Internet Clean-Slate Design: What and Why?

Anja Feldmann
Deutsche Telekom Laboratories / TU Berlin
anja.feldmann@telekom.de

This article is an editorial note submitted to CCR. It has NOT been peer reviewed. Authors take full responsibility for this article's technical content. Comments can be posted through CCR Online.

ABSTRACT

Many believe that it is impossible to resolve the challenges facing today's Internet without rethinking the fundamental assumptions and design decisions underlying its current architecture. Therefore, a major research effort has been initiated on the topic of Clean Slate Design of the Internet's architecture. In this paper we first give an overview of the challenges that a future Internet has to address and then discuss approaches for finding possible solutions, including Clean Slate Design. Next, we discuss how such solutions can be evaluated and how they can be retrofitted into the current Internet. Then, we briefly outline the upcoming research activities both in Europe and the U. S. Finally, we end with a perspective on how network and service operators may benefit from such an initiative.

Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design

General Terms

Design, Management, Performance, Reliability

Keywords

Clean-Slate, Post-IP, Internet, network architecture

1. INTRODUCTION

The Internet is a social phenomenon that has changed, and continues to change how humans communicate, businesses work, how emergencies are handled, the military operates, etc. It has redefined expectations about how interactions between humans, computers and humans, and between computers function. Without question, almost all major industrial sectors take advantage of the Internet. This includes software companies such as Microsoft, Google, and SAP, as well as more traditional manufacturing companies (including the automotive industry), service providers (including banks and insurance companies), as well as the entertainment industry.

Given its impact, the question is what people mean when they refer to the Internet. For the billions of users of networking technology, the Internet equates to the applications enabled by the technology: the Web, file sharing, chat, IP telephony, to name just a few. For some it is the protocol suite underlying the Internet including the Internet Protocol (IP), the User Datagram Protocol (UDP), and Transmission Control Protocol (TCP) as well as the routing protocols. For others it consists of the networking elements such as hubs, switches and routers, as well as the manner by which information is transmitted, optically, electronically, or wirelessly. A further group views the Internet as building, operating, and maintaining an infrastructure such as a LAN or the Internet Service Provider (ISP) backbone. Others still are interested in observing and characterizing the traffic, and the users who are responsible for the traffic on such networks. Another group focuses on how to achieve commercial success in the Internet age.

This wide spectrum of views reflects the huge success of the Internet, but it also hints at the complexity and diversity of a system which has grown from interconnecting merely a few supercomputers to interconnecting the world. Users of the Internet value the new and diverse set of applications with their interactivity, while developers of applications value the ease with which they can develop new functionality and reach a large and diverse set of users.

Indeed, the success of the current Internet is highlighted by how it has influenced our society. Yet at the same time, society is daring the Internet to face the following set of **challenges**:

Security: The lack *of security* in the Internet is worrisome to everyone including users, application developers, and network and service operators.

Mobility: Currently, application developers find little support for new *mobile* applications and services.

Reliability and availability: ISPs face the task of providing a service which meets user expectations of the Internet's crucial role in both business and private life, in terms of reliability, resilience, and availability, when compared, for example, to the telephone network (five nines[1]). Furthermore, the service has to be seamless.

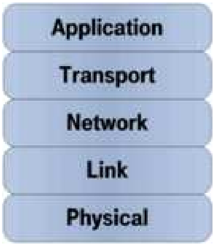
Problem analysis: The toolset for *debugging* the Internet is limited, e. g., tools for root cause analysis.

Scalability: Questions remain regarding the scalability of some parts of the current Internet architecture, e. g., the routing system.

Quality of Service: It is still unclear how and where to integrate different levels of quality of service into the architecture.

Economics: Besides these more technical questions, there is also the question of how network and service operators can continue to make a profit.

[1]Five nines implies an availability of 99.999%. This means that the system is highly available, delivering its service to the user 99.999% of the time it is needed.



