Computer Communications 34 (2011) 2–42

Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

Architectures for the future networks and the next generation Internet: A survey

Subharthi Paul [*], Jianli Pan, Raj Jain

Department of Computer Science and Engineering, Washington University in Saint Louis, United States

ARTICLE INFO

Article history:
Received 21 August 2009
Received in revised form 23 July 2010
Accepted 3 August 2010
Available online 25 August 2010

Keywords:
Future Internet architecture survey
Future Internet design (FIND)
Future network architecture
Next generation Internet
Global Environment for Networking Innovations (GENI)

ABSTRACT

Networking research funding agencies in USA, Europe, Japan, and other countries are encouraging research on revolutionary networking architectures that may or may not be bound by the restrictions of the current TCP/IP based Internet. We present a comprehensive survey of such research projects and activities. The topics covered include various testbeds for experimentations for new architectures, new security mechanisms, content delivery mechanisms, management and control frameworks, service architectures, and routing mechanisms. Delay/disruption tolerant networks which allow communications even when complete end-to-end path is not available are also discussed.

1. Introduction

The Internet has evolved from being an academic pursuit to a huge commercial commodity. The IP thin waist associated with the simplicity of the present design has been a remarkable architectural choice, motivated by the need to converge multiple link layer technologies and end-to-end transport mechanisms. However, the assumptions under which the original Internet was designed have changed. Newer contexts and specific requirements have subjected the original design paradigms of the Internet to a lot of abuse. Due to the limitations of the underlying architecture, such overlaid hacks have limited effectiveness and are often highly inefficient.

Commercialization of the Internet has introduced concerns about security, trust, and value added services. Introduction of networkable wireless systems has brought about a mobile paradigm. Use of the Internet as a communication commodity upon which business communications depend has raised the need for better resilience and fault tolerance through fine-grained control and management. A best effort delivery model of IP is no longer considered adequate. Routing is no longer based on algorithmic optimization, but rather has to deal with policy compliance. Assumptions about persistently connected end systems do not hold with the introduction of delay tolerant networking paradigms. Protocols designed without concern for energy efficiency cannot integrate energy conscious embedded system networks such as sensor networks. Initial projections about the scale of the Internet have long since been invalidated, leading to the current situation of IP address scarcity, BGP table growth, etc. The wide scale proliferation and service diversification of the Internet have led to forceful "plumbing-in" of external architectural artifacts into the core design. Such plumbing-in is not seamless, marring the simplicity of the original IP design and introducing numerous side effects.

Several of the most relevant and immediate problems for which the current Internet design has failed to provide a satisfactory solution have been discussed in [78]. Another reference to a comprehensive discussion on the history of the Internet is John Day's book on "Patterns in Network Architecture: A Return to Fundamentals" [40]. The book characterizes the underlying motivations and reasoning behind the key technologies of the current Internet. It also describes in detail how factors other than technical ones affected the shape of the current Internet architecture.

Over the years, networking research has introduced newer protocols and newer architectural designs. However, as already mentioned, the Internet is its own worst adversary. It has not been possible to introduce any fundamental changes to its basic underlying architecture. Small and incremental changes solving the current problems have introduced scores of others. The myopic view of incremental approaches has arguably stretched the current design to the maximum. The Internet needs to be redesigned for the present needs, while at the same time ensuring enough flexibility to adequately incorporate future requirements.

A new paradigm of architectural design described as "clean-slate design" goes against the more traditional approach of incremental design. The theme of "clean-slate design" is to design the

[*] Corresponding author. Tel.: +1 773 679 7723.
E-mail addresses: pauls@cse.wustl.edu (S. Paul), jp10@cse.wustl.edu (J. Pan), jain@cse.wustl.edu (R. Jain).

0140-3664/$ - see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.comcom.2010.08.001

system from scratch without being restrained by the existing system, providing a chance to have an unbiased look at the problem space. However, the scale of the current Internet forbids any changes, and it is extremely difficult to convince the stake-holders to believe in a clean-slate design and adopt it. There is simply too much risk involved in the process. The only way to mitigate such risks and to appeal to stake-holders is through actual Internet-scale validation of such designs that show their superiority over the existing systems. Fortunately, research funding agencies all over the world have realized this pressing need and a world-wide effort to develop the next generation Internet is being carried out. The National Science Foundation (NSF) was among the first to announce a GENI (Global Environment for Networking Innovations) program for developing an infrastructure for developing and testing futuristic networking ideas developed as part of its FIND (Future Internet Design) program. The NSF effort was followed by the FIRE (Future Internet Research and Experimentation) program which support numerous next generation networking projects under the 7th Framework Program of the European Union, the AKARI program in Japan, and several other similarly specialized programs in China, Australia, Korea, and other parts of the world.

The scale of the research efforts to develop a next generation Internet proves its importance and the need for its improvement to sustain the requirements of the future. However, the amount of work being done or proposed may baffle someone who is trying to get a comprehensive view of the major research areas. In this paper, it is our goal to explore the diversity of these research efforts by presenting a coherent model of research areas and by introducing some key research projects. This paper does not claim to be a comprehensive review of all of the next generation Internet projects but may be considered as an introduction to the broader aspects and some proposed solutions.

Next generation Internet research efforts can be classified under the primary functions of a networking context such as routing, content delivery, management and control, and security. We argue against such an organization of the research efforts with the view that this organization is contrary to clean-slate design. A clean-slate view of isolated problems in a specific functional area do not necessarily fit together to define a seamlessly integrated system. This is because they are defined under fixed assumptions about the other parts of the system. The result is that the best individual solutions often contradict each other at the system level. For example, a clean-slate centralized management and control proposal may interfere with the objectives of a highly scalable distributed routing mechanism, rendering both the solutions useless in the systems perspective. Also, we believe that the current Internet and its success should not in any way bias "clean-slate" thought. Designers should be able to put in radical new ideas that may have absolutely no semblance to any design principle of the current Internet. At present, there are very few architectures that actually focus on a holistic design of the next generation Internet. Some holistic designs have been proposed under service centric architectures [discussed in Section 7]. Most service centric architectures design new service primitives and propose holistic architectural frameworks for composing applications over these federated service primitives. An example of such an architecture is the Internet 3.0 architecture [discussed in Section 7.6].

In this survey, a major portion of the research being undertaken in the area of next generation Internet research is covered. First, we survey some of the more progressive and interesting ideas in smaller, more independent research areas and classify them in various sections as follows:

1. Security: In the current Internet, security mechanisms are placed as an additional overlay on top of the original architecture rather than as part of the Internet architecture, which leads to a lot of problems. In this section, several new propositions and on-going research efforts that address the problems of security from a different perspective are analyzed and discussed. This includes proposals and projects related to security policies, trust relationships, names and identities, cryptography, anti-spam, anti-attacks, and privacy.
2. Content delivery mechanisms: This section deals with research on new mechanisms for content delivery over the Internet. The next generation Internet is set to see a huge growth in the amount of content delivered over the Internet, and requires robust and scalable methods to prepare for it. Also, discussed are newer paradigms for networking with content delivery at the center of the architecture rather than connectivity between hosts, as in the current architecture.
3. Challenged network environments: Contrary to the intrinsic assumption of "continuously connected" context over which communication protocols are developed, "challenged network" research focuses specifically on heterogeneous networking environments where continuous end-to-end connectivity cannot be assumed. The intermittent connectivity could be due to either planned or unplanned disruptions. Planned space networks are examples of planned disruption contexts depending on fixed schedules of satellite and planetary motions. Wireless ad hoc networks represent an unplanned disruption context wherein unplanned disruptions may be caused by a variety of factors, such as node failures, mobility, limited power, and disconnected topology. The discussions in this section relate to two important perspectives of the future Internet design requirements: Energy efficient protocol design and implementation and federation of heterogeneous networking environments.
4. Management and control framework: The current Internet works on a retro-fitted management and control framework that does not provide efficient management and troubleshooting. The proposals for the future Internet in this area vary from completely centralized ideas of management to more scalable and distributed ideas. The discussion in this section relate to the issues of management and control in the current Internet as well as some of the proposals for the future.
5. Internetworking layer design: This section is mainly dedicated to novel and futuristic proposals addressing problems at the internetworking layer of the Internet. The primary functions of the internetworking layer are routing and forwarding. In this section, we will discuss some of the design proposals for the internetworking layer of the future Internet. While some proposals try to address the immediate concerns with IP based routing, others are more futuristic and propose fundamental changes to the routing paradigm.

Next we look at some holistic architectural frameworks under Section 7 on "Service Centric Architectures." The commercial usage of the Internet, ubiquitous and heterogeneous environments, and security and management challenges require the next generation Internet to provide a broad range of services that go far beyond the simple best effort service paradigm of today's Internet. In this section, several proposals on designing next generation service architectures are discussed. Some key design goals for the next generation service architecture include flexibility and adaptability, avoiding the ossification of the current Internet and facilitating mapping of user-level service requirements onto the lower infrastructure layers.

Finally, we take a look at the next generation research on "Future Internet Infrastructure Design for Experimentation" in Section 9. This section discusses the various efforts to develop testbed architectures that can support the experimentation and validation needs of research on next generation Internet design proposals. Two basic ideas are those of virtualization and federation. Virtualization provides isolation and sharing of substrate experimental resources including routers, switches, and end-hosts. Federation provides both realistic and large scale testing environments through federation of multiple diverse testbeds designed to represent diverse contexts.

2. Scope

This paper does not claim to present an exhaustive survey of all of the research efforts that are presently underway, in the area of next generation Internet design. It is, at best, a comprehensive coverage of relevant next generation networking research. It should also be noted that, unlike conventional surveys, we refrain from passing judgmental remarks (except with some reference to historic perspectives) or establishing any form of conventional wisdom, due to the lack of concrete results at this very early stage of research on next generation network architectures. Likewise, this paper presents a broad perspective of the wide spectrum of highly diversified research efforts in this area rather than biasing any particular approach. We expect that this survey will need to be followed up by future surveys with much narrower perspectives when research in these areas reach the required levels of maturity. Another point to note is that there are many references in this paper to work that is neither recent nor particularly next generation. Our claim is that although research efforts to define a next generation Internet architecture are "officially" fairly recent, almost all proposals in this area are extensions of established knowledge from past research efforts. Thus, both past and present research efforts that we feel will impact future Internet research in any significant way have been included in this survey.

3. Security

The original Internet was designed in a trust-all operating environment of universities and research laboratories. However, this assumption has long since been invalidated with the commercialization of the Internet. Security has become one of the most important areas in Internet research. With more and more businesses online and a plethora of applications finding new uses for the Internet, security is surely going to be a major concern for the next generation. In the next generation Internet, security will be a part of the architecture rather than being overlaid on top of the original architecture, as in the current Internet. Years of experience in security research has now established the fact that security is not a singular function of any particular layer of the protocol stack, but is a combined responsibility of every principal communication function that participates in the overall communication process. In this section, we present several next generation proposals that address the problem of security from a different angle. This includes the security policies, trust relationships, names, identities, cryptography, anti-spam, anti-attacks, and privacy.

3.1. Relationship-Oriented Networking

The basic goal of the Relationship-Oriented Networking project [5] is to build a network architecture that makes use of secure cryptographic identities to establish relationships among people, entities, and organizations in the Internet. It tries to provide better security, usability, and trust in the system, and allow different users and institutions to build trust relationships within networks similar to those in the real world.

Relationship-Oriented Networking will mainly:

1. Consider how to pervasively incorporate cryptographic identities into the future network architecture.
2. Use these strong identities to establish relationships as first-class citizens within the architecture.
3. Develop an architectural framework and its constituent components that allows users and institutions to build trust relationships within the context of digital communications. These can be viewed and utilized in a similar fashion to relationships outside the realm of digital communications.

3.1.1. Identities

The traditional Internet uses unique names to identify various resources. These names can be email addresses, account names or instant messaging IDs. For example, we use the email address "user@organization.com" as the identifier for the email service. However, these identities offer little security since they can be easily spoofed. Moreover, they are invalidated after a change of service providers. In Relationship-Oriented Networking, these problems are solved by cryptographic identities that are used throughout the architecture. These identities are more secure than the plain, name-based schemes because security features are integrated in the form of keys or certificates.

3.1.2. Building and sharing relationships

The Relationship-Oriented Network architecture permits relationships to be established implicitly or explicitly. Allman et al. [5] provide an example in support of this requirement. For sensitive applications with tight access control, such as banking, the relationship between a bank and a patron, and the patron with their account, would need explicit configuration. In comparison, less sensitive services may be able to rely on less formal opportunistic relationships. For example, a public enterprise printer may not need tight access control, and the relationship may be opportunistic and less formal. The relationship between people can also be built implicitly or explicitly. As with trust relationship formations in our society, the relationship can also be setup by "user introductions." Also, the sharing of a relationship among different people or entities is allowed, which represents some degree of transitivity in the relationship. Moreover, the relationship can also be leveraged as a vote of confidence when trying to decide whether an unknown service provider is legitimate or malicious. Thus, the sharing of the relationship should be limited by the potential downside and privacy implications.

3.1.3. Relationship applications

Access control is one of the relationship applications. It spans from low-level access controls on the physical network infrastructure to high-level, application specific control. The first level of enhanced access control comes from having stronger notions of identity due to the adoption of cryptographic-based schemes. Thus, access control can be implemented based on the users or the actors rather than on rough approximations, such as MAC addresses, IP addresses, and DNS names. Typical examples are "Allow the employee in the human resource department to access the disk share that holds the personnel files" and "Allow Bob, Jane, and Alice access to the shared music on my laptop."

Relationships can also be used for service validation. In practice, users need to know that they are communicating with the expected service provider and not a malicious attacker.

Relationship oriented networking also tries to build a naming system that follows the social graph to an alias resource. The resource with a name can also be aliased in a context-sensitive manner by the users. Users can expose their name to the social networks which in turn provides ways to share information. For example, the name "babysitter" can be set in the personal namespace and expose the resource to a friend who is in need of child care. The name will be mapped to the unique email address of a babysitter.

In summary, relationships are a very important component of security, identity, and policy enforcement. Research in relationship oriented networking is expected to be of significant use for the future Internet. However, it is not trivial, as multi-layer relationships can be extremely complex and spawn many other issues such as security, identity and naming, service, and access control policies. Nevertheless, research in this area is expected to result in deeper insights into the nature of relationships and the complexities of constructing security models around them.

3.2. Security architecture for Networked Enterprises (SANE)

The SANE architecture [23] is designed to enhance security. The basic idea is to develop a clean-slate security architecture to protect against malicious network attacks. SANE achieves this goal by requiring all network traffic to explicitly signal their origin and their intent to the network at the outset.

With this design goal in mind, SANE includes a tailored security architecture for private networks (enterprise network) with tight policy control. It does this by using a domain controller to control the network-wide policies at a single location. For public settings, the SANE architecture requires the end-host APIs to be changed to allow the end-hosts to signal their intent to the large scale Internet.

The SANE architecture implements the network-wide policies in a central domain controller. It is claimed by Boneh et al. [23] that such a centralized solution prevents inconsistencies in network security policies by separating them from the underlying network topology. A default-off mode is also enforced in the SANE architecture, which means that any host must get permission before they can talk to other hosts. Any unauthorized transmission is disallowed at the source. Network entities are granted access to a minimum set of resources, and the information about network structure and connectivity is hidden from the end-hosts. Precise control over traffic is also implemented in SANE. SANE decides the exact paths to be taken by the network traffic. The source routes are also encrypted, which helps integrate middle-boxes and application-level proxies without sacrificing security.

As shown in Fig. 1, hosts are only allowed to communicate with the domain controller by default. In Step 0, a client sets up a secure channel with the domain controller for future communication through authentication. In Step 1, server B publishes its service, “B.http”, to the network service directory. In Step 2, before talking to client B, client A must obtain a capability for the service. In Step 3, client A prepends the returned capability on all the packets to the correspondent. SANE offers a single protection layer for the private networks, which resides between the Ethernet and the IP layer. Note that all of the network policies are defined and granted at the domain controller.

One possible issue with SANE could be the central control strategy, which introduces single point of failure, single point of attack, and scalability problems into the architecture. There are also some additional issues that need to be solved. For example, SANE requires the switches to perform per-packet cryptographic operations to decrypt the source route. This requires modifications and redesign of the switches and may slow down the data plane. Moreover, the end-hosts also need to be modified to address the malicious attacks. Mechanisms to integrate the middle-box and proxies into the SANE architecture pose important research challenges. More detailed mechanisms and designs to address these challenges need to be presented and validated before they can be applied to the real world.

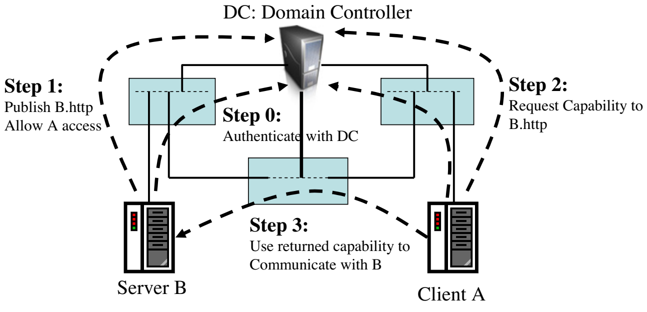

Fig. 1. SANE model.

3.3. Enabling defense and deterrence through private attribution

Current network security depends mainly on defenses that are mechanisms that could impede any malicious activity. However, deterrence is also necessary to reduce the threat and attacks in the Internet. Thus, there is a requirement for a balance between defense and deterrence in the future Internet. Deterrence is usually achieved by making use of an attribution that is the combination of an individual and an action. However, compared to the physical world, it is much more difficult to gain such an attribution in the Internet.

Two main design goals of this research project [192] are preserving privacy and per-packet attribution. Moreover, the security architecture provides content-based privacy assurance and tries to avoid any private information from leaking across the network. This proposal requires every packet to be self-identifying. Each packet is tagged with a unique, non-forgeable label identifying the source host. The private attribution based on group signatures allows the network elements to verify that a packet was sent by a member of a given group. Through the participation of a set of trusted authorities, the privacy of the individual senders can be ensured.

The per-packet attribution and the privacy preservation ensure that all of the packets are authenticated and traceable. This reduces potential attacks and offers deterrence to some extent, while at the same time maintaining sender privacy by the use of a shared-secret key mechanism.

Some of the challenges that need to be addressed are: (1) decisions that determine the source of the traffic in situations where traffic may be relayed by an intermediate host on behalf of the source host, (2) tradeoff between the need for attribution security and the user’s privacy, and (3) technical details for packet transformation, overhead reduction, and guaranteeing minimum changes and impact on the existing software.

3.4. Protecting user privacy in a network with ubiquitous computing devices

Ubiquitous presence and the use of wireless computing devices have magnified privacy concerns [185]. These concerns are inherent to the design of the link-layer and lower layer protocols and are not well addressed by the currently available approaches. In the next generation Internet, proliferation of these wireless computing devices is expected to worsen the issue of privacy.

The central problem is to devise a mechanism that conceals the end-point’s information from all parties that do not need to know it for the network to function. For example, IP addresses only need to be revealed to the immediate provider, not to all providers along the network path. It is assumed that sources trust their immediate provider more than any other transit provider on the network path. In this way, the user’s privacy can be guaranteed as well as be both manageable and accountable.

In this proposal, encrypted addresses are used to provide privacy. Entire packets, including their addresses, can be encrypted over links, hiding identities from other users of the network.

Re-routing is also avoided in this architecture, maintaining efficiently across the whole path.

For service discovery, cryptographic mechanisms can also be explored to protect the privacy of searches, responses, and beacons. However due to different privacy requirements, this issue is difficult to resolve. An effort is made to develop a single mechanism that can transition between these different classes of networks. Also, methods to allow the client devices to privately discover a service even when they are connected to an un-trusted provider are explored.

In addition to host addresses and network names, some other "implicit identifiers" can also leak information about the user's identity. The project proposes to commit in-depth research on defining communication privacy in human terms, since privacy is ultimately about humans and cannot be delivered by the network without human interaction. Thus, privacy scenarios and policies need to be explicitly presented to the users in order to keep them informed, and the users need to be able to dictate their policy requirements as necessary.

There are several unavoidable challenges facing these design goals. First, names and addresses should be designed to conceal identity instead of leaking them. However, identity cannot be concealed completely since some information needs to be understood by the network devices in order to accomplish certain functions. Thus, the names and addresses need to be designed carefully to conceal important information from the un-trusted parties and to reveal proper information to the authorized or trusted parties. Also, broadcast links, such as wireless networks, have different requirements than wired network paths. Moreover, different layers may have bindings of the names and addresses and the identities may be revealed in multiple levels. Thus, an additional requirement is to ensure that an identity is revealed only after it is known that the binding is authorized. This new requirement forces major changes to the design in the current Internet. Managing information exposure of implicit names and identifiers are some of the major design challenges that need to be addressed.

3.5. Pervasive and trustworthy network and service infrastructures

"Trustworthy networks and service infrastructure" [39] is the name of the European Union's Framework Program 7 (FP7) research plan on security for the future Internet. This is an umbrella project for security specific research consisting of many projects researching different aspects of network security. There are four main goals:

1. trustworthy network infrastructure;
2. trustworthy service infrastructure;
3. technologies and tools for trustworthy networks;
4. networking, coordination, and support.

Most of these projects are still in their initial phases, so only initial proposals and task goals are available at this point.

The trustworthy network infrastructure research is dedicated to finding new architecture designs for future heterogeneous networks and systems. These are designed with built-in security, reliability, and privacy, with secure policies across multiple domains and networks, and with trustworthy operation and management of billions of devices or "things" connected to the Internet. It also includes the research and development of trustworthy platforms for monitoring and managing malicious network threats across multiple domains or organizations. Typical E.U. FP7 projects on this topic include ECRYPT II [44] (on future encryption technologies), INTERSECTION [70] (on the vulnerabilities at the interaction point of different service providers), AWISSENET [13] (on security and error resilience on wireless ad hoc networks and sensor networks), and SWIFT [196] (on future cross-layer identity management framework).

