nodes to compete with custom hardware. Similarly, the overlay's virtual links cannot compete with dedicated links. Techniques such as those used in OverQoS [18] will allow an overlay to provide better service than a naive tunnel would provide and could be used in cases where performance is crucial.

The proxy technology is at a similar stage of maturity; moderately developed but still in need of work. Our prototype will leverage proxy technology developed in [10, 13]. These proxies interpose themselves on any IP address or port as seen by the legacy client software. Since almost all client applications use name translation as the first step in communication, we can interpose on DNS requests, and either return the true IP address of the server (if these packets are for the normal Internet) or a fake IP address (if the packets are for the virtual testbed). By then interposing on the fake IP addresses, the packets can be forwarded to the nearest VT node (the VT ingress node). The VT is then free to do whatever it wants with the packets, using whatever IP or non-IP protocols are appropriate to service the packet, and tunneling over protocols it hopes to replace. Since it is essential to provide access to legacy servers to gain real users, on the far end of the VT (the VT egress node), the node reconverts the packet back into Internet format for delivery to the server. The egress node behaves as a NAT, manipulating the source address to ensure that reply packets also enter the VT.

Hosting a service within the VT that is visible to non-participating clients is not difficult either. In this case, the VT provides DNS resolution to point the client to a nearby VT representative, in much the same way that CDNs operate. The local representative can then translate the packets into an internal format for delivery to the server and translate the packets back to Internet format for the reply. In addition, this approach can be used to point to multiple virtual testbeds.

There are still some security issues to be resolved, particularly about how to respect server address-based policy restrictions when the IP address of the source is shielded by the overlay. And work also remains to more fully generalize these proxies, so that they are as architecturally neutral as possible.

One drawback that the virtual overlay approach has is that it cannot control the quality of service (QoS) of packets traversing the virtual testbed. This limits the extent to which QoS architectures can be adequately tested with virtual testbeds, or in fact with any overlay. We do not consider this a fatal flaw. A QoS architecture deployed on a virtual testbed would still deliver *relative* QoS, even if the absolute QoS were not maintained. Moreover, simulation and emulation are quite effective in evaluating QoS proposals and, given the enormous literature on QoS in the past decade, QoS is probably the least-mysterious aspect of new architectures. We think there are many other issues involving routing and addressing that more urgently warrant attention, and for which the virtual testbed approach is well-suited.

Of course, many of these ideas discussed here have been floated before. The virtual testbed borrows heavily from the ideas of the X-bone [20] and Virtual Internet [21], but our emphasis is different. The X-bone suite of tools supports automated establishment and management of overlays; these tools could be used by individual experiments being run on the VT. The focus on the VT is on the virtualization of the overlay nodes themselves; while this is supported in the X-bone architecture (called revisitation), it is not the major focus. The Virtual Internet architecture [21], based in part on the X-bone work, allows multiple levels of virtualization. However, it is closely tied to the current Internet architecture, and is not suitable for experimenting with radical deviations from it.

Beyond this initial prototype, our future plans also include a high performance backbone, built around a set of scalable substrate routers and links provided through the National LambdaRail (NLR) [14]. This will allow the testbed to support larger traffic volumes, with PlanetLab nodes aggregating traffic from local sites and feeding it to the backbone nodes, while also enabling higher bandwidth applications at sites close to backbone nodes. This hybrid approach allows us to capture the benefits of traditional testbeds without inheriting their flaws.

The plan is for the NLR backbone to include routers that also support virtualization. This can be accomplished at sufficient speeds using a pool of processing engines (PEs) interconnected through a high-speed switch. We envision that most processing elements will include a network processor system (NP) capable of high performance packet processing, hosted on a general-purpose processor (CP) that can provide control functions, storage services and facilitate migration from lower performance sequential software designs to the parallelized designs needed to fully exploit network processor architectures. Current-generation network processors provide enough processing resources to deliver approximately 3-5 Gb/s of throughput for moderately complex applications [25], so a backbone node capable of supporting 50 Gb/s of throughput (3 backbone links at 10 Gb/s each, plus 20 Gb/s of access bandwidth) will require 10-16 such processing engines. Processing engines may provide even higher-performance by incorporating advanced FPGAs [26] that combine reconfigurable hardware and multiple processor cores in a single device.