- **Application:** supporting network applications
- **Transport:** host-host data transfer
- **Network:** uniform format of packet, routing of datagrams from source to destination
- **Link:** data transfer between neighboring network elements
- **Physical:** bits "on the wire"

Figure 1: Internet Protocol Stack

The above challenges are well recognized. Indeed, within the last decade the community has tried to find solutions. But the proposed solutions are only partial solutions to each individual challenge at best. Why? To understand this, Section 2 reviews the design goals and resulting principles of the current Internet architecture; Section 3 discusses why the current design decisions hinder the solutions. Many believe that it is necessary and timely to rethink the fundamental assumptions and design decisions, and start from scratch: via a Clean-Slate Design approach, see Section 4, to achieve solutions which address all of the above challenges simultaneously. The strategy for proceeding along this path is discussed in Section 5. Possible impacts are discussed in Section 6.

2. REVIEW OF THE CURRENT INTERNET ARCHITECTURE

Before we can discuss why it is difficult to address the above challenges within the current Internet architecture we need to briefly review how the current Internet works.

The **design goals** [1] underlying the current Internet architecture in order of importance are:

(0) to connect existing networks,
(1) survivability,
(2) to support multiple types of services,
(3) to accommodate a variety of physical networks,
(4) to allow distributed management,
(5) to be cost effective,
(6) to allow host attachment with a low level of effort and,
(7) to allow resource accountability.

To achieve these goals, the following **design principles** have been used:

(a) layering,
(b) packet switching,
(c) a network of collaborating networks,
(d) intelligent end-systems as well as the
(e) end-to-end argument.

Next, we review how these design principles enable today's Internet to fulfill most of the design goals laid out above.

Layering.

The use of network layers, see Figure 1, leads to a network stack and offers a reduction in complexity, isolation of functionality, and a way to structure their network protocol designs. Each layer in the network stack offers a service to the next layer up in the stack. It implements this service using the services offered by the layer below. This results in a situation where the logical communication happens within each layer. Yet during actual communication, the data passes the network stack at the sender from the top to the bottom and at the receiver from the bottom to the top.

The Internet has the following five layers (top to bottom): application, transport, network, link, and physical. The physical layer is responsible for coding the data and transporting it over the wire/ether. The link layer enables neighbor-to-neighbor communication. The network layer, also often called the IP layer, enables host-to-host communication and, as such, provides a way of addressing hosts (via IP addresses), sending data (via IP packets), as well as determining routes. The transport layer enables application-to-application communication either as bitstream via TCP or as message service via UDP. TCP offers reliable data transfer, with flow and congestion control, while UDP allows the chance of sending and/or receiving of messages. These are two types of services (design goal 2) currently offered by the Internet. The application layer implements the application-specific protocol exchange, e. g., HTTP or FTP. The interface between the application and the transport layer is the Socket API.

The use of communication layers enables the simple interconnection of existing networks (design goal 0) and enables the accommodation of a variety of networks (design goal 3). As soon as a network offers the service required by a specific layer it can be seen as implementing that layer. In the case of the Internet this happens at the network layer. Almost any network fulfills the criteria of the service needed by the network layer: to deliver packets to their neighbor where some packets may be lost.

Packet switching.

The decision to use packet switching implies that the data has to be split into packets. Each packet carries the address of its destination and traverses the network independently of the other packets. Any packet can use the full link bandwidth on any link but may have to wait in a queue if other packets are already using the link. Should a packet encounter a full queue it is simply dropped, which corresponds to the best effort service principle. This means that it is possible to use a stateless routing system at the network layer, which does not require per connection state. This ensures scalability and contributes to cost effectiveness (design goal 5).

Network of collaborating networks.

In the Internet, routing decisions are taken on a per-IP-network-basis (a set of related IP addresses) based on the routing table at each router, which is computed in a distributed manner. Indeed, the Internet is divided into a collection of autonomous systems (ASs). Each AS is managed by an Internet Service Provider (ISP), which operates a backbone network that connects to customers and other service providers. Within an AS, routing is determined by interior gateway protocols such as OSPF and IS-IS [2]. Routing between ASs is controlled by the Border Gateway Protocol (BGP). BGP is a policy-routing protocol, which distributes routing information between routers belonging to different autonomous systems. Each router determines the next hop router by combining the information learned via these routing protocols. This design of the routing system ensures survivability (design goal 1) and allows for distributed management (design goal 4) as long as the ISPs collaborate.

Intelligent end-systems / the end-to-end argument.