The second research area is to develop a secure service architecture. Secure and trustworthy service architectures are an immediate requirement to support the huge growth of Internet business applications and services. Thus, a strong need for service security properties such as reliability, availability, trust, data integrity, information confidentiality, and resilience to faults or malicious attacks is identified. To meet the various requirements, new advances in the fields of secure software engineering, modeling, and languages and tools need to be achieved. Two important research goals of this effort are the specification and validation of security properties of the service architecture and platforms and the technologies for managing and ensuring security levels in different environments. Typical projects under this research topic include MASTER [95] (managing and auditing using secure indicators), TAS3 [197] (trusted Service-Oriented Architecture based on user-controlled data management policies), and AVANTSSAR [11] (specifying and validating the trust and security properties of service).

Technologies and tools for trustworthy network research include pro-active protection from threats in future networks with a high volume of network entities, user-centric privacy and identity management, and management and assurance of security and integrity. Typical E.U. projects on this topic include MOBIO [102] (on biometric technologies), ECRYPT II [44] (on cryptology), TECOM [198] (on trustable systems), and SHIELDS [189] (secure software engineering technologies).

3.6. Anti-Spam Research Group (ASRG)

There has been a substantial increase in the number of problematic e-mails, which are generally called spam. At an extreme point, spam could threaten the usability of the e-mail service. The situation is already quite severe and is getting worse.

ASRG [10] is a working group of the Internet Research Task Force (IRTF), which is focusing on research on anti-spam technologies. ASRG investigates tools and techniques to mitigate the effects of spam. Using the underlying basic characteristics of spam, this group is motivated to develop solutions and approaches that can be designed, deployed, and used in the short-term. The related work areas include new administrative tools and techniques, improved anti-spam tools and techniques, evaluation frameworks and measurement, and approaches that involve changes to the existing applications and protocols.

In the past decade, many anti-spam technologies have been invented to tackle the current challenges in anti-spam [9]. Typical examples include:

1. Message Content Techniques: This is a basic anti-spam technique and includes three categories: static filtering, adaptive filtering, and URL filtering. Static filtering filters the spam by setting the static addresses or subject keywords. Adaptive filtering is relatively advanced in that it can adjust the filtering based on experience. A typical example is Bayesian filters. URL filtering is based on the fact that spam always contains redirecting URLs to certain websites. Software can be used to extract the URLs from the body of a message and check them against a blacklist. Since URLs in the spam change so frequently, it is a hard task to maintain this blacklist and a lot of spam traps are required to collect spam.
2. Techniques based on Simple Mail Transfer Protocol (SMTP): Another category of anti-spam techniques that makes use of the SMTP protocol. It includes timing and protocol defects techniques, greylist, callbacks, and rate limits techniques. Timing and protocol defects techniques detect extra data in the input

buffer prior to the server sending the HELO/EHLO response, thus reducing the spread of spam. Greylist is effective against those spammers who use cracked PC to send spam but ineffective against spammers sending from conventional machines. The greylist technique attempts to detect SMTP clients that are not true SMTP servers and do not maintain a message queue. It does this by initially deferring the incoming messages and giving a 4xx (temporary failure) response during the SMTP protocol dialog. The callbacks technique is relatively inefficient because spammers can easily escape this mechanism. The basic idea behind the rate limits technique is that the robot spammers always send bursts of messages faster than humans and legitimate mail servers. Thus, an SMTP server can count the number of connections per client over a time window. It is obvious that the rate limiting technique is ineffective as a real anti-spam solution; however, it is very effective against DoS (denial of service) spam.

3. Address management: Address management techniques include tagged addresses, code words, disposal addresses, and DNS (domain name system) blacklists. Tagged addresses and code words are similar in that they add a second part to an existing address that is used for sorting or filtering mail instead of mail routing. A disposable address is an address that can be disabled when spam comes. A disposable address is used in situations where users need to receive emails from unknown entities that may send spam in the future. Thus, the disposable email address is revealed rather than the real email address, which remains hidden from the attacks of spam bots.
4. Network techniques: Network techniques include DNS blacklists, DNS validation, and HELO/EHLO pattern matching. DNS blacklists are lists of IP addresses that share an undesirable characteristic, such as a history of sending spam. DNS validation techniques verify the SMTP client by comparing the proper DNS records related to it. HELO/EHLO pattern matching techniques look for strings with a high likelihood of being a spam sender and a low likelihood of being a legitimate organization or user.
5. White-list techniques: White-list techniques are typically achieved by recognizing known correspondents and adding them to a whitelist. The disadvantage is that it requires users to manually maintain the list.

4. Content distribution mechanisms

The content distribution mechanisms of the Internet have evolved from centralized server based distribution mechanisms to the more modern distributed approaches of Content Distribution Networks (CDNs) and Peer-to-Peer (P2P) networks. The popularity of the web, high quality content creation for dissemination, and the increased bandwidth provisioned at the network edge can be credited to this evolution. In this section, we will retrace this evolution and motivate the need for future Internet research in content delivery mechanisms. We will also introduce some innovative proposals that are being studied to define the future of content delivery on the Internet.

4.1. Next generation CDN

Initially, the concept of CDNs was introduced to mitigate the load on central servers. Central servers could offload the responsibility of delivering high bandwidth content to CDNs. For example, a web page downloaded from abc.com would contain pictures, videos, audio, and other such high bandwidth multimedia content. The central server at abc.com would serve only the basic web page while redirecting the browser to a CDN to fetch all of the multimedia content. This mechanism worked, since CDN servers were networked and placed strategically in the core network and were provisioned with high bandwidth links. CDNs moved the content closer to the end-user, ensuring less delay. However, measurements of content distribution data show that only 50% of Internet traffic is served from the top 35 core networks [88]. The rest of the data distribution has a long tail and spread across 13,000 network sites in the current Internet. As a result, the present state-of-the-art CDNs still suffer from the "Fat File Paradox" [143,88].

Since data travel on the Internet at almost the speed of light, it might seem that the distance between the source and the destination should not matter, hence a paradox. However, it turns out that even in the absence of congestion, the "middle mile" encounters delay as a result of peering problems between transit ISP's, DoS attacks, link failures, etc. Congestion in the intermediate routers worsens the problem. Also, neither the servers nor the clients have any control over the "middle mile." This is the "Fat File Paradox," which states that "it is the length of the pipe rather than its width that determines how fast a large file can travel through it" [143].

It is projected that with high quality content, such as high definition television, soon making its way to the Internet, the Internet would need to provision a bandwidth of 100 TB/s in the near future [88]. The "middle mile problem" discussed above will become more pronounced in the presence of such high data volumes. To mitigate this, a new solution for highly distributed CDNs has been proposed. These highly distributed CDNs place servers at the edge networks, thus abolishing the "middle mile" completely. However, these CDNs still suffer from the limitation of being able to serve only cacheable content. Also, highly distributed architectures come at the cost of increased security, management, scalability and synchronization problems. Thus, future CDN research shall involve addressing these challenges to mitigate the enormous growth of content distributed over the future Internet.

4.2. Next generation P2P

Another paradigm of data distribution that has evolved over the years is P2P networks. Initially born as the simple music sharing application Napster [144,223], P2P networks have progressed tremendously and are responsible for much of the Internet traffic today [179,16]. The key idea is that peers (or end-hosts) share content among themselves, thus abolishing the need for a central server. In doing so, peers act as "servents" (servers when uploading data for other peers or clients when downloading data from peers). An extensive survey on P2P networks can be found in [6].

The self-organizing and self-healing properties of P2P networks have the potential to become the predominant content distribution mechanism of the future Internet. However, there has been a declining trend in the popularity of P2P networks over the past year or so, due to the advances in streaming video technologies [145,24]. The reason for this decline may be attributed to certain fundamental problems underlying the basic mechanisms of P2P networks.

The first problem is that bandwidth provisioning to end-hosts at edge networks is generally asymmetric. The download bandwidth is far higher than the upload bandwidths. This leads to instability when the number of peer-clients for a particular content far outnumber the peer-servers.

The second problem is related to the dynamics of sharing. Selfish behavior is common in peers, wherein the peers want to act only as clients and never as servers. Incentive based mechanisms controlling the download bandwidth available to a peer depending on its upload bandwidth have been devised in modern P2P networks such as BitTorrent [146].

Finally, the third problem is the tussle of interests between P2P networks and ISPs. P2P networks form an overlay network of peers oblivious to the underlying IP network topology. This results in data dissemination among P2P peers such that they may contradict the traffic engineering policies of the underlying provider IP networks. This leads to a selection of more expensive routes, endangering peering policies between ISPs. The P4P [147] group is investigating methods for the beneficial co-existence of P2P networks and ISPs [216,217]. One possible solution is to develop P2P mechanisms that are aware of the underlying topology and location of peers [212]. An oracle mechanism wherein the ISPs assist the P2P networks in selecting peers has been described in [1].

P2P could be useful in serving as the next generation content delivery mechanism, mostly because of its scalability, resilience, self-configuration, and self-healing properties. Research groups, such as P2P-Next [148], are working towards solutions for topology-aware P2P, carrying legal and licensed content for media channels such as IPTV and video on demand. We think, these research efforts are important for P2P networks to alleviate the huge data dissemination needs of the future Internet.

4.3. Swarming architecture

"Uswarm" [207]proposes a data dissemination architecture for the future Internet based on some established techniques of the P2P world. A "swarm" (as used in the context of P2P systems) is a set of loosely connected hosts that act in a selfish and highly decentralized manner to provide local and system level robustness through active adaptation. BitTorrent is an extremely successful "swarming" P2P system. BitTorrent solves the traditional P2P problems of "leeching" (clients downloading files and not sharing it with other peers) and low upload capacity of peers. To counter leeching, Bittorrent employs a tit-for-tat mechanism wherein the download speed of a peer is dependent on the amount of data it shares. Also, BitTorrent implements a multi-point-to-point mechanism wherein a file is downloaded in pieces from multiple locations, thus ensuring that the download capacity of a peer is generally much higher than the upload capacity.

However, it is argued [207] that although BitTorent solves the problem of flash crowds (sudden high popularity of a piece of content) through its swarming model, it does not have good support for a post-popularity download when only a few seeds for the content may exist and the demand for the content is not very high. Also, BitTorrent uses a centralized architecture for its tracker which introduces a single point of failure. Thus, in scenarios such as delay tolerant networks (DTN), if the tracker is unreachable from the peer, then the peer cannot download data even though all the peers uploading the file may be within communication reach of the DTN peer. The mechanisms introduced to counter this situation are the use of replicated trackers or Distributed Hash Tree (DHT) tracking mechanisms. However, replicated trackers result in un-unified swarms (multiple swarms for a single file), while DHT mechanisms introduce additional latency and burden on the peers.

Despite some of these drawbacks, as of 2004, BitTorrent was reported to be carrying one-third of the total Internet traffic [142,199]. Motivated by the huge success of swarming systems such as BitTorrent, "Uswarm" [207] proposes to investigate the feasibility of a swarming architecture as the basis for content delivery in the future Internet. Some of the key modifications needed to define an architecture based on swarming rather than an isolated service are: (1) a generic naming and resolution service, (2) a massively distributed tracking system, (3) economic and social incentive models, and (4) support for in-network caches to be a part of the swarm architecture.

Uswarm needs to devise a generic naming and resolution mechanism to be the basis for content distribution architecture. The objective of this mechanism called the Intent Resolution Service (IRS) is to translate the intent specified in an application specific form (URL, CSS, etc.) to a standardized meta-data, and resolving the meta-data (Meta-data Resolution Service or MRS) to a set of peers that can serve the data. The MRS service is devised using a combination of highly replicated tracking using logically centralized tracking system (such as DNS), in-network tracking where a gateway may intercept the request and process it, and peer-to-peer tracking using peer-to-peer gossip mechanisms (as in KaZaa [149], Gnutella [150], etc.). All of these tracking mechanisms are highly distributed and are expected to significantly improve the availability of the system.

Uswarm is a unified swarming model. Unlike models similar to BitTorrent where each file is associated with its own swarm, uswarm advocates a unified swarm. In a unified swarm, peers are not connected loosely together based on a particular content, rather they are all part of the system and help each other attain their objectives. For example, suppose there are two files, A and B, each with their associated swarm, A_swarm and B_swarm, respectively. Also suppose that the peers of B_swarm already have the file A and similarly the peers of A_swarm already have the file B. In such a situation, A_swarm could contribute to B_swarm by providing a pre-formed swarm for file B and vice versa.

The co-operative swarming mechanism requires some fundamental extensions to incentive mechanisms similar to BotTorrent. Uswarm uses the same "tit-for-tat" principle of the BitTorrent incentive mechanism but also provides incentive for a peer to upload blocks from multiple files (rather than only the file that it is presently downloading) to support the co-operative swarming paradigm of uswarm. A control plane incentive mechanism also needs to be developed for uswarm since it depends on a distributed P2P mechanism for control messages for the MRS. The control plane incentive mechanism includes tit-for-tat (keeping track of peers that are most helpful for resolving control messages), and dynamic topology adaptation (in which peers select their neighbors dynamically based on how helpful they are).

Uswarm looks to solve some very relevant problems of P2P networks. Menasche et al. [99] have presented a generalized model to quantify the availability of content in swarming systems such as BitTorrent. This supplements previous studies on the robustness, performance, and availability of swarming systems similar to BitTorrent [103,173] and is expected to advance the feasibility analysis of such systems as a candidate data dissmination mechanism for the next generation Internet. Research in this area addresses some general issues relevant to other research areas as well. For example, leveraging in-network caches, uswarm addresses some of the concerns of the P2P-ISP tussle and also has some similarities to the Content Centric Networking architecture mechanisms discussed in Section 4.4.

4.4. Content Centric Networking

Although classified as a content delivery mechanism, Content Centric Networking (CCN) [30–34,75,76,79] offers much more than that. It proposes a paradigm shift from the traditional host centric design of the current Internet to a content centric view of the future Internet. CCN is motivated by the fact that the Internet was designed around 40 years ago and has lost its relevance in the present context of its use. While designed originally as a mechanism to share distributed resources (for example, access to a printer attached to a single remote host in the organization), today the Internet is used more for content delivery. Since resource access and data access are fundamentally different with completely different properties, the Internet needs to be re-designed to accommodate the present context. Although, the ideas of a content centric network have existed for quite some time through a series of papers on this topic at the University of Colorado [30–34], it has gained momentum only recently in the context of the next generation Internet design initiatives [75,76,79]. In this subsection we shall discuss the specifics of two of the most recent efforts in this area namely Networking Named Content (NNC) [75,76] and Data Oriented Network Architecture [79].

NNC is based on the observation that it does not really matter most of the time where data comes from as long as it is valid, secure, and authentic. The idea of NNC is to design a distribution mechanism as an overlay above the IP networks (at least in the first phase of NNC deployment), leveraging the low cost of persistent storage. Data has the property that it is replicable. Also, data may be cached at various points in the network. Popular content dissemination on the current Internet involves millions of unicast copies of the same content to be distributed end-to-end. Though serving duplicate copies of the same content, the routers are neither designed nor have an incentive to cache the content and serve from a local copy whenever a request for the same content is encountered. The primary motivation for ISPs to deploy NNC is the cost savings from not having to pay transit fees to provider ISPs for content that is requested by multiple users within a time window. Users gain by higher perceived quality of service since content is now cached and served from a nearer location. Content providers gain in terms of lower CDN bills and higher user satisfaction.

NNC describes a scenario where special intermediate ISP routers (NNC nodes) cache content, clients request content (interest packets) is broadcast in a controlled manner, and intermediate nodes that have incentive to serve the content from their caches and receive a request for the content may serve the content from their local storage. The NNC node maintains three tables: the "Content Store" (CS), the "Pending Interest Table" (PIT), and the "Forwarding Information Base" (FIB). When an NNC node receives an "interest packet," it first matches the content name to the CS. If the content is found in the CS, then a data packet is served that consumes the interest packet. If the content is not found in the content store, it is matched against the PIT to check whether it is already waiting on another request for the same content. If a match is found in the PIT, then the NNC node appends the interface on which the new interest packet arrived. When the data packet that consumes the interest arrives at the node, it is replicated and sent out on all the interfaces that have an entry for the content in the PIT. If the content name matches neither the CS nor the PIT, then the FIB is referenced to determine the interface on which the interest packet should be forwarded. Also, an entry is appended to the PIT for the forwarded interest packet. Thus, the core NNC architecture is designed around named content instead of location. For a detailed discussion on the NNC naming convention, see [76].

The other proposal that we will discuss is DONA. DONA shares similar views to those of NNC but is fundamentally different in its implementation ideas.

The Data Oriented Network Architecture (DONA) [79] proposes a clean-slate architectural idea similar to NNC. Both DONA and NNC advocate a paradigm shift from the present host centric architecture of the Internet to a data centric architecture. NNC proposes a network-wide caching mechanism at various network nodes, leveraging the dipping cost of persistent storage and defining an efficient content dissemination system as an overlay over the present IP networks. DONA on the other hand emphasizes a novel mechanism for the naming of content and name resolution to build an architecture around service and data access.

According to Koponen et al. [79], the three most desirable properties for data and service access are: 1. Persistence – the name of a service or data object remains valid as long as the service or data are available, 2. Availability – data or service should have a high degree of reliability and acceptable latency, 3. Authenticity – data can be verified to have come from a particular source. Unfortunately, in the present host centric design of the Internet, these three basic requirements of data and service access are not provided naturally. The current design defines mechanisms to access particular hosts, implicitly limiting data to a host. DONA proposes a novel mechanism of explicitly naming the data or service and routing on these names for data or service access.

The key mechanism in DONA involves the explicit naming of the data or service around a principal (the owner/creator of the data or service). The names are of the form P:L, where "P" is the cryptographic hash of the principals public key and "L" is a label for the data/service chosen by the principal. The next step of mapping the data/service name to a location is done through a routing on name mechanism. The routing structure is composed of entities called routing handlers (RHs) which are responsible for routing data names (P:L) to particular data servers. A data server may be any host that has a copy of the data and is entitled to serve it.

Two basic primitives "FIND" and "REGISTER" are defined. Any host entitled to serve data P:L may register it with its local RH in the same autonomous system (AS). The local RH advertises it to the RHs in the neighboring ASs following the AS level routing policies of the Border Gateway Protocol (BGP). A client seeking access to data sends out a FIND (P:L). The FIND message is routed across the RHs, until the nearest copy of the data is found. FIND also initiates a transport level connection. In the case where RHs cache data, data exchange starts between the client and the RH. Otherwise, after the FIND has been resolved to a particular host, a direct IP level exchange between the client and the server is initiated.

Routing has the desirable property of finding the shortest or the most optimal path and also routing around failures. Thus, by routing on data names, DONA achieves the same reliability and self-healing properties in the context of data access that the current Internet has for host access. Flat cryptographic names associated with principals help authenticate data validity and data source. Also, the DONA mechanism of late binding of data to the server host achieves persistence of data (data is available as long as it exists) and thus freeze its dependency from the persistence of the host.

NNC and DONA define the whole architecture of the future Internet around data delivery. In the DONA context, other mechanisms such as P2P and CDNs will be special cases using the DONA primitives in different ways.

We have discussed some of the potential mechanisms that will contribute to content delivery services in the next generation Internet. Some of these mechanisms, such as CDN, may not be considered strictly next generation even with their extensions since they are not clean-slate. P2P mechanisms are already a dominant carrier of content in the current Internet and their incorporation into a systematic architectural design (as in uswarm, P2Pnext, and P4P) is expected to prepare it for the next generation. However, we believe that content in the Internet cannot be generically classified under a few common attributes, hence more than one of these mechanisms are expected to co-exist. This reiterates the requirement that the future Internet needs to support diversity even at the core architectural levels.

5. Challenged network environments

"Challenged network" [36,46,47] research focuses on heterogeneous network environments where continuous end-to-end connectivity cannot be assumed. Examples of such network environments are interplanetary networks, wireless sensor networks, wireless ad hoc networks, post-disaster networks, etc. Challenged network research is relevant to the discussion of future Internet architectures on two perspectives. Firstly, future Internet architectures may be able to borrow techniques developed in the challenged networks context to design more energy efficient protocols that strike a feasible tradeoff between performance and energy efficiency. Secondly, research in diversified network environments such as "challenged networks" is likely to collaborate and advance the future Internet requirement to federate diversified networking environments.

5.1. Delay Tolerant Networks (DTN)

Delay Tolerant Networks (DTN) is already an active area of research, guided mostly by the DTN working group at the Internet Research Task Force (IRTF) [152]. Developed initially as part of an effort to develop Interplanetary Internet (IPN) [151] for deep space communication, the scope of DTN was generalized to ‘‘address the architectural and protocol design principles arising from the need to provide interoperable communications with and among extreme and performance-challenged environments where continuous end-to-end connectivity cannot be assumed’’ [151]. Examples of such environments include spacecraft, military/tactical, some forms of disaster response, underwater, and some forms of ad hoc sensor/actuator networks. It may also include Internet connectivity in places where performance may suffer such as in developing parts of the world.

The key research contribution of DTN research has been the development of an ‘‘end-to-end message oriented overlay’’ [36] called the ‘‘bundle layer’’. The ‘‘bundle layer’’ is a shim-layer between the transport layer (or other) of the underlying network below it and the application layer above it. It implements the ‘‘bundle protocol’’ [183] that provides ‘‘store-forward’’ services (through management of persistent storage at intermediary nodes) to the application layer, to help cope with intermittent connectivity. It stores and forwards ‘‘bundles.’’ Bundles are the protocol data unit (PDU) of the ‘‘Bundle Protocol’’ and are variable-sized, (generally) long messages transformed from arbitrarily long application data, to aid in efficient scheduling and utilization of intermittent communication opportunities or ‘‘contacts’’.

In DTN networks, the end-to-end principle is re-defined for applicability to environments with intermittent end-to-end connectivity. Accordingly, the bundle protocol defines the mechanism of ‘‘custody transfer’’. When an intermediate DTN node N receives a bundle with custody transfer, and if it accepts custody of the bundle, then it assumes the responsibility for reliable delivery of the bundle. This allows the node that transfers the custody to node N to delete the bundle from its buffer. Such a notion of reliability is relevant in the DTN context as against the end-to-end principle since the source node may not be connected long enough to ensure end-to-end reliability. Recently, there have been criticisms of the bundle protocol about its efficacy in disrupted and error prone networks [214]. Several other features of the bundle protocol may be obtained in detail from the relevant RFCs [183,36,48,174,25].