Unlike other pure physical testbeds, our plan to integrate a high-speed backbone with PlanetLab has two major advantages. First, PlanetLab-based overlays serve as an access network for the backbone, bringing traffic from a large user community onto the backbone. Second, developing and deploying the hardware does not gate the architectural work. It is possible for researchers to first experiment with their architecture as an overlay, and later expand it to include the high-speed backbone as the platform supports it.

4. DEPLOYMENT

The traditional but now discredited deployment story is that a "next generation" architecture, after having been validated on a (traditional) testbed, would, through some magical process of consensus and daring, be simultaneously adopted by ISPs and router vendors alike. With this story no longer even remotely possible, is there *any* plausible deployment story? We use the term "plausible" because the adoption of new technologies is an unpredictable process, confounding the expectations of even the most informed observers. Thus, we do not need to know precisely how, and certainly not which, new architectures might be adopted; all we require is that deployment is at least remotely possible.

Our deployment strategy is to leverage the strength of overlays, and not be constrained by their previously limited ambitions. In this scenario, a *new-generation service provider* (NGSP) chooses a particular new architecture and then constructs an overlay supporting that architecture. The NGSP also distributes proxy software that allows anyone, anywhere, to access their overlay. Users of this NGSP that are not directly connected would still be purchasing Internet service from their ISP, but if the overlay is successful then either the NGSP begins offering direct access to customers (by signing agreements with current ISPs, or setting up access technology of their own), or current ISPs, seeing a viable competitive threat, begin to support this new architecture. Note that while we call this an overlay, the NGSP could easily support the new architecture natively on most of its network, so only the first-hop access for users not directly connected would be using the architecture in an overlay mode. Thus, architectures that promised enhanced quality of service could still be deployed in this fashion.

This is little different than the normal overlay deployment story (except that the proxy mechanism is not IP-centric). *Our point, though, is that we should seize overlays as an opportunity to radically change the architecture, not merely provide limited enhancements.* This could be accomplished by a single daring NGSP. It might also arise more naturally because a long-running experiment running on a large and well-maintained virtual testbed is, in fact, nothing more than an NGSP (though, in this case, probably offering its services for free). If the architecture in question offers substantial advantages, it will attract an increasing number of users over time. The architecture could gradually (and seamlessly) migrate from the virtual testbed infrastructure to a more dedicated one, or even remain on a commercial version of a virtual testbed (just as many commercial web sites are located on web hosting services). In this way, natural market forces could take us gradually into a new architectural world.

However, easing the creation of new overlays might not result in a single and radical architectural *winner* but in a large, and ever-changing, collection of more narrowly targeted overlays. In order to avoid architectural chaos and achieve some form of synergy, there must be some consideration, by the overlay designers, of how this union of overlays might be brought together to form a coherent framework and thereby become more than the sum of their individual functions.

Such joint deliberations on how to achieve synergy among overlays may require a sociological change in how we (the research community) interact. When designing a single Internet architecture, we could not afford to ignore each other, since there was only one place that research advancements could take effect. But overlay deployments can occur quite independently without any coordination between, or even cognizance of, other efforts. If overlays are to lead to a substantially different future, such coordination will be required.

5. VIRTUALIZATION: MEANS OR ENDS

The virtual testbed approach uses virtualization in two crucial ways. First, virtualization is used in typically overlay sense; the client proxy plus the virtual links between overlay nodes allows the overlay to, within its confines, be qualitatively equivalent to a native network. This frees users from the tyranny of their local ISP and network providers from having to deploy new functionality at every node. Second, multiplexing of overlay nodes allows there to be many virtual testbeds operating simultaneously. This greatly reduces the barrier-to-entry for any particular experiment.