The fact that the network layer can simply drop packets is a result of keeping the network dumb and placing the intelligence at the end-system. Should the application require reliable data transfer, then it is the responsibility of the end-system to provide the service, e. g., in the transport layer via TCP. Indeed, the end-to-end argument can be used as a way to place functionality. There are two reasons to place functionality inside the network rather than at the end-systems: if all applications need it, or if a large number of applications benefit from an increase in performance. This is not the case for reliability. Not all applications require it, e. g., VoIP, and applications often have to implement end-to-end reliability anyhow, e. g., the domain name system (DNS). Accordingly, both packet switching and the end-to-end argument, help to ensure survivability (design goal 1) and cost effectiveness (design goal 5).

The original Internet design principles ensure that the Internet fulfills most of the original Internet design goals (0-5). The other design goals have been addressed by crutches such as DHCP (design goal 6) or the simple network management protocol (SNMP) and NetFlow[2] (design goal 7).

3. WHY TODAY'S INTERNET ARCHITECTURE CANNOT FULFILL THE CHALLENGES

Unfortunately, if we compare the original list of Internet design goals with today's challenges, see Section 1, we note that these challenges are not addressed by the current Internet architecture.

While a lot of work is underway to add **security** to each individual protocol used in the Internet, e. g., IPsec, DNSSEC, this has not resulted in a secure Internet (design principle a). Indeed, a composition of two secure components does not necessarily result in a secure system. Any system is only as secure as its weakest component. In fact the smallest oversight can lead to a global security problem, especially since significant functionality is placed on vulnerable end-systems (design principles d and e). Moreover, adding security to the Internet, which is fundamentally based on the idea of trusted and cooperating systems (design principle c), is a difficult balancing act between usability, performance, and security.

Adding **mobility** to the current Internet architecture is also difficult, as the current Internet naming system is based on the host address, typically the IP address (design principle d). To achieve scalability of routing, the Internet uses an address hierarchy, which imposes a structure on the host addresses that relates to its location within the Internet (design principles b and c). Most suggestions regarding how to enable mobility either break the routing hierarchy or require the use of another IP address. The first threatens scalability and IP address filtering, a security crutch that provides a primitive firewall. The latter either requires changes at all servers or a decoupling of addressing for the purpose of routing and addresses as used by applications (such as address-based authorization), or a fundamentally new approach to naming.

Given the distributed management of the current Internet (design principle c), and the lack of tools for identifying which applications are currently using which specific network resources and vice versa (which network components are being used by which users), **network management** is an unsolved problem (design principle b). While we understand quite well how to forward packets quickly in the "forwarding plane", we still do not understand how to set up the "control plane" in such a manner that the network operates **reliably**, is easily **manageable**, **debuggable**, and still **scales** well. Network management is another topic which spans all network layers including the political. For her part, this author is scared if the Internet is not working and the phone dead, due to VoIP, and it is therefore impossible to call one's network administrator.

While mechanisms for providing **Quality of Service (QoS)** within the Internet as well as Asynchronous Transfer Mode (ATM) networks have been very well studied, the interaction problems between the network layers (design principle a) are still unresolved and the management of such services, including configuration, policy setup, charging, inter-provider setups, etc. is still open (design principles b and c).

All topics, security, mobility, network management, and QoS, span the whole network stack. Accordingly, it is time to rethink the design principles: layering, packet switching, collaborating networks, as well as the end-to-end argument. It is, however, impossible to rethink the network structure on a technical level without addressing the **business aspects**. In the current Internet, network infrastructure providers, such as ISPs, get their main revenue from end-users who pay for network connectivity. Service providers, e. g., Google, get their main revenue from advertisers who pay for eyeballs. In the past, user micro-payments have proven to be too unpredictable and too much of a burden to be acceptable to users. History has shown that users usually prefer flat rates or subscriptions. Any new architecture has to have a simple way to handle financial settlements and to accommodate federations of networks. Indeed, there is a need for an economic model as well as a technical one that makes sense for the evolution of the network and its services, and the continued viability of both.