Another important research issue is DTN routing [77]. Supposedly, ‘‘intermittent connectivity’’ seems to be the only common attribute of all DTN environments. Other than that, DTNs vary greatly on the parameters of delay, error, mobility, etc. Moreover, based on the nature of topological dynamicity, they can be re-classified into deterministic and stochastic systems. Various routing protocols specified for DTNs try to address routing in any one of these operating environments.

While routing in deterministic contexts is easier, an ‘‘epidemic routing’’ [172] scheme has been designed for routing in highly random conditions. In epidemic routing, a message received at a given intermediary node is forwarded to all nodes except the one on which the message arrived. Also, a relay based approach may be used in networks with a high degree of mobility. In the relay-based approach, if the route to the destination is not available, the node does a ‘‘controlled broadcast’’ of the message to its immediate neighbors. All nodes that receive this packet store it in their memory and enter a relaying mode. In the relaying mode, a node checks whether a routing entry for the destination exists and forwards the packet. If no paths exist and if the buffer at the node is not full, the packet is stored in the node’s buffer replacing any older copies of the packet already in the buffer. There are a plenty of routing protocols for delay-tolerant networks and [222] presents an exhaustive survey of the existing routing protocols and the context within which they are most suitable for operation.

While the DTN research discussed here is not strictly next generation, a basic understanding of the DTN architecture provides a more clearer perspective to architectures that derive from its fundamental concepts to apply to diverse heterogeneous challenged network environments.

5.2. Delay/fault tolerant mobile sensor networks (DFT-MSN)

Classical sensor networking research is generally focused on developing techniques to achieve high data throughput while minimizing power consumption. As a matter of fact, the radio module is one of the significant power consumers on the sensor node. Hence, a lot of energy efficiency mechanisms of sensor networks involve optimized use of the radio resource. A significant gain in power conservation can be achieved by turning the radio to sleep for most of the time, waking it up periodically to receive or send data. Such schemes can benefit from the store and forward methods developed for DTNs to handle communication over intermittently available links. SeNDT [154], DTN/SN [155], ad hoc seismic array developed at CENS [153] projects are some examples that employ this technique to attain higher power utilization on their sensor nodes.

Apart from DTN techniques to optimize power consumptions, DFT-MSNs represent actual scenarios where a DTN-like context is experienced. An example of such a scenario with node mobility, intermittent connectivity and delay and fault tolerant networking context of wireless sensor networks is presented in [69]. For applications such as environmental pollution monitoring using mobile sensors, conventional sensor network protocols do not suffice since they are designed to optimize throughput versus power consumption while assuming abundant bandwidth and deterministic and controlled connectivity. On the other hand, classical DTN networks represent the context of intermittent and opportunistic connectivity, high delay and error rates, but without much concern for power conservation. DFT-MSNs, thus, represent a new class of networks that resemble the context of DTNs with the additional constraints of optimizing power consumption.

A cross-layer protocol design for DFT-MSN communication is described in [211]. The idea is to design a data delivery protocol based on two parameters: (1) nodal delivery probability and (2) message fault tolerance. In the context of a network of sensors with random mobility patterns and hence intermittent delivery opportunities, the nodal delivery probability is a parameter that depends on the history of the node’s successful/unsuccessful transmission of data to another node that has a higher probability of forwarding the data towards the sink. Message fault tolerance is achieved by having multiple copies of the message in the buffers of various mobile nodes, thus having a high probability of getting at-least one copy to be eventually forwarded to the sink. To control the level of redundancy a fault tolerance degree (FTD) parameter for the message is calculated each time it is transmitted from one node to the other. FTD is zero when the message first originates and increases (thus losing priority) each time it is transmitted. The FTD serves as the parameter for data queue management at each node thus bounding the level of redundancy. Based on these parameters, the cross-layer protocol itself consists of two modes: (1) sleep mode – to conserve power and (2) work mode. The work mode has two phases: (1) asynchronous phase and (2) synchronous phase.

1. Asynchronous phase: This is similar to conventional asynchronous phase RTS/CTS (Request to send/clear to send) handshaking of the IEEE 802.11 protocol where the node wakes up from sleep, contends over the shared channel for a chance to transmit, sends an RTS message, waits for a CTS from the receiver

and finally starts transmitting the message in the synchronous mode. In DFT-MSN, the wireless nodes exchange the parameters of nodal delivery probability and available buffer space in the RTS/CTS exchange. These parameters are the basis of the nodes' decision process of whether to forward a message at the given opportunity that shall maximize the chances of the message reaching the sink and at the same time keeping redundancy under bounds.

2. Synchronous phase: In this phase, the data transmission is synchronized and hence there is no contention. After receiving the CTS from multiple nodes in the asynchronous phase, the node selects a subset of nodes fit for data forwarding and accordingly sends out a "schedule" for synchronized data dissemination.

Based on these phases, the protocol can be optimized to achieve a tradeoff between sleep time and link utilization. A simple scheme, proposed in [211], allows the node to sleep for a specific time "*T*", determined by two factors: (1) the number of successful transmissions in the last "*n*" working cycles and (2) available message buffer, enforcing short sleeping periods if buffer is full. This mechanism allows the nodes of a DFT-MSN to conserve their power and at the same time maximize the utility of communication opportunities.

Many networks of the future should benefit from the research of DFT-MSN as we move towards an energy efficient system design paradigm in all spheres of engineering. The research in this area is still not mature with only a few proposals and application areas defined as yet. However, owing to the context in which it operates, it is certainly going to add value to the efforts of future Internet designs.

5.3. Postcards from the edge

The cache-and-forward paradigm of delay/disruption tolerant network has been proposed by Yates et al. [220] to be developed as the basis for an independent network level service to accommodate the huge growth in wireless access technologies at the edge of the Internet. The key motivations for this project are:

1. Advances in wireless access technologies have spawned a huge growth in the number of wireless devices connected at the edge of the Internet. Most of these devices are mobile leading to intermittent connectivity due to factors such as failure of the radio path and contention for access. The original Internet was designed under the assumption of persistent end-to-end connected hosts and, thus, the TCP/IP protocols fail to accommodate such an operating environment.
2. Original routers were designed when storage at routers was expensive. The diminishing cost of memory makes architectures like store-forward (requiring persistent storage resources at the routers) more feasible today than before.
3. The future Internet is being designed to allow the coexistence of multiple architectures through virtualization. Thus, it is much easier for newer paradigms in networking architecture to be defined today, than ever before.

The key elements of the proposed architecture consist of wired backbone routers, access routers and mobile nodes. It is assumed that each of these network elements shall have considerable amount of persistent storage. The idea is to develop an architecture based on the store-forward paradigm of DTNs such that every mobile node is bound to a set of "postoffice" nodes. These postoffice nodes are responsible for caching the data on behalf of the mobile node during periods of disconnection and opportunistically deliver it when feasible, either directly or through a series of wireless hops.

The design space for the transport layer looks pretty similar to that in classical DTN networks in the sense that they deviate considerably from the end-to-end paradigm of conventional transport protocols of the Internet. Additionally, a naming protocol needs to be specified that maps a node to a set of postoffice nodes. The routing protocol to route packets to a wired cache and forward (CNF) node is similar to the Inter-AS and Intra-AS routing of the current Internet. CNFs belonging to the same AS exchange reachability information among themselves along with detailed path descriptions (link state, preferred modes of data reception, etc.) while Inter-AS routing involves exchange of just reachability path vector information.

However, defining an Internet-wide store and forward file delivery service has lots of additional challenges. A primary challenge would be that of security, with the file being cached at various nodes in the network. Two conceivable security threats are those of (1) unauthorized access to a file from the cache of an intermediate node and (2) DoS attacks on network elements by artificially filling up their storage. Also, congestion control mechanisms in ORBIT-like scenario become more important than in DTN scenarios because of the scale of operation of such a network and the finite memory. Another issue that we think might be relevant is that of controlled redundancy. A sound principle needs to be developed to control the number of copies of the file existing at the various intermediate nodes. This would have huge implications on the scalability of the system.

The proposed architecture could produce relevant research results that advance next generation Internet efforts in the general area of data centric networking paradigms. Such data centric network designs could benefit from a Internet-wide efficient caching and delivery system for data, which is one of the proposed research outcomes of this architecture.

5.4. Disaster day after networks (DAN)

A DTN-like challenged network scenario is encountered in disaster-day after networks (after a hurricane or a terrorist attack). An instance of a disaster-day after networks (DAN), Phoenix [92], proposes a novel architecture for "survivable networking in disaster scenarios.

The robust mechanism built-in into the original Internet was designed such that it could isolate troubled areas (congested routes, broken links, etc.) and ensure connectivity to the existing parts of the infrastructure. Phoenix [92] claims that such "fail-stop" robustness is not suitable in scenarios where disasters are expected to be of smaller scale, localized, partial or intermittent connectivity, heterogeneous contexts and severely limited resources. This proposal seeks to define a new architectural framework for providing communication support across diverse, mobile and wireless nodes, intermittently connected to each other, to cooperatively form a rescue and recovery communication service network under challenged conditions.

The two major design requirements for Phoenix are: (1) Role-based networking and (2) communication over heterogeneous devices. The networking paradigm in such situations is mostly host-service based rather than being host–host based. Role-based anycast routing mechanisms are best suited, both, for routing efficiency in such challenged conditions and contextual mapping of the services to the available resources. The main objective of Phoenix is to utilize all available resources for communication, power supply, etc. This motivated the design of an architecture that allows the co-existence of multiple heterogeneous communication devices.

Although inspired by the design of delay/disruption tolerant network (DTN), DANs present a new networking context as opposed to the classical networking contexts of DTNs. Since the topology in a DAN is extremely dynamic, traditional topology based naming of the Internet and DTN [77] routing are not appropriate. Most other classes of DTNs such as inter-planetary networks and rural connectivity networks have almost exact knowledge about available storage resources and mobility patterns. Such information is not available to DANs. Also, being a service-host paradigm and limited in topological and service diversity, DAN is able to optimize its routing using anycasting. Such role-based methods are generally not employed for traditional DANs. Apart from these, DANs also have to (1) deal with a higher degree of diversity in its underlying communication technology, (2) offer better optimizations in the use of redundancy for resilience, (3) better use resources such as storage and communication opportunities, (4) define a more stricter prioritization of traffic to ensure timely dissemination of critical life-saving data, (5) formulate incentive schemes for sharing personal resources for common good, and (6) define security mechanisms to protect against potential abuse of resources, compared to most classical DTN scenarios.

The architectural elements of Phoenix incorporate all available resources that include personal wireless devices such as cellular phones and home WLANs, external energy resources such as car batteries, wide-area broadcast channels, and dedicated short-range communication systems (DSRCs). They incorporate these resources into one cohesive host-service network and provide an unified communication channel for disaster recovery and rescue operations, till the original infrastructure for communication is re-instated. To achieve this convergence and the stated objectives of DANs in general, Phoenix relies on two underlying communication protocols: (1) The Phoenix Interconnectivity protocol (PIP) and (2) The Phoenix Transport Protocol (PTP).

1. Phoenix Interconnectivity Protocol (PIP): In a DAN scenario, the communication nodes are expected to be partitioned into a number of temporarily disconnected "clusters" and each cluster comprises of one or more "network segments" using different communication technologies. A multi-interface node supporting multiple access technologies can bridge two or more network segments. Also, node mobility, disaster recovery activities and topology changes may initiate connection between clusters. In Phoenix, the PIP layer provides role-based routing service between nodes belonging to connected clusters. Each node advertises its specific roles. The forwarding table of PIP maintains entries mapping routes to specific roles and an associated cost metric. Thus, PIP provides an abstract view of a fully connected cluster of nodes to the upper layers while managing all the heterogeneity of access technologies, role-based naming of nodes, and energy efficient neighbor and resource discovery mechanisms within itself. An energy-aware routing protocol for disaster scenarios has been more recently proposed by the same group [205].
2. Phoenix Transport Protocol (PTP): DAN operates in an environment of intermittent connectivity, like DTNs. Also, negotiation based control signaling to optimize bandwidth utilization is not possible in such scenarios. Thus, the Phoenix Transport Layer (PTP) is responsible for optimization of storage resources to guarantee eventual delivery of the message. This "store and forward" paradigm of Phoenix is pretty similar to DTNs except that in DANs like Phoenix, storage resources are highly constrained and congestion control issues are more important in DANs than in other types of DTNs. In an attempt to optimize storage resources at forwarding nodes, PTP follows strict prioritization in data forwarding during contact opportunities.

   To deliver data between PTP neighbors (logically connected nodes, similar to the concept of neighbors in the end-to-end paradigm) belonging to the same connected cluster, PIP routing may be used. However, for PTP neighbors in disconnected clusters, opportunistic dissemination techniques need to be used. PTP tries to optimize this dissemination process through "selective dissemination" – deciding what data to be given to whom to maximize the eventual delivery probability of the data. However, lack of pre-estimated knowledge about node mobility and capability makes it challenging for PTP to optimize selective dissemination. A mechanism of diffusion filters based on exchange of context information (neighbors encountered in a time window, current neighbors, degree of connectivity of nodes, etc.) between PTP peers has been suggested as a solution for such situations.

Other architectural considerations of Phoenix include those of security, role management, context sensing and localization, and accounting and anomaly detection issues.

Phoenix is, thus, an instantiation of a more general class of disaster Day After Networks (DAN), that is expected to use established concepts and techniques of DTNs and spawn an important research area for future networking research.

5.5. Selectively Connected Networking (SCN)

Most future system designs will need to be energy efficient. Networking systems are no exception. The original design of the Internet assumed an "always-on" mode for every architectural element of the system – routers, switches, end-hosts, etc. Sleep-modes defined in modern operating systems are capable of preserving the local state of the end-hosts, but not their network states. This incapability can be attributed to the design of the networking protocols. Most protocols implicitly assume the prolonged non-responsiveness from a particular end-host to be signs of a failure and thus discard all associated communication state with the end-host. Obviously, a new paradigm of energy efficient protocol design is required to design energy efficient networking systems.

Methods for developing a "selectively connected" energy efficient network architecture are proposed for study by Allman et al. [3,4]. Although not particularly similar to DTNs, research in designing selectively connected systems could benefit from the existing ideas in DTNs, particularly when sleep modes of end-hosts render an environment of intermittent connectivity. The key ideas in the design of selectively connected systems are: (1) Delegation of proxy-able state to assistants that help the end system to sleep, (2) policy specifications by the end system to be able to specify particular events for which it should be woken, (3) defining application primitives allowing the assistant to participate in the application (e.g., peer-to-peer searches) on behalf of the host and wake up the host only when required, and (4) Developing security mechanisms to prevent unauthorized access to the systems state from its patterns of communication.

The delegation of proxy-able state to the assistant and also delegating application responsibilities to it on behalf of the host bear some resemblance to the transfer of custody transfer mechanisms of DTNs. Nonetheless, custody transfer has the implication of defining a paradigm wherein end-to-end principle is not strictly adhered to while it seems that the assistant mechanism simply acts as a proxy for the host for control messages of distributed protocols (thus maintaining selective connectivity) and is authorized to wake up the host whenever actual end-to-end data communication is required. We believe that the design of assistants can be further extended using the concepts of custody transfer and store-and-forward networks such as DTNs.

6. Network monitoring and control architectures

The Internet has scaled extremely well. From its modest beginnings with a few hundreds of nodes, the current Internet has evolved into a massive distributed system consisting of millions of nodes geographically diversified across the whole globe. However, with the commercialization of the Internet, vested economic, political and social interests of the multiple ownership network model have added huge complexities to the elegance and simplicity of the distributed algorithms that were not designed for such a context. As a result, management of this massively distributed, multi-ownership network is significantly more complex than the initial single owner, all trusted network of a few hundred nodes. Thus, with the scale-up of the Internet, both in size and complexity, the need for a separate management plane aiding in autonomic network design and management functions is being increasingly felt.

Another design weakness of the current Internet is that the present management and control plane ride on the data plane. This creates: (1) security concerns wherein any misbehaving or compromised entity may send out unauthorized management or control packets and jeopardize any network function, (2) bootstrapping problem wherein the network cannot self-configure itself, thus depending on manual configurations for initial boot up of the network, and (3) Poor failure mode operation [65] wherein the management protocols are un-available when they are most required – during failures.

In this section we discuss some of the clean-slate architectural ideas that have been proposed to alleviate the above anomalies in the current Internet architecture. Also, some novel proposals aiding network trouble shooting and debugging are also discussed.

6.1. 4D architecture

The 4D architecture [156,219,64,178,65] presents a complete grounds-up re-design of the Internet management and control planes. It proposes the paradigm shift from the current "box-centric" [65] management and control to a completely centralized solution. The 4D architecture mostly addresses the routing related management issues and those that apply to management and control within an autonomous system.

Every autonomous system (AS) of the Internet is bound by some common local policies. Most of these policies are related to routing and access related functions. However, such centralized policies have to be translated to "box-level" [65] policies, wherein a box may be a host, internal router, border router or other network entity within the AS. These "box-level" policies have to be deployed individually (and hence a "box-centric" approach) such that they aggregate and implement the network-wide policy of the AS. The proponents of the 4D architecture claim that disadvantages of such an approach are:

1. Manual configurations at each network entity are error prone and complex. Also, manual configurations do not scale well for large networks.
2. The routing protocols are not designed to comprehend any policy language. The only way to implement a policy is by changing the input parameters (such as local preference, link weights, and DER) of the protocols to drive a desired output (Forwarding Information Base, etc.).
3. Changes in network topology (link failure, addition of a router, planned outages, etc.) require manual re-configurations in accordance with the new context.

Apart from these, network trouble shooting, debugging, problem isolation, etc. are extremely complicated for large enterprise networks and are additional motivations for the design of a more autonomic management framework for the next generation Internet. An interesting observation made by Yan et al. [219] regarding the state-of-art of management protocols in the current Internet is that problems in the data plane cannot be addressed through a management plane (when it is most required) because the management plane typically rides over the data plane itself. It is further observed that the lack of proper interface for cooperation of distributed algorithms, for example, between inter-domain and intra-domain routing protocols, leads to instabilities.

As an example from the original FIND proposal on the 4D architecture [156], Fig. 2 further illustrates the motivation. Fig. 2 presents a simple enterprise scenario, wherein AF1 and BF1 are the front office hosts of an enterprise while AD1 and BD1 are the data centers. The enterprise level policy allows front office hosts to access each other (AF1 may access BF1 and vice versa) but allows only local access for the data centers (AF1 can access AD1 and not BD1). To implement this policy, the routers at R1 and R3 place packet filters at the interfaces i1.1 and i3.1, respectively, to prevent any non-local packets to have access to the data center. Now, suppose a redundant or backup link is added between the routers R1 and R3. Such a small change requires positioning of additional packet filters at interfaces i1.2 and i3.2 of routers R1 and R3, respectively. However, such packet filters prevent the flow of packets between AF1 and BF1 through R2–R1–R3–R4, in case of failure of the link between R2 and R4, even though a backup route exists.

The four Ds of the 4D architecture are: data, discovery, dissemination and decision. These four planes are related to each other as shown in Fig. 3 to define a "centralized control" architecture based on "network-wide views" (view of the whole network) to be able to dictate "direct control" over the various distributed entities for meeting "network level objectives" of policy enforcements. The individual functions of each plane in the four dimensional structure are as follows:

1. Discovery plane: Responsible for automatic discovery of the network entities. Involves box-level discoveries – router characteristics, neighbor discovery, link layer discovery-link characteristics. The discovery plane is responsible for creating the "network level views."
2. Dissemination plane: Based on the discovery plane data a dissemination channel is created between each network node and the decision elements.
3. Decision plane: The centralized decision elements form the decision plane. This plane computes individual network entity state (e.g., routing tables for routers, etc.) based on the view of the whole network topology and network level policies to be enforced.
4. Data plane: The data plane is responsible for handling individual packets and process them according to the state that has been output by the decision plane. This state may be the routing tables, placement of packet filters, tunnel configurations, address translations, etc.

Thus, the 4D architecture sets up a separate dissemination channel for control and management activities through link layer

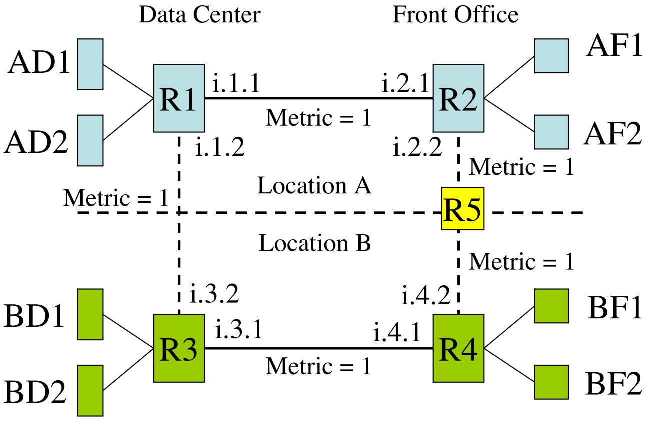

Fig. 2. A management mis-configuration scenario [156].

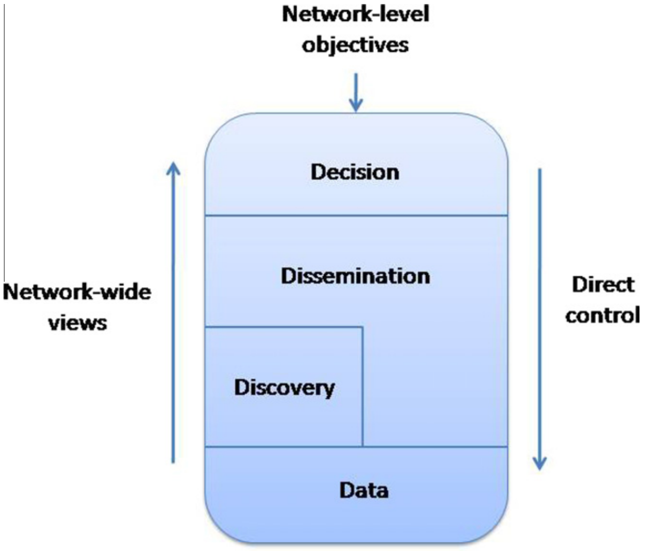

Fig. 3. 4D architecture [156].

self-discovery mechanisms. This gets rid of the management and control plane bootstrapping problems and makes a basis for auto or self-configurable networks. The centralized decision elements are responsible for implementing dynamic configurations based on topology information and organizational policy inputs. As an example, in the case study presented in Fig. 3, the change in the network topology as a result of the additional link between R1 and R3 is discovered by the discovery plane. The change is communicated to the decision plane through the dissemination channel. The decision plane re-evaluates the configuration of the network and places additional filters to conform to the organizational policies.