As we have described it here, these virtualization techniques are used for experimentation and perhaps deployment, but are not tied the nature of the architectures being tested. If architectural changes are rare, with long periods of quiescence (or incremental evolution) between times of architectural revolution, then virtualization is only a means to accomplish these architectural shifts. In that case, one would want every architecture to include the seeds of its own destruction, seamlessly supporting proxy-like functionality and other hooks to make overlay establishment easier, but virtualization need not be more deeply embedded.

If the Internet is, instead, in a constant state of flux, with new architectures always competing against the old and with many narrowly-targeted architectures existing simultaneously, then virtualization may play a more central role. The functionality to support overlays—virtual link establishment and proxy-like reachability—could conceivably become the core functionality of the architecture, its *narrow waist*. In this scenario, PlanetLab would become the model of the Internet.

But this question is not just about virtualization, but also about what we mean by the term “architecture”. To frame this debate, we outline the two extreme points in the spectrum.[1] Internet purists have a very monolithic view of architecture: there is a single universal protocol, currently IP, that is required in each network element and around which all else revolves. Overlays are seen as blights on the architectural landscape, at best necessary evils that are reluctantly tolerated. In this view, virtualization is only a means by which new architectures are installed, not a fundamental aspect of the architecture itself.

Others take a more pluralist approach to architecture, with IP being only one component of an overall system we call the Internet. Overlays are just one more way to deliver the service users want, and are no less appropriate than any other approach to providing functionality. In this view, the architecture is dynamic and evolving and, at any point in time, is defined the union of the various existing overlays and protocols. The ability to support these multiple coexisting overlays then becomes the crucial universal piece of the architecture.

The purist/pluralist split is not just apparent in defining an architecture, but also in evaluating it. Purists aim for flexibility of an architecture, since it will remain in place for a long time; often this flexibility does not result in user benefits in the short-term. Pluralists, on the other hand, put more emphasis on short-term performance improvements, arguing that the desired flexi-

---

[1] The authors of this paper span the entire range of this spectrum, so our extreme characterizations are meant not to belittle any opinion but to clarify, if somewhat overstate, the differences.

bility comes from adding or augmenting overlays rather than in the nature of each individual overlay.

We do not pretend to know which position is right. We hope, however, that the virtual testbed will serve as a fertile *petri dish*, allowing the flowering of many different overlays, with their different characteristics. Perhaps this process will itself be an experiment, out of which we might observe either a drive towards uniformity, or instead a synergy out of dynamic diversity.

6. CONCLUDING REMARKS

The canonical story about the potential impact of architectural research has long been that if testbed experiments show an architecture to be promising, it might be adopted by ISPs and router vendors. This story may have been realistic in the early days of the Internet—certainly DARTnet and other testbeds played an important role in the development of IntServ and Multicast—but it no longer applies. We as a community have long known that there is little chance for adoption of any non-incremental architectural change, and we are rapidly reaching consensus that traditional testbeds are no longer an effective, and certainly not a cost-effective, way of experimenting with new architectures.

As a result of losing this motivating deployment story, the research community has greatly narrowed its focus. The vast majority of Internet research is either empirical (measurement studies) or incremental (modifications that can be deployed without a major change in the architecture). While empirical and incremental research are valuable, they are not sufficient to meet the broader and more fundamental challenges the Internet faces. Our hope is that by providing easy access to virtual testbeds, there will be a renaissance in applied architectural research that is not restricted to incrementally deployable designs. Moreover, by replacing a discredited deployment story with a plausible one closely linked to the experimental methodology, we hope to raise the sights of the research community. It is not enough to complain about our current impasse; it is time to directly confront it and overcome it.

Acknowledgments

We would like to thank Tom Anderson for being an essential contributor to this paper, and to Guru Parulkar for many helpful comments and conversations.