4. THE CLEAN-SLATE APPROACH

There are two principal ways in which to evolve or change a system:

incremental: a system is moved from one state to another with incremental patches.

clean-slate: the system is redesigned from scratch to offer improved abstractions and/or performance, while providing similar functionality based on new core principles.

In the past 30 years the Internet has been very successful using an incremental approach. However due to its success, the community has now reached a point where people are unwilling or unable to experiment on the current architecture. Therefore, it might be time to explore a clean-slate approach consisting of: out of the box thinking, the design of alternative network architectures, and experimentation with the architecture in order to evaluate the ideas and to improve them as well as to give them a realistic chance of deployment either in a new system or incrementally on/in today's network.

Why now? The current set of design principles are intrinsic to the current Internet architecture of the Internet and therefore hard to challenge and hard to change. Yet as we have seen, the challenges above, individually and together, are hard to address given these design principles. Furthermore, advances in technology have made new capabilities available, which question some of the old design principles: fast packet optical components, wireless networks, fast packet forwarding hardware, virtualization techniques, and significant computational resources.

[2]NetFlow is an open but proprietary network protocol developed by Cisco Systems for collecting IP traffic information.

4.1 Clean-Slate thinking

As the community does not know what a new architecture will look like, out of the box thinking is necessary. In fact, the ability to rethink the network and service architecture is likely to result in a different network architecture. Particular drivers include the current state of technology, a different set of design goals, different priorities, and, therefore, an alternative placement of functionality.

However, a new network architecture is not sufficient in itself to address the network management questions, as network management includes the processes surrounding the network. Accordingly it is important to give researchers a chance to experience the complexity of managing such networks themselves to induce them to explore alternatives.

4.2 Evaluation of a Clean-Slate attempt

The biggest challenge in moving forward with a Clean-Slate approach is that we need to have a way to determine when the newly designed architecture is sufficiently good. Even more challenging is that this has to be possible without knowing what such an architecture might look like. We know of multiple ways in which to approach this, including "paperware" which is insufficient and "prototypes". The use of prototypes is crucial, as one needs to build a system in order to evaluate it and to convince others that it is the appropriate solution. It is almost impossible to get a new idea adopted that has not yet been tried at scale and under realistic conditions. But more importantly it is needed intellectually as it enables the researchers to uncover things that would otherwise have been assumed away. Thus there is a further aspect to Clean-Slate Design besides research into new network architectures: *building an experimental facility*.

4.3 Experimental facility vs. testbed

The experimental facility is not yet another testbed. Testbeds, while real and not simulated, are designed for a specific purpose with a focused goal, a known success criterion, and are of limited scale. Therefore, they are not sufficient to evaluate new network architectures. Rather the purpose of the experimental facility is to explore yet unknown architectures while exposing researchers to the real thing. Furthermore it has to be of larger scale and include different technologies. But most importantly, the success criteria of each experiment remain to be determined. While the experimental facility should attract real users and their traffic it cannot, however, be a production network as the services are experimental and may fail. Some experiments may even break their part of the infrastructure. The experimental facility should enable parallel experiments with totally different networking architectures, e. g., different naming schemes, different layering approaches, and different forwarding strategies to co-exist and operate in parallel. At the same time, new services should be able to explore the new capabilities and should be made available to users who opt-in.

4.4 Deployment of Clean-Slate ideas

Just because a Clean-Slate approach is advocated does not mean that research on incrementally improving the current architecture should be halted, rather the opposite. It can be expected that some of the ideas originally proposed as part of a new architecture can be retrofitted to the current Internet. This has happened, e. g., with IPv6. Furthermore, once a new architecture has been identified, it is quite reasonable to identify intermediate steps so that the current architecture can evolve into the desired new one. In addition, experience and tools developed in operating and managing an experimental facility may prove to be very valuable for managing the current Internet or/and private Intranets. In other words, Clean-Slate should be viewed as a design process, not a result in itself.

4.5 Success stories of Clean-Slate ideas

Several different success stories of a Clean-Slate process are possible. For example, the innovative services and applications, which will be developed within the process, may become mature enough to be commercially deployed on the existing Internet. An alternative is that the research community will create a new network architecture, which eventually replaces today's architecture. A further possibility is that the insights gained from the process are taken up by commercial players and fitted back into today's network architecture. The most "conservative" outcome may be that we learn that the current Internet architecture is the "best" possible solution. The most "radical" outcome is that the experimental facility, which allows multiple sub-system architectures and network services to co-exist, may become the blueprint for the future Internet.