The ideas of centralized control and management have been here for some time. Feamster et al. [50] suggest a routing architecture wherein the routers act like forwarders while the computation of routing tables is done centrally. Also, the Routing Control Platform (RCP) [26] may be considered to be an implementation of some of the ideas of the 4D architecture. RCP proposes a similar idea of computing routing tables centrally based on data from border routers and eventually having two RCP enabled sites exchanging inter-domain routing information directly between the RCP servers.

The centralized solution though attractive may have some pitfalls in terms of scalability. An immediate scalability concern with respect to the 4D architecture is the discovery and dissemination plane. The discovery and dissemination plane depends on network-wide broadcasts. Broadcast mechanisms are essential for discovery mechanisms that do not depend on manual configuration. However, for large networks, a huge broadcast domain may pose to be bottleneck in performance. In this regard, the 4D architecture may borrow some ideas from [81], which implements an Ethernet architecture using DHT based lookup mechanism instead of network-wide flooding. A distributed control plane for the 4D architecture has been proposed by Iqbal et al. [72].

6.2. Complexity Oblivious Network Management (CONMan)

The CONMan architecture [55,15] is an extension of the 4D architecture. It re-uses the discovery and dissemination mechanisms of 4D and extends the 4D management channel to accommodate multiple decision elements or network managers. Each network manager in CONMan may be associated with particular network management tasks. In this regard, CONMan takes a more general outlook of management than 4D, not restricting it to just routing related management. Also, unlike 4D, CONMan does not present an extreme design point of completely doing away with distributed algorithms such as routing.

The motivations of CONMan are similar to those of 4D. The objectives of CONMan are: (1) self-configuration, (2) continual validation, (3) regular abstraction, and (4) declarative specification. Self-configuring networks are dynamic, adaptable and also less prone to errors because of reduced human intervention. Continual validation ensures that the networks configuration satisfies the stated objectives. Regular abstraction requires data plane distributed algorithms to implement a standardized abstract management interface through which they can be managed. Declarative specification is the ability to declare network objectives in high-level abstract terms and define an automated method to convert these high-level objectives to low-level implementations.

Based on the objectives stated above, CONMan implements an architecture based on discovery and dissemination planes, module abstractions and pipes. While the discovery and dissemination planes bear close resemblance to that of the 4D architectures, module abstractions are the primitive building blocks that implement a network-wide objective. Network wide objectives are modeled as a graph of interconnected modules spread across various nodes in the network. These modules may be data plane modules (TCP, IP, etc.) or control plane modules (IKE, routing, etc.), on the same network node or different network nodes strung together using pipes. The module abstraction, thus, model relationships, such as dependencies, peering and communication. Pipes connect modules and hide the complexity of the mechanisms needed to connect the modules,e.g., inter-process communications, socket based connections etc.

Fig. 4 shows an example of module abstraction and presents a scenario for the implementation of secure IP-Sec based communication. In the figure, the IP-Sec module delivers data over the IP module, which in turn uses the ETH module. The IP-Sec module is also dependant on the Internet Key Exchange (IKE) protocol module to set up end-to-end secure keys for a session. Similarly, the IKE module uses the UDP module over the IP module to establish end-to-end keys which it returns to the IP-Sec module. Fig. 4 is an abstract view of the module design in which each module has a switching function that allows it to pass packets between up-pipe (connecting to modules above it in the same node) and down-pipes (connecting to modules below it in the same node). The switching state may be produced locally through the protocol action or may be provided externally through a network manager. A fault management framework based on the CONMan abstraction is presented by Ballani et al. [14].

This modular view is very similar to an UML based system design, defining a system as an aggregation of distributed and interconnected function, differing, however, in the fact that it has been optimized to define highly dynamic systems that require continual

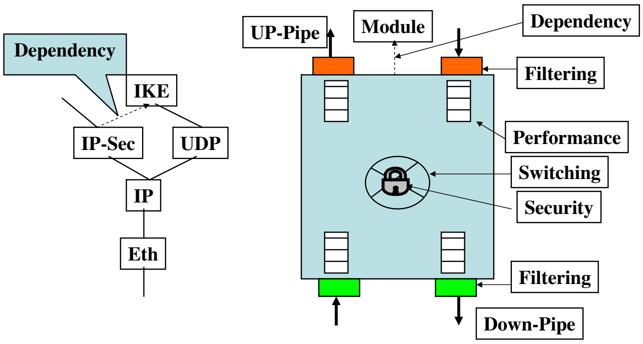

Fig. 4. CONMan: module abstraction and dependency graph.

validation and re-configuration of the system through a centralized authority. Thus, CONMan takes a less extreme measure than 4D by centralizing the configurability of the network at the granularity of module interactions rather than centralizing the whole control and management plane.

6.3. Maestro

Maestro [45,27] proposes an operating system like approach for network control and management. In such an architecture, network controls are implemented as applications over an operating environment. The operating environment provides support to the network control applications much in the same way an operating system provides support to the applications, by providing services such as (1) scheduling, (2) synchronization, (3) inter-application communication, and (4) resource multiplexing.

Maestro also proposes a clean-slate architecture and advocates the need to provide clear abstractions and interfaces between protocols, in the same spirit as that of 4D or CONMan. However, unlike 4D or CONMan, Maestro proposes implementing an explicit protection mechanism through defining network-wide invariants in the face of control mechanisms. This provides an extra cushion against any configuration errors, right from high-level configuration description to their lower-level implementation.

A high-level view of the Maestro architecture is shown in Fig. 5. Maestro uses a Meta-Management System (MMS) channel which is similar to the dissemination channel of the 4D architecture. Also, just like the discovery mechanism of 4D, Maestro collects topology and other information of the underlying network over the MMS channel. The operating platform uses this data to construct a virtual view for control applications running on top of it. Each application is provided with the specific and relevant view of the network that it needs to see.

As an example [45], a QoS routing application is not presented with the routers B3, B4, and A4 by the virtual view layer since they are relevant for the QoS routing computations. Similarly, suppose inter-domain policy necessitates the need to prevent B2 from being the egress router for ISPX. To implement such a policy, the virtual view provides a view to the shortest path routing application devoid of the information that B2 is a border router.

Hence, while the 4D architecture treats the control and management functions as one single monolithic entity, Maestro treats them as an aggregate of multiple functions, with an operating environment and network level invariants ensuring synchronization among the functions and validating their outputs.

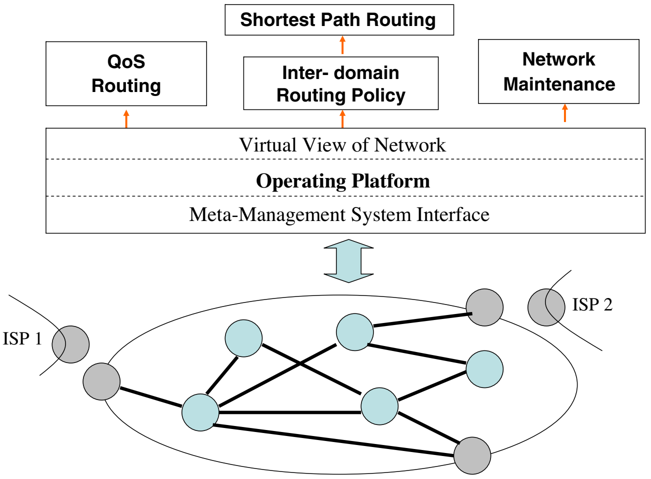

Fig. 5. Maestro architecture.

6.4. Autonomic network management

In 2001, IBM presented a holistic vision for autonomic computing in which the system as a whole would attain a higher degree of automation than simply the sum of its self-managed parts [157].

Based on this motivation, the autonomic network architecture (ANA) project [158] is a clean-slate meta-architecture for the next generation networks. The key objectives of ANA are similar to those of the self-* properties of an autonomous system stated in [157]. However, pertaining specifically to an autonomic network architecture, ANA motivates a networking architecture composed of self-configuring nodes, self-organizing into a network system through neighbor interactions, with multiple such systems self-federating into a heterogeneous Internetwork. Apart from these, the networking systems should possess the properties of self-protection and self-healing.

The ANA framework allows the co-existence of multiple heterogeneous systems composed of "compartments". The ideas are similar to the idea of realms in Internet 3.0 [166], except that rather than simply managing its own private address space, compartment membership entails being "able, willing and permitted to communicate to each other according to compartment wide policy and protocols". Every compartment has a hypothetical database that stores the information of each member. Before being able to communicate with any member of the compartment, a resolution process is required to access the database to find the way to access the member. Additionally, addressing is done through local identifiers called labels. To communicate with a remote peer, the sender sends the packet with a local label. This local label identifies and "Information Dispatch Point" (IDP) to which a "channel" is bound. The "channel" is an abstraction of the path setup as a result of the resolution process.

Additionally, functional blocks that are entities like packet processors can be inserted into the data path on demand. Using these, ANA provides multiple mechanisms to do a network operation by runtime selection and switching of protocols. Thus, functional composition and monitoring allows ANA to implement its self-* properties.

Although the ANA architecture does not define a specific method for network control and management, we include it in this section since we believe that autonomic systems and their self-* properties define a new paradigm of management and control architectures and have the potential to be the basis for the next generation networking architectures.

A holistic framework for autonomic network management based on ubiquitous instrumentation is proposed in [158]. The way protocols are built today, with measurement being just an add-on function, the future network protocols need to be built around a well-engineered instrumentation mechanism. Based on data from these measurements, local and global policies and mechanisms for global data sharing, the task of global decision making may be automated depending on centralized or distributed management paradigm.

6.5. In-Network Management (INM)

While ANA is a generic architectural framework for autonomic systems composed of autonomic devices, In-Network Management (INM) [54,42,61] proposes a more specific architectural design for embedding management capabilities in all network entities and leveraging the management capabilities that can be achieved as a result of their collaboration. Thus, INM advocates a paradigm of management service composition using several autonomous components. Also, in this regard, INM is quite different from the centralized architectures of 4D, CONMan and Maestro.

In INM, management functionalities are embedded into every network node. Different levels of embedding management capabilities into functional components (device drivers, network protocols, etc.) are defined: (1) **Inherent**: Management capability inseparable from the logic of the component (e.g., TCP congestion control), (2) **Integrated**: Management capability internal to a functional component but separable from the component logic, and (3) **External**: Management capability located on another node.

Fig. 6 shows a high-level view of the INM node architecture. The InNetMgmt Runtime environment is the container in which functional components and InNetMgmt services can run. The InNetMgmt Packages, InNetMgmt framework and InNetmgmt platform are the different levels of abstractions of management function primitives. The InNetMgmt platform provides the most primitive capabilities that can be enabled on a wide set of devices. InNetMgmt framework provides primitive capabilities for a narrower set of devices and the InNetMgmt packages provide technology specific functional add-ons. Functional components are logical entities inside a node that may have their own management capabilities or may be entities that compose a management functionality. The InNetMgmt Services are specific utilities that can be used by management applications. An example of such utility is a command mediation service which allows management applications to issue commands and receive responses from the functional components.

Fig. 7 shows the generic design of a functional component for INM. Functional components may have their own management modules with a well-defined management interface. The management interface allows functional components to exchange management information. Also, every component needs to have a service interface through which it can expose domain specific functionality and a supervision interface through which the framework may manage and monitor the component.

Having discussed the node architecture and the component architecture, we now present an overall architecture of INM in Fig. 8.

A network administrator can connect using an INM application (point of attachment) connected to the INM kernel. Instead of using a centralized policy server to disseminate and enforce policies on every node, INM allows the policies to be deployed on any node and passed onto others using a P2P mechanism implemented as a component.

We suppose that the INM design would be highly scalable compared to centralized solutions. However, there is some inherent complexity in defining abstract generic interfaces and also in converting network-wide policies into a distributed management state.

To summarize, in this section, we discussed some of the leading proposals for management and control architectures for the next generation Internet. The ideas varied from being extreme design points proposing a completely centralized design in 4D to much milder distributed designs of ANA and INM. Also, it seems that the research community shall have to reach a consensus on whether management and control functionality should be stripped away from protocols and established as a separate service or whether it should still continue to exist as part of protocols. However, there seems to be some unity of thought in the fact that protocols need to implement generic management interfaces through which a network entity may communicate management information and decisions with other entities in the network, be it a central policy server disseminating specific state information or network peers communicating local policy.

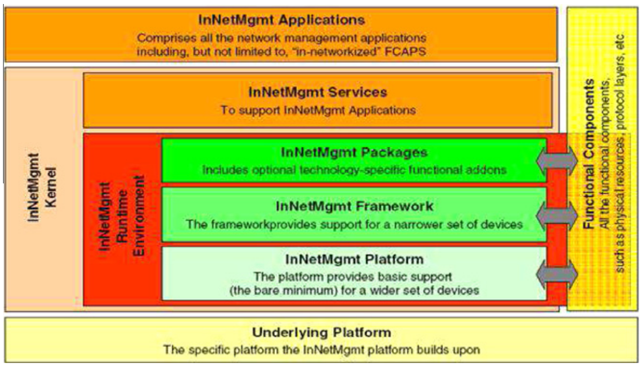
InNetMgmt Applications
Comprises all the network management applications including, but not limited to, "in-networkized" FCAPS
InNetMgmt Kernel
InNetMgmt Services
To support InNetMgmt Applications
InNetMgmt Runtime Environment
InNetMgmt Packages
Includes optional technology-specific functional addons
InNetMgmt Framework
The frameworkprovides support for a narrower set of devices
InNetMgmt Platform
The platform provides basic support (the bare minimum) for a wider set of devices
Functional Components
All the functional components, such as physical resources, protocol layers, etc
Underlying Platform
The specific platform the InNetMgmt platform builds upon

Fig. 6. INM architecture [54].

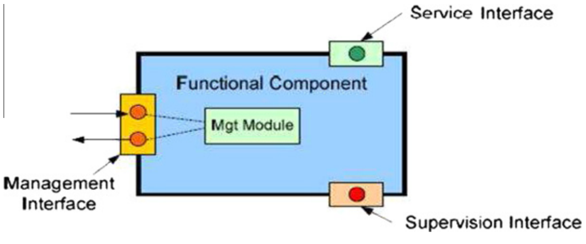

Fig. 7. INM: functional component.

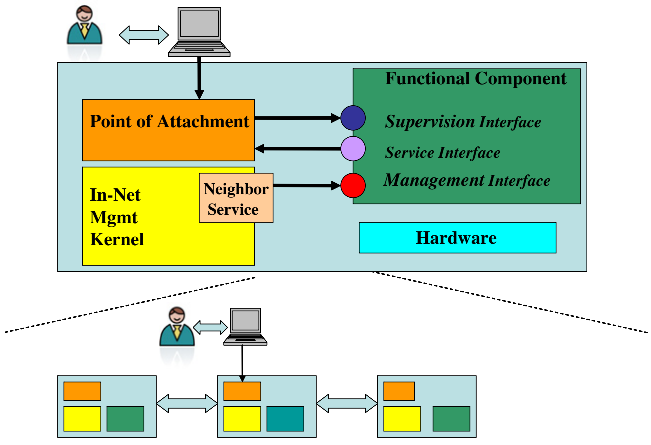

Fig. 8. INM: functional component + configuration.

7. Service centric architectures

The commercial usage of Internet, ubiquitous and heterogeneous environments, new communication abstraction, and security and management challenges require the next generation Internet to provide a broad range of services that go far beyond the simple store-and-forward paradigm of today's Internet. Research efforts focusing on defining a new service architecture for the next generation Internet are motivated by the following requirements: (1) how the architecture can be flexible and adaptive, (2) how to avoid the ossification [7] of the current Internet, and (3) how to map the user-level service requirements into the lower layers such as infrastructure layer's implementation. FIND projects on service architecture are relatively more technical or detailed, meaning that they try to make the service implementation easier and more flexible, though through different ways: (1) Service-Centric End-to-End Abstractions for Network Architecture: put application function to the routers (service-centric abstraction), (2) SILO: divide into flexible services and methods across the whole networks, and support cross-layer, and (3) NetServ:

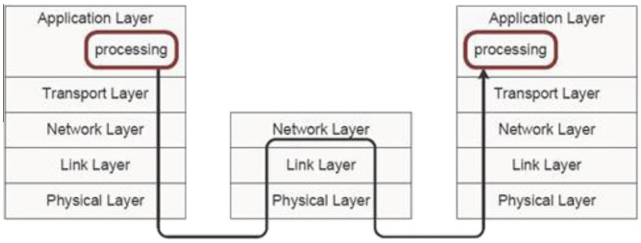

Fig. 9. Layered Internet architecture [184].

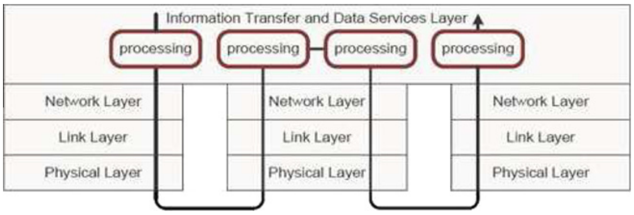

Fig. 10. Information transfer and data services architecture [184].

self-virtualized in lower layers, put service to IP layer. In comparison, the EU FP7 projects are more concerned about the relationship among different interested parties and how to setup the service agreement and achieve the service integration from business level to infrastructure level.

7.1. Service-Centric End-to-End Abstractions for Network Architecture

The traditional end-to-end based Internet design puts almost all the service intelligence into the end-hosts or servers, while the network only performs hop-by-hop packet forwarding. The network processing is performed at no higher than the network layer. The network function of packet forwarding was oblivious to the end-to-end service requirements with the network providing a single class best effort service to all end-to-end service flows. This purposeful design is the basis of the simplicity underlying the current Internet architecture and was suitable in the context under which it was designed. However, commercialization of the Internet introduced diverse end-to-end service requirements, requiring more diversified network services. The Service-Centric End-to-End Abstractions for Network Architecture [184] seek to define a new service architectural framework for the next generation Internet. The idea is to develop the communication abstraction around the transfer of *information* rather than the transfer of *data*. Information is at a higher level of abstraction than data and the basic task of the network should be transferring information rather than just data packets. Packets by themselves are just parts of the representation of information. This new abstraction idea is called Information Transfer and Data Service (ITDS).

The other key idea of this solution is that it utilizes network-process-based routers as infrastructure components. These routers will have to be aware of the application layer service information rendering the network to be an integral part of the service architecture rather than just a store-forward functionality. Figs. 9 and 10 present a comparison of the network stacks between the current Internet and the one with the new ITDS idea.

Based on the ITDS abstraction, some example scenarios of the data services are shown in Figs. 11–13. A reliable and private communication scenario is presented in Fig. 11. It consists of two data services implementing reliability and privacy functionality. The combination of the services can then be applied to the other types of point-to-point information transfer. Fig. 12 presents a scenario of combining a caching service with a reliability service. Different end-hosts then can use the same caching service. This combinational service can be applied to conventional point-to-point caching service. The scenario in Fig. 13 shows a multicast service which could include a large number of end-systems. Moreover, different end-systems can have content trans-coding operation to adapt the presentation of the information to be transferred.

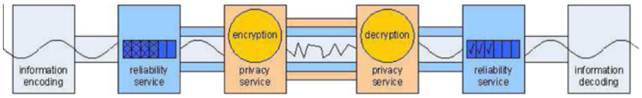

Fig. 11. Reliable and private communication [184].

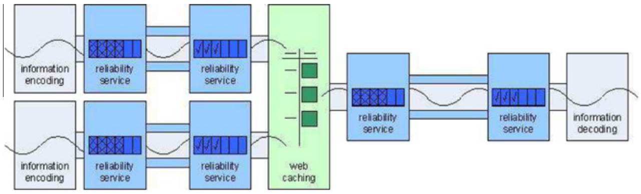

Fig. 12. Web caching [184].

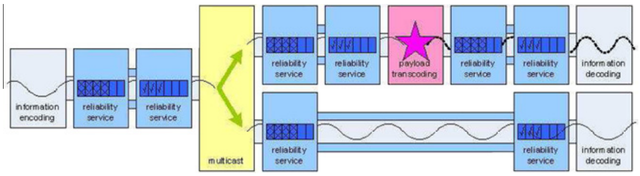

Fig. 13. Content distribution and trans-coding [184].

In such a framework, it is important to decide where has to be assigned the processing task across the Internet entities. This is also known as the service mapping problem. The service placement across the network is shown in Fig. 14.

The mapping requirements are almost on every layer of the system such as end-to-end layer, router layer, or even port processors layer. However, this mapping problem is known to be NP-complete.

This service architecture basically changes the conventional end-to-end assumption underlying the current Internet and advocates on putting more functionality into routers, besides their gen-

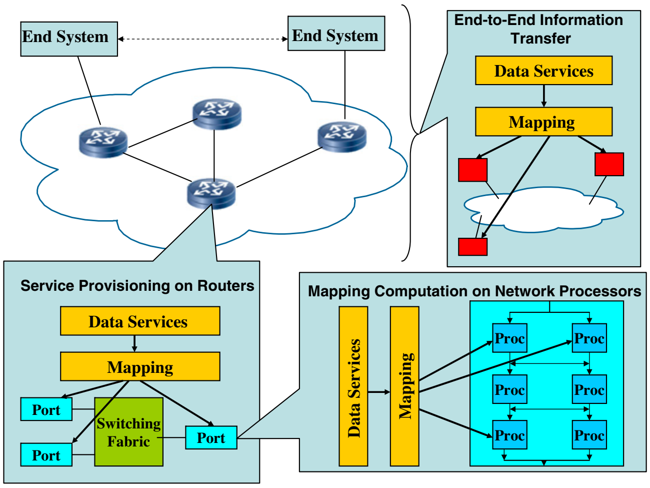

Fig. 14. Service mapping [184].

eral store-and-forwarding functionality. The architecture requires application layer processing capabilities throughout the network, including the end-systems as well as the routers. Thus, the feasibility of such a requirement of the routers remains to be validated.