7. REFERENCES

[1] D. G. Andersen. Mayday: Distributed filtering for internet services. In *Proc. USITS Conf.*, Mar 2003.
[2] D. G. Andersen, H. Balakrishnan, M. F. Kaashoek, and R. Morris. Resilient overlay networks. *18th ACM SOSP*, pages 131–145, 2001.
[3] T. Anderson, T. Roscoe, and D. Wetherall. Preventing internet denial-of-service with capabilities. *CCR*, 34(1):39–44, 2004.
[4] B. Braden, T. Faber, and M. Handley. From protocol stack to protocol heap - role-based architecture. *HotNets-I*, Oct 2002.
[5] D. R. Cheriton and M. Gritter. TRIAD: A new next generation internet architecture. http://www-dsg.stanford.edu/papers/triad/triad.html, Mar 2000.
[6] D. Clark, R. Braden, A. Falk, and V. Pingali. Fara: reorganizing the addressing architecture. *FDNA Workshop*, pages 313–321, Aug 2003.
[7] H. Balakrishnan, *et. al.* A layered naming architecture for the internet. *ACM SIGCOMM*, pages 497–506, Aug 2004.
[8] M. Handley and A. Greenhalgh. Steps towards a dos-resistant internet architecture. *FDNA Workshop*, Aug 2004.
[9] Ion Stoica, *et. al.* Internet indirection infrastructure. *ACM SIGCOMM*, pages 73–86, Aug 2002.
[10] J. Kannan, A. Kubota, K. Lakshminarayanan, I. Stoica, and K. Wehrle. Supporting Legacy Applications over Routing Overlays. Technical report, UCB, 2004.
[11] A. Keromytis, V. Misra, and D. Rubenstein. Sos: Secure overlay services. In *Proceedings of ACM SIGCOMM 2002*, Aug 2002.
[12] B. Krishnamurthy, C. Wills, and Y. Zhang. the use and performance of content distribution networks, 2001.
[13] A. Nakao, L. Peterson, and M. Wawrzoniak. A Divert Mechanism for Service Overlay Networks, Dec 2002. Tech. Rep. Princeton Univ.
[14] NLR. New Class of National Networking Infrastructure Launched to Support Cutting-Edge Research and Experimentation, Sep 2003.
[15] Report of nsf workshop on network research testbeds.
[16] L. Peterson, T. Anderson, D. Culler, and T. Roscoe. A Blueprint for Introducing Disruptive Technology into the Internet. In *Proceedings of HotNets–I*, Oct 2002.
[17] S. Savage, *et. al.* Detour: a case for informed internet routing and transport. Technical Report TR-98-10-05, UW, 1998.
[18] L. Subramanian, I. Stoica, H. Balakrishnan, and R. H. Katz. Overqos: offering internet qos using overlays. *CCR*, 33(1):11–16, Mar 2003.
[19] K. Svetz, N. Randall, and Y. Lepage. *MBone: Multicasting Tomorrow's Internet.* IDG Books Worldwide, Inc., 1996.
[20] J. Touch and S. Hotz. The x-bone. *Third Global Internet Mini-Conference at Globecom*, pages 59–68, Nov 1998.
[21] J. D. Touch, Y.-S. Wang, L. Eggert, and G. G. Finn. A virtual internet architecture. *ISI Technical Report ISI-TR-2003-570*, Mar 2003.
[22] C. Tschudin and R. Gold. Network pointers. *SIGCOMM Comput. Commun. Rev.*, 33(1):23–28, 2003.
[23] M. Walfish, J. Stribling, M. Krohn, H. Balakrishnan, R. Morris, and S. Shenker. Middleboxes no longer considered harmful. *To appear in OSDI*, Dec 2004.
[24] D. J. Wetherall. Active network vision and reality: lessons from a capsule-based system. *17th ACM SOSP*, pages 64–79, Dec. 1999.
[25] T. Wolf and M. Franklin. CommBench - A Telecommunication Benchmark for Network Processors. In *IEEE Intl. Symp. on Perf. Analysis of Syst. and Software*, Apr 2000.
[26] Xilinx. Virtex-II Pro Platform FPGAs: Introduction and Overview. DS083-1 (v3.0), Dec 2003.
[27] X. Yang. Nira: a new internet routing architecture. *FDNA Workshop*, pages 301–312, Aug 2003.
[28] D. Zhu, M. Gritter, and D. R. Cheriton. Feedback based routing. *CCR*, 33(1):71–76, 2003.