5. RESEARCH PLAN

As mentioned above, there are two aspects to Clean-Slate Design:

1. research into new network architectures.
2. building an experimental facility.

The research programs both in the U. S. as well as in Europe reflect this.

5.1 New network architectures

In the U. S. the National Science Foundation (NSF) has initiated the Future InterNet Design (FIND) research program [3] within the NSF NeTS program which poses the following questions: "What are the requirements for the global network of 15 years from now – what should that network look like and do?" and "How would we re/conceive tomorrow's global network today, if we could design it from scratch?" As such FIND has a strong focus on defining an architecture for the Future Internet. Furthermore, it plans to encourage research teams to reach consensus on broad architectural themes. The first set of proposals within the FIND program have been selected, and a detailed overview is given by David Clark [4]. A similar initiative is planned in Asia and within the upcoming FP7 calls in the EU. Indeed, the ongoing COST strategic action ARCADIA [5], the EIFFEL Think Tank [6], and the Future Internet Initiative [7] are already under way to identify and align groups within Europe.

5.2 Experimental facility

At the same time, NSF is planning the Global Environment for Networking Innovations (GENI) initiative as a NSF Request for Major Research Equipment to the U. S. congress, aiming to "Build an open, large-scale, realistic experimental facility for evaluating new network architectures." The intention of GENI is to offer a shared, global facility designed to catalyze research on network architectures, services and applications. A similar initiative is planned within the upcoming FP7 calls and in Asia.

5.2.1 GENI: one extreme

The plan is that GENI [8] consists of a set of typical networking components including links, forwarders, storage and processor

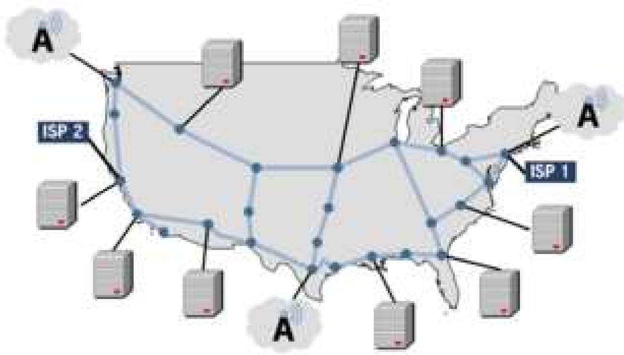


Figure 2: Planned GENI topology

clusters, and wireless subnetworks. These can then be partitioned by the management framework in such a way that each network experiment is an overlay that has exclusive rights to some subset of the components. The design of GENI is made possible by four key ideas: programmable components, virtualization of the components, seamless opt-in of the users, and modularity of the components. The envisioned topology is shown in Figure 2. A discussion of the GENI design principles is given by Peterson et al. [9]. While funding for the main GENI facility is expected to start in 2009/2010, funding from the regular NSF budget is being used to support research and prototyping efforts to reduce the technical risk in the construction of GENI.

This includes work on programmable components, such as:

- FPGAs by McKeown[10],
- routers [11] by Turner,
- wireless networks [12] by Sabharwal et al.,

on work on virtualization, such as:

- of network infrastructures [13] by Rexford et al.,
- of wireless networks by Raychaudhuri et al. [14] and by Banerjee [15],

on monitoring and measurement, such as:

- the Meta-Management System by Ng et al. [16],
- the control plane for experimental support by Anderson et al. [17],
- authorization architectures by Anderson et al. [18],
- sensor network testbeds for urban monitoring by Welsh et al. [19].

and on federation:

- Emulab-PlanetLab Federation by Lepreau [20].

The GENI vision is to have a mixture of speculative experiments and long-running deployment studies running side by side. The long-running studies are supposed to offer real services, which will attract real users and be helpful in identifying the right architecture.

Neither the research work into new architecture nor the experimental facility will start from scratch. For example, new architectural concepts have been developed within the US-NewArch [21] and the EU-Daidalos project [22]. The topic of naming and addressing for the next-generation Internet has been the focus of a recent workshop [23].