Secondly, the service mapping will not be an easy problem, that is to say, deciding how much capacities to invest into general purpose processing and how much for service processing will be an important issue, which requires a good heuristic solution easy and efficient enough for the future. This problem is known to be NP-complete and will be tackled by exploring different heuristics. It is also necessary to consider how different processing functions can be controlled from the point of view of the network as well as the end-system.

7.2. SILO architecture for services integration, control, and optimization for the future Internet

The current Internet is facing the so-called "Balkanization" [74] problem because the new emerging networks and protocols do not necessarily share the same common goals or purpose as the initial Internet.

The SILO architecture [190] presents a non-layered inter-networking framework. The basic idea is that complex communication tasks can be achieved by dynamically combining a series of elemental functional blocks to form a specific service. That is to say, it can break the general strict layered model and form a more flexible model for constructing new services. Because of this flexibility, it is also easier to do the cross-layer design which is difficult to be done in the current Internet architecture.

The design goals include: (1) Supports for a scalable unified architecture, (2) cross-service interaction, and (3) flexible and extensible services integration.

In SILO, services are the fundamental building blocks. A *service* is a self-contained function performed on application data such as: "end-to-end flow control", "in-order packet delivery", "compression", and "encryption". Each service is an atomic function focusing on providing specific function. These small services can be selected to construct a particular task, but the order of these services does not necessarily obey the conventional "layer" sequence and can embrace a much more flexible precedence constraints.

Different from *service*, *method* is an implementation of a service that uses a specific mechanism to realize the functionality associated with the service. An ordered subset of methods within which each method implements a different service is called a *silo*. A silo is a vertical stack of methods and a silo performs a set of transformation on data from the application layer down to the network or infrastructure layer. *Control agent* is the entity inside a node which is in charge of constructing a silo for an application and adjusting service or method parameters to facilitate the cross-service interaction. In SILO architecture, for each new connection, a silo is built dynamically by the control agent. The basic architecture and their components relationship are shown in Fig. 15. The cloud is the universe of services which consists of services represented by circles. Multiple methods can be used to implement the same service inside every circle. Solid arrow means the sequence constraints of constructing the service. Control agent interacts with all elements and constructs silos according to the precedence.

From Fig. 16 we can see that one of the biggest advantages of SILO is that it blurs the distinction between core and edge entities, and each network node is free to implement any service. Moreover, the modularity of services, different protocols for the same layer and different implementations of the protocol can be "plugged in and out" easily. Because of this, the fine-grained cross-layer design naturally becomes very easy and efficient.

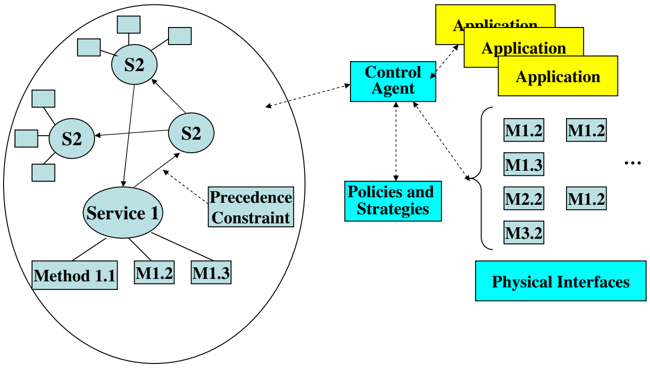

Fig. 15. SILO architecture [190].

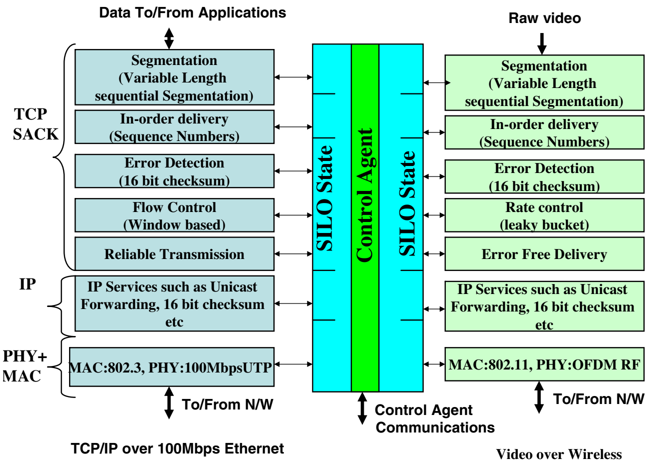
Data To/From Applications
Raw video
TCP SACK
IP
PHY+ MAC
Segmentation (Variable Length sequential Segmentation)
In-order delivery (Sequence Numbers)
Error Detection (16 bit checksum)
Flow Control (Window based)
Reliable Transmission
IP Services such as Unicast Forwarding, 16 bit checksum etc
MAC:802.3, PHY:100MbpsUTP
SILO State
Control Agent
SILO State
Segmentation (Variable Length sequential Segmentation)
In-order delivery (Sequence Numbers)
Error Detection (16 bit checksum)
Rate control (leaky bucket)
Error Free Delivery
IP Services such as Unicast Forwarding, 16 bit checksum etc
MAC:802.11, PHY:OFDM RF
To/From N/W
Control Agent Communications
To/From N/W
TCP/IP over 100Mbps Ethernet
Video over Wireless

Fig. 16. SILO examples: (a) TCP/IP emulation and (b) MPEG video transmission over wireless [190].

However, because the design is significantly different from the current Internet, one of the biggest puzzles is that it is not easy to be validated or implemented. It is also important and difficult to define and identify the appropriate building block services. Moreover, the cross-layer design is always related with optimizations, it remains as a future research topic for this issue. The control functionality of the system is also important for efficiency. Further control optimization related research may be needed.

7.3. NetSerV: architecture of a service-virtualized Internet

It is well known that the current Internet architecture is resistant against adding new functionality and services to the network core, which is also called "ossification" problem [7]. Adding new network service is not as easy as adding new application to the end-points. Two typical examples are the failure of broad scale implementation of multicast routing and Quality-of-Service (QOS). Presently, diversified network services have to be implemented over application-level overlays. However, application-layer overlays cannot effectively use the resources in the other layers. For example, each overlay network implements their own application layer measurement and monitoring framework while such information may be readily available with the generic monitoring framework of the underlying infrastructure layer.

The NetServ project [105] aims to develop efficient and extensible service architecture in the core network to overcome the ossification. As shown in Fig. 17, it tries to break up the functionalities of the Internet services and makes individual building blocks to construct network services. Each building block is a single unit of network resource or function such as linking monitoring data or

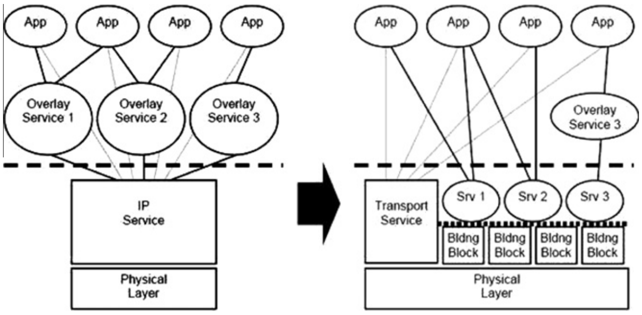

Fig. 17. NetSerV: transition to a new Internet service architecture.

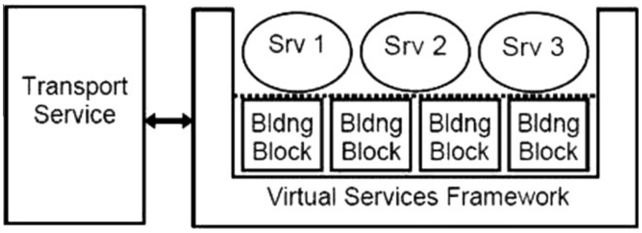

Fig. 18. NetSerV: virtual service framework.

routing tables that can further be used or assembled by the upper layer function. This structure can be hosted on any network node such as a router or some dedicated servers. Moreover, as shown in Fig. 18, network service can run over one or more nodes offering the building blocks and the services can run on a "virtualized services" framework which consists of a group of building blocks operating individually.

The idea of breaking the basic functionalities into building blocks eases the flexibilities of assembling upper-layer services and adding new functionalities into the architecture. However, it also means significant changes to the current layered structure of the network stack. It will also be a challenge to prove that the changed model or structure can offer better or similar efficiency and reliability for the current functions such as routing and data delivery. Fundamental changes to the network stack always mean risks and also new potential security holes which need further observation and evaluation. Moreover, how to build the building blocks and how to divide them into different groups (or how to do the abstraction of the basic functions), and even how to make them interact – all remain to be solved. The protocols and mechanisms for service discovery and service distribution are also important issues.

7.4. SLA@SOI: empowering the Service Economy with SLA-aware Infrastructures

SLA@SOI means Service Level Agreements (SLAs) within a Service-Oriented Infrastructure (SOI) [191]. Different from the earlier discussed service architecture research projects of FIND, which focus more on network stack modification, the SLA@SOI from EU is more about "multiple" ideas of future Service-Oriented Infrastructure. Specifically, its goal is to realize the vision of dynamic service provisioning by a SLA management framework in multi-level environment, i.e., scenarios involving multiple stakeholders and layers of business/IT stack. To realize dynamic service provisioning, there are three challenges that must be addressed:

1. Predictability and dependability of the quality of services.
2. Management of SLA transparently across the whole network.
3. Support of highly automatic and dynamic negotiation, provision, delivery, and monitoring services.

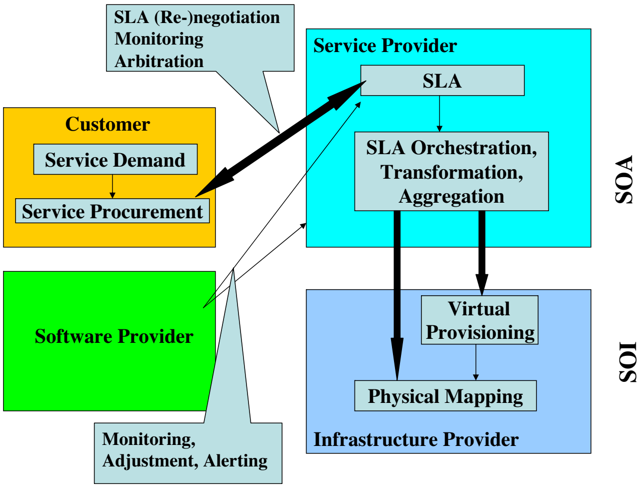

Fig. 19. Overview of the automatic SLA management process.

Thus, the main goal of SLA@SOI is to provide an SLA management framework allowing consistent management and specification of SLAs in a multi-level environment. The main innovations include:

1. SLA management framework.
2. Adaptive SLA-aware infrastructure.
3. Business management and engineering method for predictable system.

The SLA@SOI architecture is focused on the service relationship setup and maintenance between customers, service providers, infrastructure providers, and software providers. It is trying to set up a high-level business relationship or framework, business perspective framework to facilitate the service deployment or implementation from business level down to the infrastructure level. Fig. 19 offers a simple overview of the SLA management process. In today's layered system, it is not easy to map user-level SLA into physical infrastructure. Thus, in Fig. 19, we can see that SLA@SOI includes the mapping of higher-level SLA requirement onto lower levels and the aggregation of low-level capabilities to higher levels. The vertical information flow basically reflects the service interdependencies and the originating business context, and support proxy and negotiation process at each layer.

The biggest advantage of SLA@SOI is to set up an inter-party Service Level Agreement framework in multi-level environment between different parties such as customer, software provider, service provider, and infrastructure provider. Unlike other research projects in FIND which are more about long-term research rather than short-term industry need, SLA@SOI provides a more high-level architecture for the service deployment and implementation in real business environment. However, we can also notice that it is not easy to set up a simple framework and ask all the different parties to obey, and it could take more time and effort beyond the technical aspect to realize this goal. Moreover, the realization of the high-level Service Level Agreement also needs detailed technical support like other FIND projects which are researching to apply the user-level requirements to the infrastructure.

7.5. SOA4All: Service-Oriented Architectures for All

SOA4ALL stands for the Service-Oriented Architecture for All [193]. SOA4All is endorsed by the Networked European Software and Services Initiative (NESSI) [104].

SOA4ALL aims at providing a comprehensive framework that integrates four complementary and evolutionary technical advances (SOA, context management, web principles, Web 2.0 and

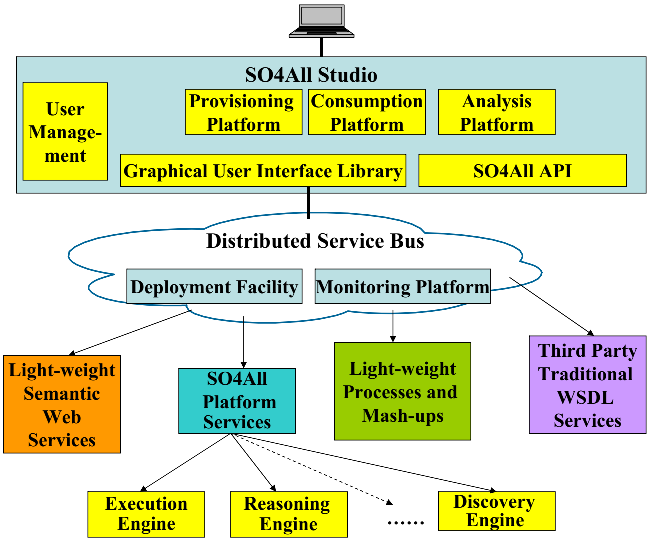

Fig. 20. SOA4ALL architecture [190].

semantic technologies) into a coherent and domain independent service delivery platform.

The overall architecture of SOA4ALL includes four parts: SOA4ALL Studio, Distributed Service Bus, SOA4ALL Platform Service, and Business Services (third party Web services and light-weight processes), as shown in Fig. 20.

In the center of the architecture is the SOA4ALL Distributed Service Bus which serves as infrastructure service and core integration platform. The Distributed Service Bus delivers the necessary extension and developments towards a large scale, open, distributed and web-scale computing environment. The SOA4ALL Studio delivers a web-based user front-end that enables the creation, provisioning, consumption and analysis of the platform services and various third party business services that are published to SOA4ALL. The Studio supports different types of users at different times of interaction. The platform services deliver service discovery, ranking and selection, composition and invocation functionality, respectively. These services are usable as any other published service in the architecture. Their functionalities are used by the Studio to offer client requested functionalities.

SOA4ALL tries to integrate the most recent and advanced technologies into a comprehensive framework and infrastructure to provide an efficient web of billions of services. Research challenges for SOA4ALL include the openness of the future web communities and whether the openness and mobility will pave the way towards a real explosion on the web.

7.6. Internet 3.0: a multi-tier diversified architecture for the next generation Internet based on object abstraction

Internet 3.0 project at Washington University in Saint Louis [78,167] is a clean-slate architecture to overcome several limitations of the current Internet. The top features are: strong security, energy efficiency, mobility, and organizational policies. The architecture explicitly recognizes new trends in separate ownership of infrastructure (carriers), hosts (clouds), users and contents and their economic relationships. This will shape the services that the network can provide enabling new business models and applications.

As shown in Fig. 21, Internet 1.0 (approx 1969) had no ownership concept since the entire network was operated by one organization. Thus, protocols were designed for algorithmic optimization with complete knowledge of link speeds, hosts, and connectivity. Commercialization of Internet in 1989 led to multiple ownership of networking infrastructure in what we call Internet 2.0. A key impact of ownership is that communication is based on policies (rather than algorithmic optimization) as is seen in inter-domain (BGP) routing. The internals of the autonomous systems are not exposed. We are seeing this trend of multiple ownership to continue from infrastructure to hosts/devices (Clouds), users, and content. Internet 3.0's goal is to allow policy-based secure communication that is aware of different policies at the granularity of users, content, hosts, or infrastructure.

Cloud computing is an example of applications that benefit from this inherent diversity in the network design. Hosts belonging to different cloud computing platforms can be leased for the duration of experiments requiring use of data (e.g., Gnome) to be analyzed by scientists from different institutions. The users, data, hosts, and infrastructures belong to different organizations and need to enforce their respective policies including security. Numerous other examples, related to P2P computing, national security, distributed services, cellular services exist.

Organization is a general term that not only includes employers (of users), owners (of devices, infrastructure, and content) but also includes logical groups such as governments, virtual interest groups, and user communities. Real security can be achieved only if such organizational policies are taken into account and if we design means of monitoring, measurement, and independent validation and enforcement.

Internet 1.0 was designed for host systems that had multiple users and data. Therefore, the hosts were the end systems for communication. Today, each user has multiple communication devices. Content is replicated over many systems and can be retrieved in parallel from multiple systems. The future user-to-user, user-to-content, machine-to-machine communications need a new paradigm for communication that recognizes this new reality and allows mobility/multihoming for users and content as easily as it does for devices. In this new paradigm, the devices (hosts) are intermediate systems while the users and content are the end-systems.

The inclusion of content as an end-system requires Internet to provide new services (e.g., storage, disruption tolerance, etc.) for developing application specific networking contexts. There will be more intelligence in the network which will also allow it to be used easily to use by billions of networking-unaware users.

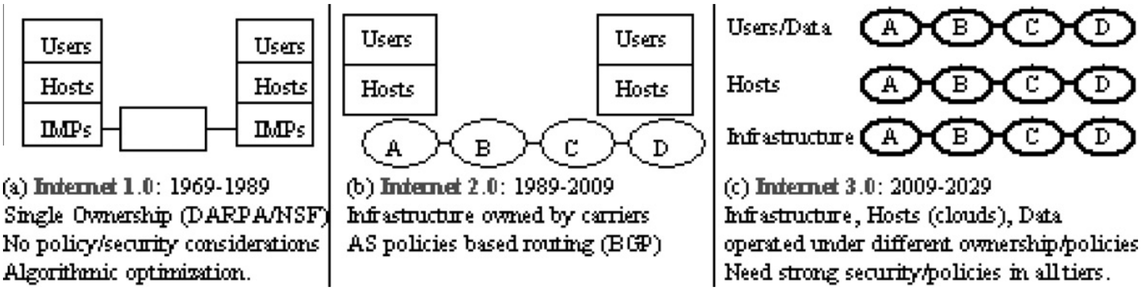
Users
Hosts
IMPs
Users
Hosts
IMPs
(a) Internet 1.0: 1969-1989
Single Ownership (DARPA/NSF)
No policy/security considerations
Algorithmic optimization.
Users
Hosts
A
B
C
D
Users
Hosts
(b) Internet 2.0: 1989-2009
Infrastructure owned by carriers
AS policies based routing (BGP)
Users/Data
Hosts
Infrastructure
A
B
C
D
(c) Internet 3.0: 2009-2029
Infrastructure, Hosts (clouds), Data
operated under different ownership/policies
Need strong security/policies in all tiers.

Fig. 21. Internet generations.

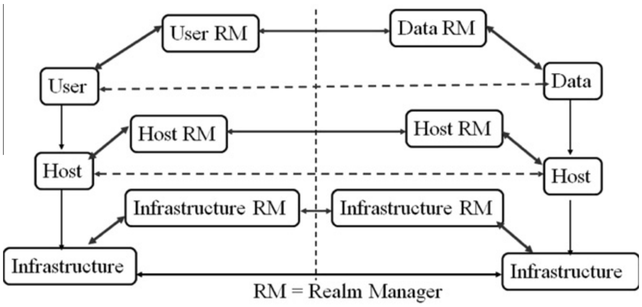

Fig. 22. Organization of objects in Internet 3.0.

Internet 3.0 uses the term "Realm" to represent a trust domain such as an organization. All entities within a tier belonging to a single organization belong to a realm. The management and control plane of the realm, which we generically call Realm Manager (RM) enforces security and other organizational policies. These policies can be very general and may include security considerations, e.g., authentication and authorization. RMs also provide additional services such as ID-locator translation that allows objects to move without loosing connections and energy management services. RMs are part of the management and control plane and are active during the start phase of a communication. Once set up, the communication can continue in the data plane without intervention of the RMs.

Realms overlay entities with a discrete ownership framework. Ownership entails related security, administrative and management responsibilities. In the "Three-tier Object Model" (Fig. 22), the bottom tier infrastructure is owned by multiple infrastructure owners. The second tier of hosts is owned by individual users or different organizations such as DoE, DARPA, and Amazon. The third tier of users and data may belong to specific organizations or individual users. Thus, realms represent logical division of entities into multiple ownership, trust, and management domains.

Explicit representation of ownership simplifies the security and policy framework design through more natural representation and enforcement of policies rather than conflating them with functionality as in the current Internet.

Realms advertise etcific services through "Objects." Objects encapsulate the complexities of resource allocation, resource sharing, security and policy enforcements, etc., and expose a standard interface representing capabilities (in standardized abstract parameters) and fixed or negotiable policies.

Objects provide services. They may use services of other objects to provide their own services. Also, a service may consist of an aggregation of objects, e.g., end-to-end transport service. The aggregated objects may belong to the same or multiple ownerships. Thus, object composition in Internet 3.0 lies at the basis of the policy and security framework of the architecture.

Like real organizations, realms are organized hierarchically. The hierarchy is not a binary tree since a realm can have two or more parents, i.e., an organization can be a part of several higher-level organizations and can have several lower-level sub-organizations. Note that the concepts of objects and realms are recursive. An object may comprise a group of objects. Thereby, a realm or a group of realms could be treated as an object and provide a service.

When a realm advertises an object, it keeps complete control over how that object is managed internal to the realm. Inside the realm, the realm members may delegate responsibilities to other objects. This allows the objects to go to sleep for energy saving. It allows specialized services in the realm that can be used by other objects. For example, all packets leaving a realm may be signed by a "realm signer" that assures that the packets originated from that realm although the source of the packet was not authenticated. In some applications, this type of assurance is sufficient and useful in accepting or discarding the packet.

Each object has an ID and a locator. The ID is unique in the realm and is assigned by the RM. The locator is the ID of the object in the lower tier. Thus, the locator of data is the set of IDs of hosts on which the data resides. The locator of the host is the set of IDs of infrastructure points of attachments to which the host is connected. This separation of ID and locators among multiple tiers is unique and is the basis for allowing independent mobility of users over hosts and hosts over infrastructure. It is also the basis for multihoming of users (a user with multiple host devices such as a smart phone, a laptop, and a desktop).

At the infrastructure tier, the object abstraction framework is implemented through a management and control plane connecting Internet POPs installed with a special Internet 3.0 node called the "context router." Fig. 23 presents a highly simplified POP design where each AS has a border router that connects to the POP, enhanced with the context router. The context router has two key functionalities: (1) It maintains a "behavioral object" repository advertised by the participating ASs and makes them available for lease to application contexts, (2) It leases "programmable objects" provisioned over packet processing hardware resources such as SRAMs, DRAMs, and network processors. This allows application contexts to set-up their own packet processing contexts at POPs (shown as the hatched and dotted contexts).

Fig. 24 presents a high-level overview of the context router design. A context router needs to have multiple virtualized contexts advertised as "programmable objects". A hypervisor is responsible for creating and controlling these "programmable objects". There is a base context called the context 0 that hosts the object store. Participating AS's advertise their objects at the POP and they are stored at the context 0 of the context router. Also, the context 0 participates in the inter-infrastructure realm management plane and stores AS level connectivity maps. It runs a brokering protocol that allows application contexts to query, block, lease and release objects from the object store. A secure software switch allows inter-context communications, mostly to allow application contexts to be able to communicate with the context 0.

At the host tier,"programmable objects" are provisioned over compute resources consolidated over end-user personal compute resources, private and public cloud computing and storage resources, Content Delivery Network (CDN) storage resources, server farms, grid resources, etc. The mechanisms for sharing common compute resources across multiple application contexts may vary from virtualization techniques [215,208,209] achieving near perfect isolation and providing strong deterministic performance guarantees to traditional operating system based resource allocations based on global optimization and fairness considerations. Similar to the infrastructure realm, Internet 3.0 allows complete autonomy to host realms to choose the specific mechanisms for allocation of compute resources to application contexts. Also, it provides a common object abstraction interface that allows host resources to be shared across multiple ownerships over a policy negotiation plane. However, unlike the infrastructure realm which was marked by a physical realm boundary, host realms could have physical as well as logical boundaries. "Behavioral objects" are provisioned similar to the Software-as-a-Service (SaaS) or Platform-as-a-Service (PaaS) paradigms in cloud computing. Security and other services may be advertised as behavioral objects which advertise the service in terms of abstracted parameters such as security level, etc. An application context should be able to choose the required level of security without worrying about how the end-to-end security service is being archestered across the different host realms. The underlying federation mechanism requires considerable efforts in standardization.

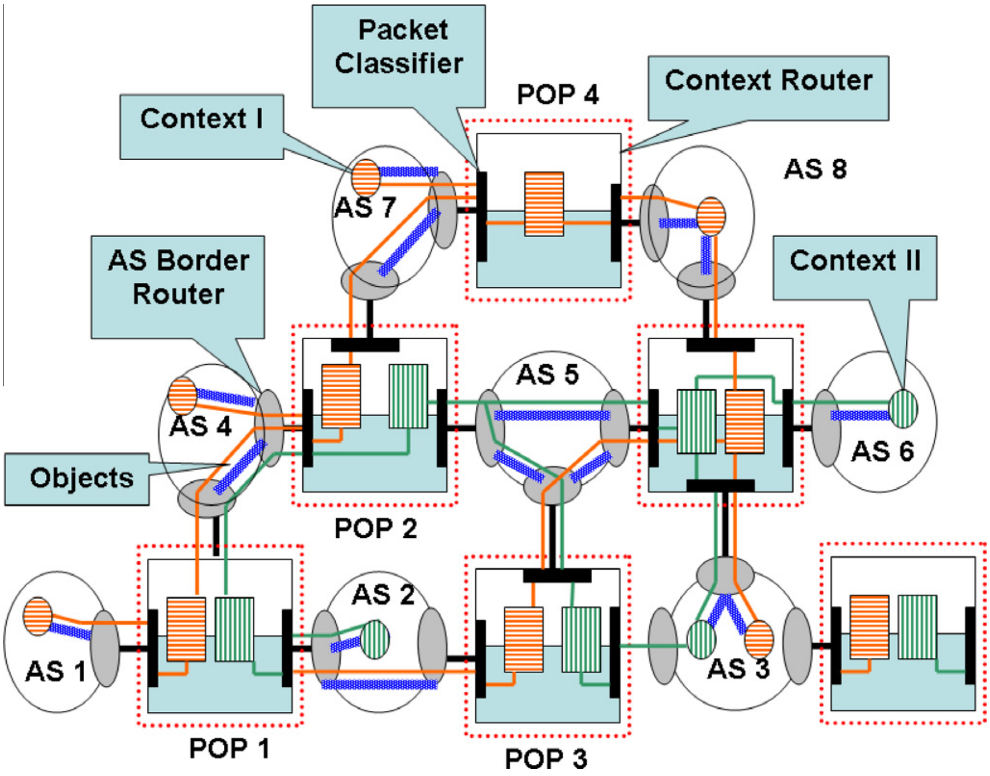

Fig. 23. Internet 3.0: POPs enhanced with context routers.

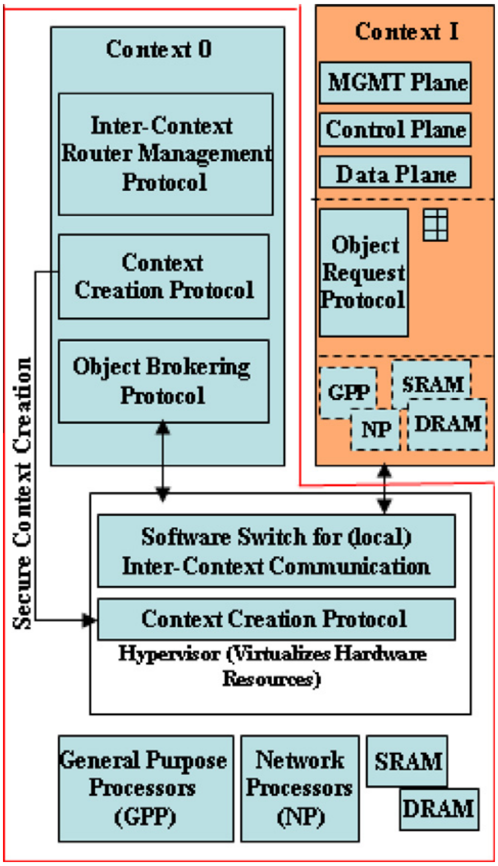
Context 0
Inter-Context Router Management Protocol
Context Creation Protocol
Object Brokering Protocol
Context I
MGMT Plane
Control Plane
Data Plane
Object Request Protocol
GPP
SRAM
NP
DRAM
Secure Context Creation
Software Switch for (local) Inter-Context Communication
Context Creation Protocol
Hypervisor (Virtualizes Hardware Resources)
General Purpose Processors (GPP)
Network Processors (NP)
SRAM
DRAM

Fig. 24. Internet 3.0: context router design.

Thus, Internet 3.0 is an overarching architecture for the next generation Internet. It identifies the key design basis and defines primitives that shall allow the next generation Internet to be diversified. It significantly improves upon the "one-suit fits" all paradigm of the current Internet and allows each application context to be able to fully program and optimize its specific context [167].

8. Next generation internetworking architectures

The current state-of-art of the routing function at the internetworking layer of the Internet is marred with numerous problems. The biggest and most immediate concern is that of scalability. With the huge growth in network-able devices participating in the Internet, the routing infrastructure is finding it difficult to provide unique locaters to each of these devices (address depletion problem) and the routing nodes are unable to cope with the exponential growth in routing table sizes, number of update messages and churn due to dynamic nature of networks [100]. Border Gateway Protocol (BGP) [20], the *de facto* inter-domain routing protocol of the current Internet, takes in the order of minutes to re-converge after a failure [84,86]. The basic advantage of packet switched networks in providing higher resilience is hardly implemented in practice. Also, basic services such as mobility, quality of service, multicasting, policy enforcements and security are extremely hard to be realized, if at all. New innovations proposed to mitigate some of the ills, such as IPv6 [71], have hardly seen any wide-scale deployment. Another issue is the tussle between user's need to control the end-to-end path and the provider policies to optimize their commercial interests. These and other weaknesses of the routing mechanism in the current Internet have resulted in a spur of activity trying to design a better routing architecture for the next generation Internet. While some of the schemes are clean-slate, thus requiring a complete architectural overhaul, others are more incremental that can be implemented over the present underlying system to procure partial benefits. In this section, we discuss some of these proposals that have the potential to change the routing architecture of the future Internet.

8.1. Algorithmic foundations for Internet architecture: clean slate approach

Leveraging the advances in algorithmic theory since the time the current routing model of the Internet was developed. Awerbuch and Haberman [12] advocate a fresh look at the algorithmic basis of the current routing protocols. A simple example to justify this claim lies in the inability of the current routing protocols to route around congestion. The current routing is based on static load insensitive metric that does not adapt dynamically to avoid congested paths. The proposed solution to this problem led to a "tragedy of the commons" condition wherein all the flows greedily try to route through the least congested path resulting in the routing protocol acting as its own adversary and causing wasteful oscillations of flows across sub-optimal paths. Also, it is rightly claimed [12] that the routing model is based on the assumption of "blind trust" where the routing system is not robust in itself but depends on the absence of intelligent adversaries, huge over-provisioning and manual interventions. Proposals to secure routing assume the presence of trusted anchors for policing and authentication, avoiding the hard condition of compromised trust anchors.

The fundamental approach advocated in this proposal to overcome the weakness of the current routing protocols is to define a new routing metric that can dynamically adapt itself to congestions and attacks by a Byzantine insider, provide incentives for selfish users, and guarantee QoS through effective and efficient sharing of heterogeneous resources. The selection of such a dynamic adaptable metric entails changes in the hierarchical IP based path computation methods. A new scalable path computation mechanism, along the lines of flexible peer-to-peer routing architectures, that can be mapped to this underlying metric needs to be determined. Also, the new metric and the path computation mechanism should be able to accommodate the future requirements of content-based routing.

A proposed economics-inspired metric with all the desired property is called the "opportunity cost" price function. The idea is to attach a cost to each resource (node memory, node bandwidth, CPU processing, etc.) such that an un-utilized resource is available at "zero" cost, with the cost becoming higher for a higher utilized resource. An application requiring such a resource needs to justify the cost of the resource against the benefit of acquiring it forming the basis of an effective QoS framework. An application is allowed to specify an arbitrary "benefit" per unit of flow. The QoS admission control and routing are done by finding the shortest path in the opportunity metric and comparing this cost to the benefit of the flow. If the benefit of the flow is more than the opportunity cost of the path, the flow is admitted. This mechanism warrants selfish applications reporting higher benefits to grab more resources for their flows. Such a condition is avoided through an incentive mechanism that assigns a fixed budget of opportunity cost to an application.

Having defined the metric, the routing mechanism needs to be made secure against insider Byzantine attacks. Greedy methods of path selection based on past history fail to counter dynamic adversaries that follow a specific attack pattern matching the greedy path selection. Such adaptive or dynamic adversaries need not be a third party attacker. But the routing system itself, owing to the weakness of its algorithmic basis, acts as its own adaptive adversary under the "tragedy of commons" situation. A simple algorithm to counter such a situation involves the adaptive metric which keeps track of the losses encountered across each edge and selecting a path probabilistically such that the probability of selecting a path grows exponentially with the past losses in that path. To avoid the "tragedy of commons" situation in adaptive routing to counter congested paths, a mechanism wherein the routers artificially suppress the acknowledgments based on a probability dependant on the current congestion condition is devised. These artificial suppression of acknowledgments feed the loss metric view of the network for each flow that try to route along the least cost path over this metric based on a flow control mechanism that adaptively re-routes the flows.

The dynamic metric discussed thus far needs to be supported over large network topologies in a scalable manner. The topological hierarchy aids aggregation (and thus scalability) of the current Internet. Such aggregation schemes designed for a static metric become ineffective for a network based on a dynamic metric. Thus, instead of aggregating based on pre-determined fixed identifiers, a new aggregation scheme based on physical location is defined. The proposal is to devise a new locality preserving, peer-to-peer directory service rather than a fixed infrastructure DNS service.

Thus, a newer algorithmic basis for Internet protocols hold the potential to free the current Internet routing from most of the current constraints that it faces, especially in the routing plane. The contributions of this proposal, if implemented, shall lay the basis of a highly dynamic and hence more robust routing function for the next generation Internet.

8.2. Greedy routing on hidden metrics (GROH Model)

One of the biggest problems with routing in the current Internet is scalability. The scalability problem is not so much due to the large space requirements at routers but is more due to the churn as a result of network dynamics causing table updates, control messages and route recalculations. The problem is expected to exacerbate further with the introduction of IPv6. This problem seems to be unsolvable in the context of the present design of routing protocols, hinting towards the need of some truly disruptive ideas to break this impasse.

The GROH model [83] proposes a routing architecture devoid of control messages. It is based on the "small world" phenomenon exhibited in Milgram's social network exercise [101] and later depicted in the famous play "Six Degrees of Separation" [66] in 1990. This experiment demonstrated the effectiveness of greedy routing in a social network scenario and can be established as the basis of routing in the Internet which shows similar scale-free behavior as that of social networks, biological networks, etc. The idea of greedy routing on hidden metrics is based on the proposition that: "Behind every metric space including the Internet there exists a hidden metric space. The observable scale-free structure of the network is a consequence of natural network evolution that maximizes the efficiency of greedy routing in this metric space". The objective of the GROH model is to investigate this proposition to try and define the hidden metric space underlying the Internet topology and develop a greedy routing scheme that maximizes the efficiency of routing in this metric space. Such a greedy routing algorithm belongs to the class of routing algorithms called "compact routing" that are aimed at reducing the routing table size, the node addresses and the routing stretch (the ratio of distance between the source and destination for a given routing algorithm to that of the actual shortest path distance). However, existing compact routing algorithms do not address the dynamic nature of networks, such as the Internet.

Three metric spaces are being considered initially as part of the investigation to model the Internet's scale-free topology. They are: (1) Normed spaces, (2) random metric spaces, and (3) expanding metrics. Now using a concrete measured topology of some network (in this case, the Internet) 'G' and these metric spaces, their combinations or additional metric spaces as a candidate hidden metric space 'H', a fit of 'G' into 'H' is found. If a fit of 'G' into 'H' is found successfully, two tasks are undertaken: (1) Label size determination – based on the metric space ‘H’, labels are assigned to each node such that they facilitate the quick calculation of distance between two nodes and (2) label assignment for new nodes – a new node inspects the labels of its neighbors in ‘G’ and deduces its location in the metric space ‘H’. Based on these, the greedy routing algorithm forwards packets to the neighbor that takes the packet more closer towards the destination than any other neighbor. Such knowledge comes at the cost of the node having to maintain the distance of every destination from each of its neighbors. However, no network wide updates are necessary to keep this information and hence avoiding network churn.

An effort towards update-less routing is a promising step towards solving the scalability problem of the Internet. However, it remains to be seen whether such a modeling of the Internet bears feasible and practically usable results.

8.3. HLP: hybrid link state path-vector inter-domain routing

Border Gateway protocol (BGP) [176] is the *de facto* standard for inter-domain routing in the current Internet. However, BGP fails to satisfy the needs of an efficient, scalable, secure, and robust inter-domain routing protocol. Well-known problems of BGP route oscillations and instabilities [62,63,206,85,86,177], slow convergence [84,93], blind trust assumptions and lack of support for trouble shooting have inspired research efforts towards a new Inter-domain routing protocol. HLP [195] is a step forward in this direction and claims to be a “clean-sheet redesign of BGP”.

The BGP routing is based on AS (autonomous system) path vectors and is agnostic to relationships between ASs. This leads to local routing events being propagated globally, thus affecting the scalability of BGP. HLP leverages the inherent peering, customer and provider relationships between ASs to define a hierarchical structure in inter-domain routing. The implicit inter-AS relationships in BGP are explicitly stated in HLP to be able to contain local routing events such as routing updates, security or configuration errors, and policy enforcements within relevant boundaries. Based on this, HLP sets two policy guidelines: (1) Export-rule guideline – routes advertised by a peer or provider are not advertised to another peer or provider and (2) route-preference guideline: Prefer routes through customers over routes through peers or providers.

Another fact used by HLP is that prefix-based routing, as in BGP, does not usually result in differing paths than when routing is done at the granularity level of ASs. Nonetheless, routing at the granularity of ASs significantly improves the scalability of the routing system and hence adopted by HLP. Thus, routing at the granularity of ASs and having established a hierarchical ordering of ASs, HLP implements a hybrid link state and path vector routing protocol such that a link state protocol is used as the routing protocol within an AS hierarchy (of provider customer relationships) while path vector is used for routing between these hierarchies. Link state protocols have their advantages of fast convergence and low churn while path vector protocols are more suitable for policy enforcements and scalability. HLP tries to exploit the advantages of both worlds. A high-level view of the HLP mechanism as discussed so far can be seen in Fig. 25.

HLP is not a clean-slate or highly innovative design. However, it is a positive step forward from breaking away from numerous incremental changes applied to BGP [176] to re-design an inter-domain routing protocols from grounds up. Thus, HLP is a starting point from where newer inter-domain routing protocol ideas may be born.

8.4. eFIT [94] enabling future Internet innovations through transit wire

ISPs and user networks have different purposes and characteristics. ISPs are used as a commodity in the present Internet with the sole purpose to maximize the efficiency of data transport while minimizing their costs. User innovations that do not have immediate positive impact or do not guarantee returns in the foreseeable future are generally not appealing to ISPs. On the other hand, user networks are generally the source of data and also the seat of innovations. However, the current Internet design is such that ISP’s and user networks share a common address space. Thus, user innovations cannot be isolated to user networks and often they roll over to requiring changes in the ISP networks. This tussle of motivation between user networks and ISPs limits the innovations that can be deployed into the Internet.

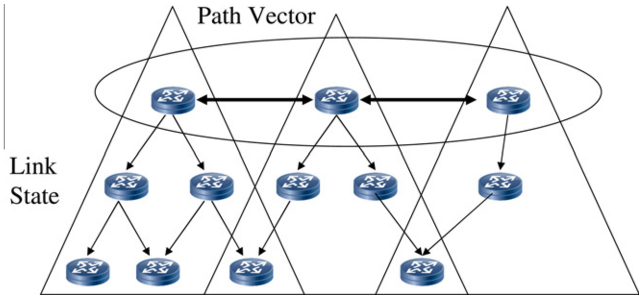

Fig. 25. HLP overview.

eFIT [94] proposes a new routing architecture based on the separation of the transit core from the user networks. Such a separation allows each of these components to evolve independently, and given the difference in their motivations and objectives, this separation allows them to evolve in the proper direction. The idea is to abstract the transit core as a wire connecting the user networks. This wire, called the “Transit Wire,” provides a strong universal connectivity between the user networks and evolves with the objective to provide efficient, effective, available, affordable and plentiful data transit service for the user networks. The user networks can thus innovate freely without the concern of having to change any part of the deployed infrastructure of the transit core. A mapping service acts as an intermediary between the two components, mapping user addresses into transit paths and also providing interoperability between diverse mechanisms and protocols used at the two ends of the wire.

The eFIT idea is thus a clean-slate extension of the already existing ideas of edge-core separation for the current Internet. However, while most core edge separation ideas are motivated to alleviate the Internet routing scaling problems, eFIT is motivated by the distinct objectives and separate paths of innovations of these two components.

8.5. Postmodern internetwork architecture

Bhattacharjee et al. in their proposal on a “Postmodern Internet Architecture” [21] address the classic tussle between users and providers at the internetworking layer of the current Internet. Users often require finer granularity control over the end-to-end path selection. However, such control is often not conducive to provider policies such as traffic engineering policies, and route export policies that try to optimize their commercial interests. It is claimed that the architecture is “postmodern” since it is a “reaction against many established network layer design concepts.” The key idea is to separate the policy plane of internetworking protocols from the forwarding or data plane, as in the current Internet. The key goal is to re-design the internetworking layer of the Internet such that it supports diverse functions that in turn can support diverse policies. Indeed, the internetworking layer of the current Internet is often overlaid with functions to support policies for which it was not designed. Some of the different policies identified in this proposal include accountability, anonymity, authentication, authorization, censorship, confidentiality, copyright enforcement, protocol filtering, spam filtering, etc.The key mechanism proposed is an explicit separation of policy enforcement from functionality at the internetworking layer.

The network in the ''Postmodern Internet Architecture'' is organized hierarchically. At the basic level of the hierarchy the network is a set of ''nodes'' interconnected by a set of ''links.'' At the higher level in the hierarchy, nodes are organized into trust domains called ''realms.'' Realms are virtual nodes at this hierarchical level and are connected through links.

The forwarding function design is motivated by the need to allow users control over the end-to-end path selection without compromising on provider policies. It is implemented over this topology with ''links'' as the basic connectivity primitive. A similar idea has been implemented by the ''Pathlet Routing'' [60] architecture presented by Godfrey et al. Each link has a linkID. Every forwarding element knows the linkIDs directly connected to it. Each packet carries in its header a forwarding directive (instead of just a destination address as in the current IP) which specifies the linkID on which the forwarding element should forward the packet. In case a forwarding element receives a packet without a forwarding directive for it, it raises a ''forwarding fault'' and the packet is forwarded to a ''fault handler.'' The fault handler is the point of policy control for the provider. A user may (at least) specify the realms over which the packet should be forwarded while allowing providers to control the packet's transit through its realm. However, if a provider chooses to advertise its internal link organization outside its realm, the user may be able to specify the exact link level forwarding directive for the corresponding forwarding element. Unspecified linksID- level forwarding directives in the end-to-end path of the packet are called ''gaps'' and are filled-in by the authorized realm when such packets raise a ''forwarding fault'' at a boundary forwarding element of the realm. Once the forwarding directive for the ''gap'' is resolved, the information may be cached by either recording the packet path in the packet header or by caching the forwarding at each intermediate forwarding element within the gap.

The routing function is hierarchical following the ''realm'' hierarchy. The inter-realm protocol is concerned with advertising policy compliant paths between realms to aid the computation of end-to-end paths. The intra-realm protocols are similar to traditional IGPs (Interior Gateway Protocols) that may be overridden by traffic engineering mechanisms. Following the realm hierarchy, these protocols too have a recursive structure wherein an intra-realm protocol maybe an inter-realm protocol at a higher level in the hierarchy.