The experimental facility can build upon the experience gathered within research networks, among which are GEANT2 and Internet2. GEANT2 is a multi-gigabit pan-European data communications network, reserved specifically for research and educational use. Internet2's goal is to develop and deploy advanced network applications and technologies for research and higher education, while accelerating the creation of tomorrow's Internet. Other experimental facilities developed by networking researchers for networking research include PlanetLab [24] and OneLab [25] as well as Emulab.

PlanetLab is the result of a joint community effort by a large international group of researchers to provide a geographically distributed platform for deploying, evaluating, and accessing planetary-scale network services. Each group contributes some hardware resource and in return has the opportunity of accessing one or more isolated slices of PlanetLab's global resources. OneLab focuses on the federation of PlanetLabs and plans to widen it by adding testbed nodes behind links that are not typical research network links. Furthermore, the ability of applications to perceive the underlying network environment will be enhanced. Emulab [26], on the other hand, provides integrated access to a wide range of experimental environments: from simulated to emulated to wide-area network testbeds.

On the software side, there are modular software routers such as Click [27] and Xorp [28] as well as virtualization systems such as Xen [29] and Vmware, which are openly available, can easily be changed and are already heavily used by the research community.

5.2.2 Alternatives

One can argue about how much flexibility such an experiment facility has to provide and initially on what layer with which granularity. GENI represents an extreme approach with virtualization at all layers of dedicated resources for reproducibility, full programmability, and user opt-in. It may, however, be quite sensible to, for example, start with dedicated resources at only the link layer, offering customization of the configuration as well as computational resources and user-opt in. Such an experimental platform might very well provide all the resources needed to experiment with alternative control planes and federation of networks. Indeed, even in such a more simple facility, the difficult questions of how to combine resources, how to manage experiments, how to allocate resources, how to extend and evolve the facility, and deciding on an appropriate usage policy have to be addressed.

This approach allows us to take advantage of already existing testbeds, e. g., Mupbed [30]. These may become components of a larger experimental facility and therefore reduce deployment time and cost. Moreover, it places an emphasis on the researchers to handle the question of federation and policy.

This brings us to the interesting question of whether the technology is going to change too fast for an experimental facility to offer attractive hardware. The way to address this issue is by adding resources as the project develops, using the facility as it is built, and adapting the process when necessary, as is planned within GENI.

6. IMPACT OF THE CLEAN-SLATE DESIGN PROCESS

The community can benefit in many different ways from the process of searching for a new network architecture. For example, applications and services will be able to take advantage of the enhanced capabilities such as security and mobility provided by the new network. It should ease access to information and also enable applications which we cannot even imagine today. Furthermore, new economic models may reshape the market-place and thus business models. Players who react fast will have an advantage compared to competitors. In addition, the resulting architecture may be less complex and therefore easier to manage and maintain than the current one.

There is a chance that a new interface between network service providers and network capacity providers will arise. Being involved in the design process will enable an operator not only to be a step ahead in business but also to shape the interface as well as the changing value-chain. Most importantly, a provider will be able to position itself in the market and react to upcoming changes appropriately. Moreover the role of the network and service provides may change. Open interfaces may enable new ecosystems of dynamically changing business alliances, for example, between providers of network/computing/storage/service resources. New roles for network and service operators (e. g., brokers, mediators, and orchestrators of complex services) may arise and it is important for an operator to shape these developments rather than just to observe them.

By operating an experimental facility, new network management and control capabilities will be developed. It is highly likely that these will immediately change the way that today's enterprise networks, and even today's Internet, are managed and therefore reduce daily operating overhead. Moreover, it has the potential to change the way that the control plane is handled in the long run and therefore have an impact on the operating and investment costs in the network itself as well as on network operation.

Acknowledgments

Many thanks go to Jennifer Rexford and Roger Karrer for useful discussions and feedback about clean slate design in general and FIND and GENI in particular. Moreover, I am thankful for input from Deutsche Telekom staff, especially Michael Dueser, Hans Einsiedler, Peter Feil, and Hans-Martin Foisel.