Apart from these, the architecture also proposes to build-in an explicit framework for ''accountability'' and ''motivation'' to aid resolve tussles in the internetworking space. The accountability framework is a security model that securely establishes the identities of entities that are related to the packets creation and/or its traversal over the network. The motivation framework on the other hand allows each packet to securely justify specific treatment at intermediate forwarding elements.

The new internetworking architecture proposed in this work addresses an extremely important issue in the current Internet- the tussle as a result of conflating policy and functionality. The Internet 3.0 architecture, discussed in Section 7.6, shares a similar motivation (among others) to the requirement to enable diversity in the internetworking layer of the Internet. The realm based organization of nodes into trust domains is similar, albeit more specific to the internetworking layer, to the ideas presented in the ''Policy Oriented Network Architecture (PONA)'' [166]. Overall, it is an interesting work on an extremely relevant area of research for the future Internet. However, the primary assumption in this work that it will be sufficiently cheap to carry extra bits on packet headers in the future Internet (with bandwidth becoming cheaper and more available) rather than sacrifice functionality remains to be validated. Also, the underlying security mechanisms, especially to enable an accountability and motivation framework, are expected to introduce considerable computational cost at the internetworking layer and it will be interesting to see experimental results of these implementations.

8.6. ID-locater split architectures

Current Internet is faced with many challenges including routing scalability, mobility, multihoming, renumbering, traffic engineering, policy enforcements, and security because of the interplay between the end-to-end design of IP and the vested interests of competing stakeholders which lead to the Internet's growing ossification. The architectural innovations and technologies aimed at solving these problems are set back owing to the difficulty in testing and implementing them in the context of the current Internet. New designs to address the major deficiency or to provide new services cannot be easily implemented other than by step-by-step incremental changes.

One of the underlying reasons is the overloaded semantics of IP addresses. In the current Internet, the IP addresses are used as session identifier in transport protocols such as TCP as well as the locater for routing system. This means that the single IP address space is used as two different namespace for two purposes, which leads to a series of problems. The Internet Activity Board (IAB) workshop on routing and addressing [100] reached a consensus on the scalable routing issue and the overloaded meaning of IP addresses. It urged further discussion and experiments on decoupling the dual meaning of IP addresses in the long-term design of the next generation Internet. Currently, there are several proposals for ID-locater split, but most of them cannot provide a complete solution to address all the challenges including naming and addressing, routing, mobility, multihoming, traffic engineering, and security.

One of the most active research groups of IRTF (Internet Research Task Force) is RRG (Routing Research Group) [181], where there is an on-going debate on deciding which way to go among several ID-locater split directions. One possible direction is called ''core–edge separation'' (or ''Strategy A'' in Herrin's taxonomy [91]) which tries to keep the de-aggregated IP addresses out of the global routing tables, and the routing steps are divided into two levels: the edge routing based on identifier (ID) and the core routing based on global scalable locaters. ''Core–edge separation'' requires no changes to the end-hosts. Criticisms to this direction include difficulty in handling mobility and multihoming, and handling the path-MTU problem [91]. In some solutions, the ''weird'' ID-based routing in the edge also makes some purist believe that it is a short-term patch rather than a long-term solution. Typical solutions include LISP, IVIP, DYNA, SIX/ONE, APT, TRRP (all from [181]). This ''core–edge separation'' can be deemed as decoupling the ID from locater in the network side, which is an intuitive and direct idea for the routing scalability issue and relatively easy to deploy, but not good at solving the host mobility, host multihoming, and traffic engineering. Other recent RRG proposals include: 2-phased mapping, GLI-split, hIPv4, Layered Mapping System (LMS), Name Overlay (NOL), Name-Based Sockets, Routing and Addressing in Next-Generation EnteRprises (RANGER), and Tunneled Inter-domain Routing (TIDR). They are related to this category from different aspects such as naming, addressing, sockets, encapsulation, mapping, and hierarchy.

The other direction is called ''ID locater split'' which requires globally aggregatable locaters to be assigned to every host. The IDs are decoupled from locaters in the end-hosts' network stacks and the mapping between IDs and locaters is done by a separate distributed system. The proposals following this direction handle mobility, multihoming, renumbering, etc., well. However, they do require host changes and it may be hard to ensure compatibility with the current applications. Typical solutions include HIP [68], Shim6 [188], I3 [194], and Hi3 [106].

It is seen that these two directions have their own advantages and disadvantages, and it is hard to judge which one is right for the future Internet. Here we describe two example solutions (HIP and LISP) of these two directions, and after that we discuss our MILSA [161,162] solution which combines the advantages of these two directions and avoids their disadvantages.

8.6.1. HIP

HIP (Host Identity Protocol) [68] is one of the most important ID locater split schemes which implements the decoupling of ID from locater in end-hosts. It has been under development in the HIP working group of IETF for couple of years.

HIP introduces a new public keys based namespace of identifiers which enable some end-to-end security features. The new namespace is called Host Identity (HI) which is presented as a 128-bit long value called Host ID Tag (HIT). After the decoupling of HIs from IP addresses, the sockets are bound to HITs instead of IP addresses, and the HITs are translated into IP addresses in the kernel. HIP defines the protocols and architecture for the basic mechanisms for discovering and authenticating bindings between public keys and IP addresses. It explores the consequence of the ID locater split and tries to implement it in the real Internet.

Besides security, mobility and multihoming are also HIP's design goals and are relatively easier to implement than the "core–edge separation" solutions. HIP supports opportunistic host-to-host IP-Sec ESP (Encapsulation Security Protocol??), end-host mobility across IPv4 and IPv6, end-host multi-address multihoming, and application interoperability across IPv4/IPv6.

However, for HIP, although the flat cryptographic-based identifier is useful for security, it is not human-understandable and not easy to be used to setup trust relationship and policies among different domains or organizations. It uses the current DNS system to do the mapping from ID to locater which is not capable of dealing with the mobility under fast handover situation, and multihoming. Specifically, mobility is achieved in two ways: UPDATE packets and rendezvous servers. First way is simple but it does not support simultaneous movement for both end-hosts. Rendezvous servers are better but do not reflect the organizational structure (realm), and there is no explicit signaling and data separation in the network layer.

Moreover, HIP requires that all the changes happen in the end-hosts which may potentially require significant changes to the current Internet structure and could lead to compatibility issues for the existing protocols and applications.

8.6.2. LISP

LISP (Locater ID Separation Protocol) [89] is another important ID locater split scheme following the "core–edge separation" approach which implements the decoupling of ID from locater in the network side instead of the host side. It is being developed by the LISP working group of IETF.

LISP is a more direct solution for routing scalability issue. LISP uses IP-in-IP packets tunneling and forwarding to split identifiers from locaters which eliminates the Provider Independent (PI) addresses usage in the core routing system and thus enables scalability. The tunnel end-point routers keep the ID-to-locaters cache and the locater addresses are the IP addresses of the egress tunnel routers. The mapping from ID to aggregatable locaters is done at the border of the network, i.e., the tunnel end-point routers.

LISP enables site multihoming without any changes to the end-hosts. The mapping from identifier to RLOC (Routing Locater) is performed by the edge routers. LISP also does not introduce a new namespace. Changes to the routers are only in the edge routers. The high-end site or provider core routers do not have to be changed. All these characteristics of LISP lead to a rapid deployment with low costs. There is also no centralized ID to locater mapping database and all the databases can be distributed which enable high mapping data upgrade rates. Since LISP does not require current end-hosts with different hardware, OS platform and applications, and network technologies to change their implementations, the transition is easier compared to HIP. The requirements for hardware changes are also small which allow fast product delivery and deployment.

However, LISP uses PI addresses as routable IDs which potentially leads to some problems. In the future, it will be necessary to create economic incentives to not use the PI addresses, or to create an automatic method for renumbering by Provider Aggregatable (PA) addresses.

Obviously, there is a tradeoff between compatibility to the current applications and enabling more powerful functions. Since LISP does not introduce any changes to the end-host network stack, by design it cannot support the same level of mobility as HIP. The host multihoming issue is similar. Specifically, from design perspectives, LISP lacks support for host mobility, host multihoming, and traffic engineering. Some researchers argue that LISP is a short-term solution for routing scalability rather than a long-term solution for all the challenges listed in the beginning of this section.

8.6.3. MILSA

MILSA [161–164] is basically an evolutionary hybrid design which has combined features of HIP and LISP, and avoids the disadvantages of these two individual solutions. Since there is still a debate regarding whether the ID locater split should happen in end-host side such as HIP or in network side such as LISP, it is hard to decide which is the right way to go at this point of time. Thus, MILSA is designed to be adaptive; it supports both directions and allows them to evolve to either direction in the future. By doing this, we can avoid the deployment risk at the furthest.

Specifically, MILSA introduces a new ID sublayer into the network layer in the current network stack, i.e., it separates ID from locater in the end-host and uses a separate distributed mapping system to deliver fast and efficient mapping lookup and update across the whole Internet. MILSA also separates trust relationships (administrative realms) from connectivity (infrastructure realms). The detailed mechanisms on how to setup and maintain this trust relationship are presented in [161]. A new hierarchical ID space is introduced which combines the features of flat IDs and hierarchical IDs. It allows a scalable bridging function that is placed between the host realms and the infrastructure realms. The new ID space can be used to facilitate the setup and maintenance of the trust relationships, and the policy enforcements among different organizations. Moreover, MILSA implements signaling and data separation to improve the system performance, efficiency, and to support mobility. Detailed trust relationship setup and maintenance policies and processes are also presented in MILSA.

Through the hybrid combination, the two approaches are integrated into one solution to solve all the problems identified by the IRTF RRG design goals [90] which include mobility, multihoming, routing scalability, traffic engineering, and incremental deployabiltiy. It prevents the Provider Independent (PI) address usage for global routing, and implements identifier locater split in the host to provide routing scalability, mobility, multihoming, and traffic engineering. Also the global routing table size can be reduced step by step through our incremental deployment strategy which is also one of the biggest MILSA advantages. Specifically, in MILSA, different deployment strategies can be implemented to gain fastest routing table size reduction considering the different incentives or motivations from both technical and non-technical aspects, i.e., the strategies make sure that each incremental deployment step of MILSA can pay off with reasonable and acceptable balance between costs and benefits. Different incentives such as scalability, mobility, and multihoming lead to different deployment models which have different effect in reducing the routing table size gradually.

8.7. Other proposals

Several other routing ideas, spanning diverse issues in routing such as user control, simplified management and control, and multipath routing have been proposed. These are discussed in this section.

8.7.1. User controlled routes

This [221] is a source routing proposal in which users are allowed to choose the path to destinations. The motivation for this work is similar to other source routing schemes: (1) foster competition among ISP's, and (2) allow more diversity and control to users in path selection. The mechanism involves route maps which are static maps of preferred routes of a user. Unlike traditional path vector mechanisms, route maps are learnt through advertisement about customers and peers initiated at the provider. Also, these advertisements specify costs involved with the paths. The route maps of a user along with their preference are stored in a Name-to-route-lookup service (NRLS). To formulate a route to a destination, the user first needs to obtain the destination's route map and preference and try and intersect the best possible combination with its own route map. While the route maps are static information about AS connectivity, more dynamic link state information using "connectivity maps" are also disseminated. Connectivity maps allow users to update their preferences and route around problem areas. The impact of such a mechanism shall be to support application specific networking paradigms more naturally as part of the architecture.

User controlled routing is still in its nascent stage with no discussion on the analytical concerns regarding engineering and it would be interesting to monitor how it progresses.

8.7.2. Switched Internet Architecture

The "Switched Internet Architecture" [187] proposal advocates a clean slate approach to re-design the Internet by combining the best characteristics of telephony and data. It proposes a new hierarchical addressing scheme along the lines of addressing in cellular and telephone networks. The two-level hierarchy consists of a network ID and a host ID. The network ID is a concatenation of a hierarchical geographical addressing scheme (continent code, country code, state code, area code) with an organization code. Based on this naming scheme, the architecture consists of a hierarchical "bank of switches", switching packets on predefined digit position in the addressing scheme. The network protocol stack as a result of this simplified switching architecture is reduced to an application layer operating on a port layer (providing port id and data buffering). This port layer operates on the switching layer above the physical substrate.

Though it is true that such a simple architecture shall allow many of the management, control, security and QoS concerns to be taken care of, there remain serious questions about dynamicity and ownership of such a network. The growth and success of the Internet to what it is today can be attributed to user demands fostering mutual cooperation among ISPs in a fair competitive environment. Introducing geographical ID into the addressing scheme fosters an implicit relation between all providers in the same geographical area. Also, the Internet model was designed to serve as a highly resilient and dynamic network, which may not be the case if fixed switching state is introduced in the routing plane.

8.7.3. Routing Control Platform (RCP)

RCP [26] has already been discussed (Section 6.1) in the context of the centralized approach towards network management and control. RCP is the extension of the idea presented in [50]. It proposes a centralized routing server (RCP) that computes BGP routes on behalf of the routers in the AS. RCP receives all BGP routing advertisements through iBGP, computes routing tables for each router subject to IGP view and domain policies, and disseminates routing tables to routers. Thus, RCP centralizes the distributed routing protocol allowing a cleaner and more effective routing management and control. Details of RCP implementation can be found in [96].

As already discussed earlier, policy enforcements in the current routing protocols cannot be enforced through a clean interface. They need to be implemented indirectly through tweaking routing parameters of specific routing protocols and hope for the desired output in routing tables. The increased complexity of routing management subject to the increasing needs of fine-grained policy control clearly suggests that this approach shall not scale in terms of increasing configurational complexity. Proposals such as RCP are thus extremely potent in defining the routing management and control of the future.

In summary, routing is undoubtedly one of the major functions of the network. Since there is a lot of concern about the scalability and security of the Internet routing mechanisms, the future Internet may see a complete paradigm shift. Research in areas of content distribution are advocating towards content centric [75] and data centric networks [79]. Along similar lines, Paul et al. [166] advocates the necessity of a finer granularity of policy enforcements wherein the user, data, hosts and infrastructure exist as separate entities logically grouped into trust/application domains. Virtualization techniques are touting co-existence of multiple application specific networks locally optimized for their specific purpose or objective. Next generation routing proposals, however, are all designed around the assumption of the present networking environment with added concerns of security, scalability and management. We feel that there is a disparity in the next generation Internet objectives between disruptive next generation architectural ideas and conservative routing architects.

9. Future Internet infrastructure design for experimentation

9.1. Background: a retrospect of PlanetLab, Emulab and others

The fast growth and diversification of the Internet made it extremely difficult to introduce new technologies and protocols backed up with sound experimental validation at realistic size testing environments. PlanetLab [168,111] was the first effort to design such a testbed facility that would effectively mimic the scale of the Internet by organizing thousands of Internet nodes, spread out at different geographic locations, under a common control framework. These Internet nodes, offered by various research, educational and industrial organizations, run Linux virtual server software to virtualize its resources, providing isolated resource allocation (called "slivers") to multiple concurrently active experiments. Experiments are allocated a "slice" which is composed of multiple slivers spanning multiple sites.

The node's virtual servers ("slivers") are managed by a "Node Manager" (NM), which also interacts with a centralized control module called the "PlanetLab Control" or PLC. Such a federated and distributed organization involving node contributors demand-ing control over the nodes that they own and users running experiments on these nodes, warrant the requirement of a trust based security model that can scale. To avoid a $N \times N$ blow up of the trust relationship model, the PLC acts as a trusted intermediary that manages the nodes on behalf of its owners according to a set of policies specified by the owners, creates slices by combining resources from these nodes and manages allocation of "slices" to experimenters. PLC supports two methods of actual slice instantiation at each node, direct and delegated. PLC runs a slice creation service called "pl_conf" at each node. In the direct method, PLC front-end directly interacts with the pl_conf service to create a corresponding virtual machine and allocate resources to it. However, in the "delegated" method, a slice creation agent on behalf of a user contacts the PLC for a "ticket". This "ticket" encapsulates rights to instantiate a virtual machine at a node and get specified resources allocated to it. The agent then contacts the pl_conf of each node to redeem this ticket and create a slice for the user. Currently, two slice creation services are supported on PlanetLab: (1) PLC, implementing the direct method and (2) Emulab, implementing the delegated method.

Over time, the PlanetLab design has been extended and modified to provide better and more efficient control and support. One such extension, within the PlanetLab control framework itself is to allow federation of separate and independent PlanetLab instances. Federation of such nature necessitates separate instances of PLC's to be able to communicate and coordinate with each other through well-defined interfaces. It can be easily observed that the PLC conducts two distinct functionalities: (1) node management on behalf of node owners and (2) slice creation on behalf of users, allowing the PLC to export two distinct interfaces. Also, adopting a hierarchical naming system for slices establishing a hierarchy of slice authorities ease trust and delegation related issues in federation. These extensions combined with added facility at the "pl_conf" to create slices on behalf of multiple slice authorities has lead to the development of regional and private PlanetLab instances that may peer with the "public" PlanetLab instance.

An instance of PlanetLab federation extension is the Planetlab-Europe testbed, supported by the Onelab project [112], which is the European contribution to the world-wide publicly available Planetlab testbed. However, the Onelab project is contributing to enhancing the monitoring infrastructure of Planetlab [180], extending Planetlab to newer contexts such as wireless testbeds [41,28,29], adding capability for IPv6 based multihoming of sites [107,108], dealing with unstable connectivity [97], integrating and mixing emulation tools [35], and providing a framework for network measurements.

PlanetLab being organized as an overlay over IP, it is not necessarily a realistic experimental substrate for network layer protocols. As such, actual routing protocols and router level code cannot be run effectively on a PlanetLab slice. The VINI [113,17] "running the Internet in a slice" (IIAS) effort was aimed at filling this void by leveraging the existing widely distributed PlanetLab network, User Mode Linux [43] and advances in open source router code. Fig. 26 presents the PlanetLab VINI slice organization. Router code requires root level kernel access. Thus, running router code directly over a Planetlab slice is not possible. VINI installs User Mode Linux (UML) [114,43] over the PlanetLab slice and installs open source router code, XORP [115] over it. UML provides a virtual Linux kernel implementation at the user-level. This sets up a distributed set of routers over a PlanetLab slice allowing network level experimentation. However, VINI routers are not directly connected to each other being part of the PlanetLab overlay network. Thus, any network level experimentation is hindered by interfering effect of actual path routers and corresponding routing protocols implemented on them.

Another extension of PlanetLab concerns extending the core mechanism of the overlay hosting facility. Overlay nodes run distributed applications that might involve a lot of packet routing and forwarding functionality. However, traditional overlay nodes are simple computers and are not designed for fast routing and forwarding of packets. Turner et al. [203] have designed a Supercharged PlanetLab Platform (SPP) that implements separate slow and fast paths for data processing and forwarding. The slow path is chosen for application specific processing while the fast path is optimized for line-speed packet forwarding and can be used by overlay applications needing large amounts of packet processing and forwarding. The biggest challenge facing the design of such an overlay node is compatibility with existing PlanetLab nodes and hiding the complexities of the node design from experimental code. Thus, SPP introduces a new genre of networking devices designed for optimized overlay hosting.

One drawback for experimental validation over realistic testing environments such as PlanetLab is poor repeatability and lack of experimental control. As an example, a researcher testing a new application optimized to handle intermittent network failures has to wait for the underlying network environment to face such a situation. Also, the nature of failures cannot be controlled and hence it is difficult to test the applications response to a wide range of failure modes. Additionally, the experiments cannot be repeated so that deterministic application behavior can be verified. On the contrary, a simulated testing environment can handle these requirements though not able to mimic the realistic scale and diversity of a realistic testbed. This clear partition of capabilities call for a solution that can leverage the best of both worlds. Emulab [109] is an effort in this direction. Emulab, as the name suggests, provides an "emulation" environment for network experimentation. The original Emulab has since been extended to accommodate simulated links and nodes within PlanetLab slices. This extension allows researchers access to realistic experiments and at the same time allowing fine-grained control and repeatability.

9.2. Next generation network testbeds: virtualization and federation

The next generation of network testbed research is primarily focused on virtualization and federation. Virtualization proposes efficient methods for resource sharing by multiple concurrent experiments on the testbeds subject to the constraints of maintaining high degree of isolation, fairness, and security. Federation research looks at the methods to unify multiple diverse testbeds designed to serve diverse experimental contexts and realistic experimental environment.

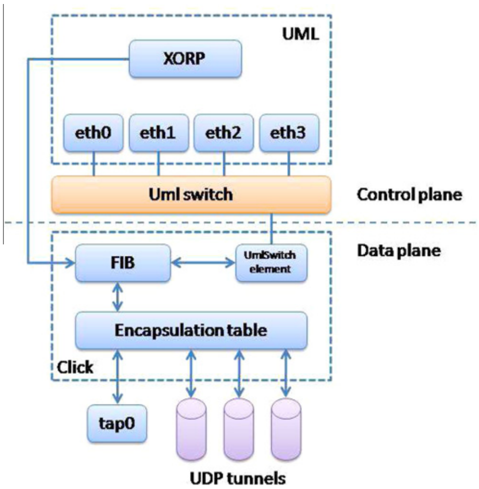

Fig. 26. An IIAS router on PL-VINI.

9.2.1. Federation

Networking testbeds strive to provide a realistic testing and experimentation facility to researchers. The prime goal is to be able to provide a platform that is as close to the production environment as possible.“Federation” helps realize this goal through [159] (1) Providing larger testbed scenarios, (2) providing a diverse testbed with specialized or diverse resources such as access technologies, (3) creating scientific communities with diverse research backgrounds and inspiring cross discipline research, and (4) amortization of costs through more efficient sharing.

However, there exists a lot of challenges that make federation an interesting research problem. These challenges can be categorized into technical challenges and political or socio-economic challenges.

The technical challenges involve problems such as (1) homogenization of diverse contexts to facilitate easy deployment of experiments, (2) fair and efficient sharing of scarce resources, and (3) interoperability of security protocols.

The political or social-economic challenges are based more on the implications of economic and organizational policies of sharing such as policies of governments, conflicts between research agencies, conflicts between commercial and non-commercial interests, and intellectual property rights related conflicts.