7. REFERENCES

[1] D. D. Clark, "The design philosophy of the darpa internet protocols," in *Proc. ACM SIGCOMM*, 1988.
[2] C. Huitema, *Routing in the Internet*. Prentice Hall, 1999.
[3] "Future INternet Design (FIND)." http://www.nsf.gov/pubs/2007/nsf07507/nsf07507.htm.
[4] D. D. Clark, "Summary of 2006 FIND proposals," 2006.
[5] "Building the future internet: From fundamentals to experiments." http://www.cost.esf.org/index.php?id=989.
[6] "Eiffel." http://www.future-internet.eu/.
[7] "The future-internet initiative." http://www.future-internet.eu.
[8] L. P. (Ed), "GENI design principles," Tech. Rep. 06-08, GENI Design Document, 2006.
[9] L. Peterson, T. Anderson, D. Blumenthal, D. Casey, D. Clark, D. Estrin, J. Evans, D. Raychaudhuri, M. Reiter, J. Rexford, S. Shenker, and J. Wroclawski, "GENI design principles," *IEEE Computer Magazine*, 2006.
[10] N. McKeown, "NetFPGA." http://www.geni.net/dev_NetFPGA.txt.
[11] J. Turner, "Development of a diversified router for experimental research in networking." http://www.geni.net/dev_diversified_router.txt.
[12] A. Sabharwal and E. Knightly, "Wireless open access research platform (WARP)." http://www.geni.net/dev_WARP.pdf.
[13] J. Rexford and L. Peterson, "VINI: Virtual network infrastructure." http://www.geni.net/dev_VINI.txt.
[14] D. Raychaudhuri, S. Paul, M. Gruteser, and I. Seskar, "Prototyping for wireless virtualization and wired-wireless virtualization." http://www.geni.net/dev_GENI_wireless.doc.
[15] S. Banerjee, "Time-based wireless virtualization." http://www.geni.net/dev_banerjee.pdf.
[16] E. Ng and H. Zhang, "Meta-management system for GENI." http://www.geni.net/dev_metamgt.doc.
[17] T. Anderson and A. Krishnamurthy, "A control plane for GENI experiment support." http://www.geni.net/dev_control_plane.txt.
[18] T. Anderson, L. Bauer, A. Krishnamurthy, and M. Reiter, "A prototype of the GENI authorization architecture." http://www.geni.net/dev_SecurityPrototype.txt.
[19] M. Welsh and J. Bers, "Large scale open sensor network testbed for urban monitoring (CitySense)." http://www.geni.net/dev_citysense_summary.pdf.
[20] J. Lepreau, "Emulab-PlanetLab federation." http://www.geni.net/dev_emu-plab-fed.pdf.
[21] "NewArch Project: Future-Generation Internet Architecture." http://www.isi.edu/newarch/.
[22] "Designing Advanced network Interfaces for the Delivery and Administration of Location independent, Optimised personal Services." http://www.ist-daidalos.org/.
[23] "Dagstuhl Seminar 06441: Naming and Addressing for Next-Generation Internetworks." http://kathrin.dagstuhl.de/06441/.
[24] T. Anderson, J. Crowcroft, D. Culler, R. McGeer, L. Peterson, and T. Roscoe, "PlanetLab." http://www.planet-lab.org/.
[25] T. Friedman, S. Fdida, P. Duval, N. Lafon, and X. Cuvellier, "OneLab." http://www.one-lab.org.
[26] J. Lepreau, "EmuLab." http://www.emulab.net.
[27] E. Kohler, B. Chen, D. D. Couto, F. Kaashoek, R. Morris, and M. Poletto, "The click modular router project." http://pdos.csail.mit.edu/click/.
[28] F. Bauer, A. Bittau, J. Cardona, A. Ghosh, A. Greenhalgh, M. Handley, O. Hodson, E. Kohler, P. Radoslavov, L. Rizzo, B. M. Simpson, and M. Zec, "XORP – open source IP routers." http://www.xorp.org.
[29] P. Levine, I. Pratt, S. Crosby, F. Artale, J. Bara, E. Charles, J. Glendenning, and G. Mangione, "Xen." http://www.xensource.com.
[30] "MUPBED – multi-partner european testbeds for research networking." http://www.ist-mupbed.org/.