Thus, the problem of federation of testbeds has different contexts and the solution to a specific scenario for federation varies in accordance with the context. We shall discuss three approaches to federation that are under research currently in the European network community.

9.2.2. Virtualization

In spite of the tremendous success of the Internet, it is often made to deliver services that it was not designed for (e.g., mobility, multihoming, multicasting, anycasting, etc.). However, the IP based one-suite-fits-all model of the Internet does not allow innovative new architectural ideas to be seamlessly incorporated into the architecture. Innovative and disruptive proposals, either never get deployed or are forced to resort to inefficient “round about” means. The huge investments in the deployed infrastructure base of today’s networks add to this ossification by preventing newer paradigms of networking from being tested and deployed. Virtualization seems to be the only possible solution to break this current impasse [7].

Turner et al. [204] propose a diversified Internet architecture that advocates the ideas of virtualization of the substrate elements (routers) of the network infrastructure. Such an approach would allow researchers to implement and test diverse routing protocols (non-IP based) and service paradigms. The argument is that multiple competing technologies shall be able to co-exist in large scale experimentation and thus the barrier to entry from experimentation to production environments shall be reduced considerably. Such a testbed shall also be free from all intrinsic assumptions that commonly malice the credibility of conventional experimental testbeds.

CABO (Concurrent Architectures are Better than One) by Feamster et al. [51] is a design of the next generation Internet that allows concurrent architectures to co-exist. The key idea is to decouple the infrastructure from the infrastructure services. The infrastructure providers in CABO are expected to lease infrastructure entities such as backbone routers, backbone links, and switches, over which service providers could deploy their own specific protocols and run their own network services optimized to specific service parameters such as quality of service, low latency, and real-time support. The infrastructure providers may virtualize their infrastructure substrate and thus allow the isolated co-existence of multiple service providers.

An interesting new paradigm to allow the natural integration of virtualized network contexts implementing newer services is presented in the “Recursive Network Architecture (RNA)” proposed by Touch et al. [200,201]. RNA allows protocol instances to be dynamically created, bottom-up to implement a new networking context. It is argued that the present networking protocol stack is static although it operates in a dynamic environment wherein the protocol stack might need to change over time due to addition of new services and capabilities through newer protocols, different versions or implementations of existing protocols, etc. This leads to a very interesting observation made in the RNA proposal that protocols in the current Internet cannot dynamically modify their behavior based on sound assumptions of services implemented at a lower protocol layer, and thus the decision to include/exclude a new protocol service at a given layer is pushed all the way up to the user. An example in support of this observation is the decision to bind TCP to either IPv4 or IPv6 is taken at the application layer rather than at the TCP layer, where it would have been more appropriate. The key motivation of RNA is to develop a framework wherein protocols may be dynamically composed as per the functional requirements of its context (functions implemented in the protocol layer below it and the end-to-end region of the protocols extent). The key idea is to implement a generic meta-protocol that implements a generic set of functions and services and exposes knobs to allow configuration of these services based on the context of its instantiation at a protocol layer.This allows natural support for virtualized contexts to be dynamically defined over optimally composed network stacks to implement newer network functions. Also, the process of dynamic composition entails cross-layer interaction across the protocol instances, thus preventing re-implementation of redundant services at each layer.

Although not directly relevant to the central theme of the discussion in this section, the discussion on RNA provides a good context to discuss the “IPC model” of networking proposed by John Day [40]. The basic principle underlying the proposed model is that the author believes that “networking is basically inter-process communication (IPC).” Accordingly, in this model, “application processes communicate via a distributed inter-process communication (IPC) facility, and application processes that make up this facility provide a protocol that implements an IPC mechanism, and a protocol for managing distributed IPC (routing, security and other management tasks).” The model is “recursive” in that the IPC processes are able to request services from lower IPC facilities. It is argued by the author that this fundamental view of networking has the potential to solve some of the most fundamental problems in the current Internet including security, manageability, multihoming, mobility, and scalability.

The Internet 3.0 architecture, discussed in Section 7.6, also proposes an architectural framework that allows virtual networks optimized to serve specific application requirements to be dynamically setup over a policy and functionally “federated” resource base of the multi-ownership network. Also, the term “virtual networks” in Internet 3.0 is defined in the context of the “communication paradigm” based view of networking rather than the “communication system” based view of the current Internet. In the “communication system” based view of networking, the network is treated as a separate system that provides service(s) to application contexts through standardized interfaces. On the other hand, in a “communication paradigm” based view of networking, the network is an integral part of the application context and needs to be diversified and programmable such that it can be dynamically configured to optimize the specific requirements of an application context. Thus, virtual networks in Internet 3.0 are dynamically configured, programmable networks that serve specific application requirements and at the same time achieve performance and functional isolation between the separate virtualized network instances.

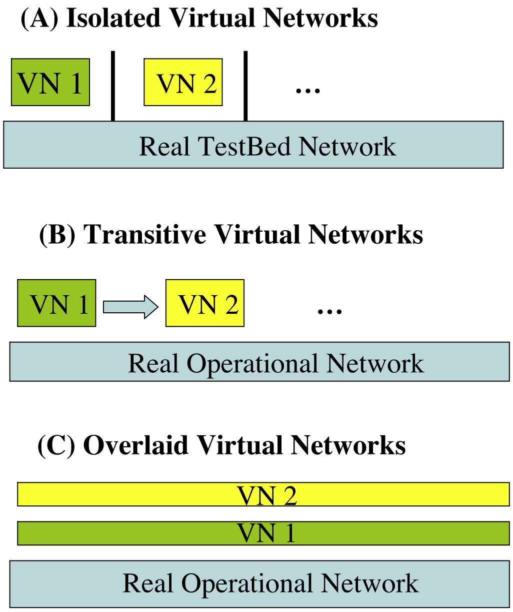

Fig. 27. AKARI: different virtualization models.

The AKARI Project [116,2] of Japan also advocates the use of virtualization as the basis of the Internet architecture in the next generation [67]. As shown in Fig. 27, the AKARI project extends the idea of isolated virtual networks to: (1) Transitive virtual networks – cooperation and/or communication between virtual networks and (2) overlaid virtual networks: one virtual network over the other.

However, though Internet-scale deployment of virtualization as the basis of the Internet architecture may not be possible in the near future, network testbed designs may immensely benefit from it. The properties of isolation and flexibility of virtualization suit the needs of next generation testbeds that need to be able to support diverse architecture experiments on a shared substrate such that they do not interfere with each other. Also, the feasibility of the core idea of virtualization as the basis of an Internet-scale network can be tested through experiences in deploying testbeds based on virtualization.

Virtualization in testbed design. The idea of virtualization to isolate network experiments running on shared substrate is not new. However, existing networking testbeds operate on an overlay above the IP based networks, seriously constraining the realism of network level experiments. To overcome this impasse, the future of networking testbeds shall have to be designed for end-to-end isolation, requiring the virtualization of end-hosts, substrate links and substrate nodes.

Turner [202] proposes a GENI substrate design that allows multiple meta-networks to co-exist. Each meta-network consist of a meta-router (a virtualized slice from a router) and meta-links joining the meta-networks. The design of substrate routers that support co-existence of several meta-routers has to cope with the challenges of flexibly allocating bandwidth and generic processing resources among the meta-routers, maintaining isolation properties. The three main components of a router are: (1) line cards – terminate physical links and process packets, (2) switching fabric – transfers data from line cards where they arrive to line cards connected to outgoing links, and (3) control processor – a general purpose microprocessor for control and management functions of the router such as running routing protocols and updating tables at the line cards.

A natural design choice of virtualizing such a hardware would be to virtualize the line cards to derive meta-line cards. However, this approach fails since the multi-core network processors on these line cards share a common memory causing the meta-line cards to interfere with each other. Instead, a "processing pool architecture" is employed in which the processing resources used by the meta-routers are separated from the physical link interfaces. As shown in Fig. 28, a set of processing engines (PE) are connected to the line cards through a switch. The line cards that terminate the physical links abstain from doing any packet processing and just forward the packets to the PE's through the switching fabric. A meta-network may use one or more than one PE's for packet processing. Details of the isolation of the switching fabric and other architectural details can be found in [202].

Developing specialized substrate nodes as discussed in [204] shall take considerable amount of time, effort and research to develop. Also, such substrates present only in research facilities shall greatly constrain the magnitude and realism of experiments. A short-term solution that can allow similar experimentation flexibility over substrate nodes in campus networks is proposed in [117,98]. To be able to do so, substrate production nodes in campus networks need to provide an open, programmable virtualized environment for researchers to be able to install and run their experiments. However, this approach has two problems. Network administrators shall not be comfortable to allow running experimental code on production routers or switches and commodity router and switch manufacturers are ever reluctant to divulge the technology that sits inside their high-end products, thus providing no chance for virtualization, either software or hardware.

To break this impasse, an open-flow switch has been designed that (1) provides complete isolation of production traffic from experimental traffic thus easing the anxiety of network administrators and (2) does not require commodity hardware manufacturers to open their internal architecture except for incorporating the open-flow switch into their hardware. The design of the switch takes advantage of TCAM (Ternary Content-Addressable Memory) based flow tables used mostly by all routers and switches. The idea is to identify incoming packets based on flow parameters (IP addresses, ports, etc.) and take appropriate action as directed in the flow table for a packet belonging to a certain flow. The action can be as simple as forwarding the packet to a particular port (for production traffic) or encapsulating and forwarding the packet to the controller (for the first packet of any flow or for a certain experimental traffic). The exact details of the switch specification are beyond the scope of the current discussion and may be found at [98].

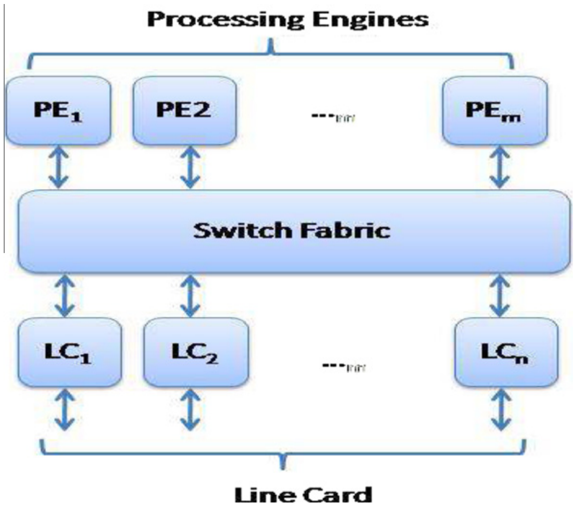

Fig. 28. Architecture of a programmable router design.

The virtualization techniques discussed in these two schemes are in addition to the various other schemes of virtualization of end systems through virtual machine or virtual server techniques. However, these virtualization techniques do not support the needs of wireless environment. The key problems are: (1) **Isolation**: While it is not possible to over-provision the wireless bandwidth, the scarcity of the wireless bandwidth resource forces new partitioning models to be able to support a reasonable number of isolated experiments and (2) **Uniqueness of nodes**: wireless signal propagation is a node specific property (coding, multiplexing, etc.) and difficult to control. Some techniques for virtualization of wireless network elements are discussed in [165]. Some of the techniques for sharing the wireless resources are: (1) Frequency Division Multiple Access (FDMA), (2) Time Division Multiple Access (TDMA), (3) Combined TDMA and FDMA: Virtualize the node by allowing different users to use given frequency partition for a specific period of time, (4) Frequency Hopping: Virtualize the node by allowing different users to use different frequency partitions at different time slots, and (5) Code Division Multiple Access (CDMA): Each user is give a unique and orthogonal code and is allowed to use the entire frequency for the entire time without interference with each other.

Using a combination of these virtualization techniques, a wireless testbed may offer sliceability through (1) Space Division Multiple Access (SDMA): A node with a fixed wireless range is dedicated fully to a user and partitioning is done using spatial separation of multiple nodes in the testbed, (2) combined SDMA and TDMA: The nodes are spatially separated and also each node is partitioned using TDMA creating time slots, (3) combined SDMA and FDMA: the nodes are separated spatially and each node is partitioned using FDMA, creating frequency partitions, and (4) combined SDMA, TDMA and FDMA: The nodes are spatially separated, and each node is partitioned by frequency partitions and each frequency partition is partitioned into time slots.

Thus, virtualization is widely accepted to be the basis for enabling a flexible Internet architecture for the future that would accommodate multiple architectures and allow disruptive innovations and technologies to be easily incorporated into the core architectures. As for the present, testbed designs based on virtualization concepts serve, both as a proof-of-concept for virtualizable Internet architecture of the future and a hosting substrate for testing of disruptive technologies for the future.

9.3. Next generation network testbeds: implementations

The two biggest efforts in this direction are the GENI (Global Environment for Network Innovations) [118] effort in the US and the FIRE (Future Internet Research and Experimentation) [119] effort in Europe. While the primary GENI objective is to make a dedicated shared substrate facility available for large scale and long-lived experiments, the primary focus of the FIRE project is to federate multiple existing network testbeds in Europe (as a result of prior programs) and provides a large multi-context realistic testbed available for research. In the next two subsections, we shall briefly discuss the GENI and FIRE projects limiting our scope to the GENI substrate architecture and FIRE federation efforts.

9.3.1. Global Environment for Network Innovations (GENI)

GENI or Global Environment for Network Innovations is an effort by the National Science Foundation (NSF) in the United States to design and implement a network testbed to support at-scale experimentation on shared, heterogeneous long-lived and highly instrumented infrastructure [57]. GENI shall have its own dedicated backbone link infrastructure through partnerships with the National LambdaRail [120] and the Internet2 [121] projects. GENI is also expected to federate with a wide range of other infrastructural facilities to add to its diversity and support for realism. In the rest of this subsection on GENI, we first discuss the key GENI requirements, the generalized GENI control framework and finally we look into the five different cluster projects, each developing a prototype control framework for GENI underlying the components of the generalized GENI control framework.

GENI requirements. GENI comprises of a set of hardwire components including computer nodes, access links, customizable routers, switches, backbone links, tail links, wireless subnets, etc. Experiments on GENI shall run on a subset of these resources called a “slice”. In general, two types of activities shall be supported over the GENI testbed: (1) deployment of prototype network systems and observing them under real usage and (2) running controlled experiments. Some of the key requirements for the GENI infrastructure are:

1. Sliceability: In order for GENI to be cost-effective and be able to cater to as many experimental requirements as possible, GENI shall need to support massive sharing of resources, at the same time ensuring isolation between experiments.
2. Programmability: GENI is a testing environment needing generality. All GENI components need to be programmable so that researchers are able to implement and deploy their own set of protocols at the component level.
3. Virtualization and resource sharing: Sliceability entails sharing of resources. A common form of resource sharing is through virtualization techniques, wherever possible. However, for some resources, owing to the some inherent properties of the resource (e.g., an UMTS link can support only one active connection at a time), other methods such as time-shared multiplexing may be employed.
4. Federation: The GENI suite is expected to be a federated whole of many different parts owned and managed by different organizations. Federation also adds diversity to the underlying resource pool, thus allowing experiments to run closer to real production systems.
5. Observability: One of GENI's core goals is to provide highly instrumented infrastructure to support accurate measurements of experiments. Hence, the GENI design should allow an efficient, flexible, robust and easily specifiable measurement framework.
6. Security: GENI is expected to run many disruptive and innovative protocols and algorithms. Also, GENI experiments may be allowed to interact with existing Internet functionality. Hence, security concerns require that GENI nodes cannot harm the production Internet environment, either maliciously or accidentally.

Several other requirements and detailed discussions can be found in the GENI design documents [8,186,38,175,22,18,80]. However, the key value proposition of GENI that separates it from smaller scale or more specific testbeds are:

1. Wide scale deployment – access not restricted to those who provide backbone resources to GENI.
2. Diverse and extensible set of network technologies.
3. Support for real user traffic.

In the rest of this discussion on GENI, we focus specifically on the control architectural framework of GENI and also look at some of the protocol designs that are being undertaken as the first phase of prototype design.

GENI generalized control framework. Before looking at the specific prototype designs for the GENI generalized control framework in Fig. 29, we need to look at the generic GENI control framework as defined in [58]. GENI consists of several subsystems:

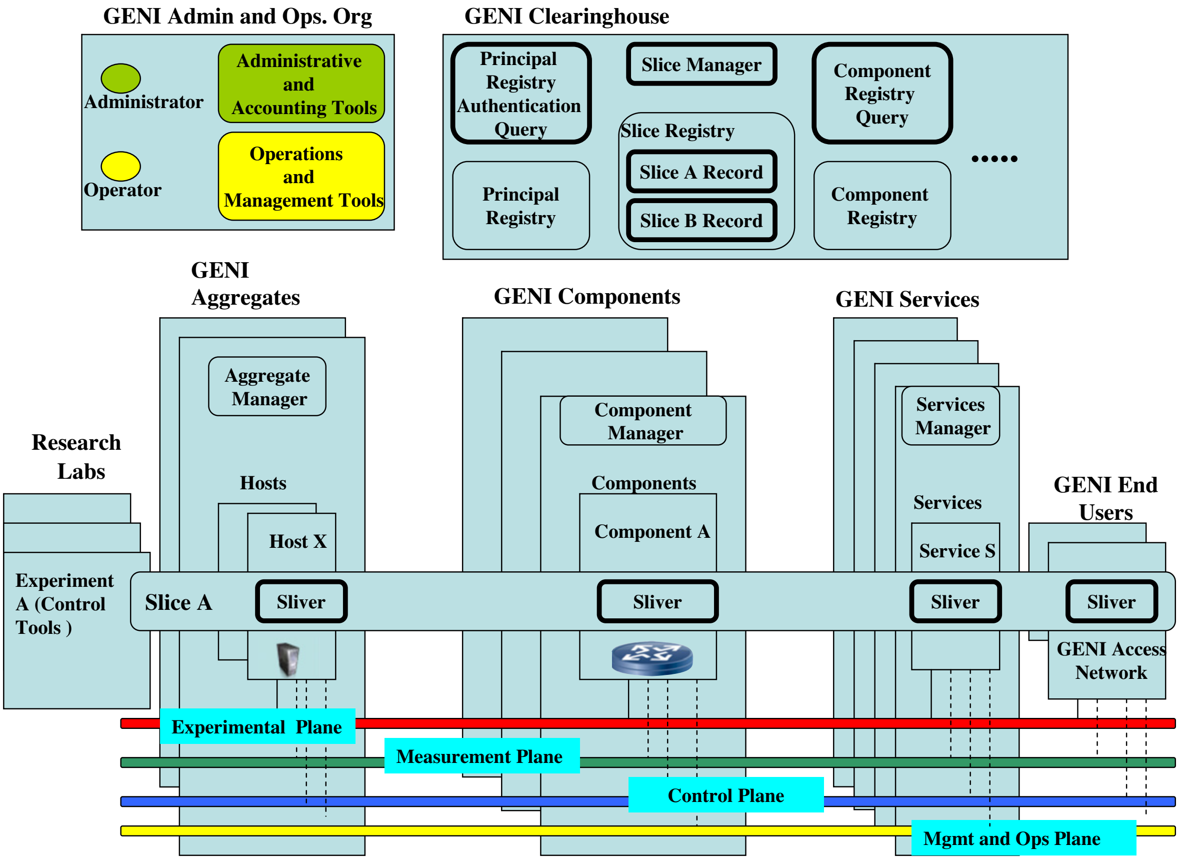

Fig. 29. GENI: generalized control framework and constituent subsystems.

1. Components and aggregate components: A device which hosts a set of resources is called a component. The resources of a component may be shared through virtualization or other methods among multiple experiments such that they satisfy the properties of programmability, isolation, and security. A set of components under a central control is called an aggregate. A component may belong to one or more such aggregates.
2. Clearinghouses and control framework: A clearinghouse is a centralized registry that maintains the information for principles, slices, and components. This information in the registries may be used to drive access control policies, control policies, trust mechanisms and federation mechanisms for the components or the aggregates within its scope of control.
3. Measurement subsystem: The measurement subsystem satisfies the "Observability" goal of GENI. It provides a framework for measurement, archival and retrieval of experimental data.
4. Administration and operations: This subsystem provides tools, services, and technical support for enabling incorporation of new resources into GENI, identifying and managing misbehaving resources and assisting researchers using GENI.
5. Experimenter tools and services: This subsystem provides support tools for easy experiment deployment and execution. These tools include functionalities such as resource discovery, resource reservation, designing, composing, debugging, instrumentation, and access policies.

Apart from the components discussed above, in GENI control framework, each aggregate has a Aggregate Manager (AM) and every component has a Component Manager (CM). Also, the clearing house has a Slice Manager (SM) that can reserve slices for a particular experiment. Also, the control framework defines (1) Interfaces between the entities, (2) Message types, (3) Message flow between entities to realize an experiment, and (4) a control plane for transporting messages between entities. More details of the control framework of GENI can be found at [58].

GENI control framework: prototype clusters. The GENI generalized control framework defines the entities, interfaces, and semantics for running experiments within a sliced, federated suite of infrastructure. However, the exact nature of the control activities, the design of the control plane and its implementation are still under active consideration. As such, under the spiral 1 [122], the GENI has set up five clusters, with each cluster responsible to implement and deploy a prototype implementation of a control mechanism suitable to be incorporated as the control mechanism of the GENI control framework. These five clusters are: (1) Cluster A – TIED, (2) Cluster B – Planetlab, (3) Cluster C – ProtoGENI, (4) Cluster D – ORCA, and (5) Cluster E – ORBIT. The discussion is restricted to discussing the control framework design and the federation mechanisms in each cluster prototype development. Ancillary projects within each cluster developing aggregates, virtualized nodes, etc. are beyond the scope of the current discussion.

Cluster A: Trial Integration Environment with DETER (TIED) control framework. The "Cluster A" GENI prototype uses the DETER [123,19] control framework and designs a federation architecture for the security experiment testbeds anticipating the GENI control framework. DETER is an Emulab based testbed architecture extended to specifically support robust experiment isolation for cyber-security experimentation [124]. Cyber-security experimentations enforce added concerns of security in which an experiment may try to break-free from its isolated environment and attack other experiments, testbed control hardware and also the Internet. Malicious code running as experiments inside the testbed with root access on the nodes can spoof its IP or MAC address. Hence, isolation needs to be implemented right at layer 2 of the protocol

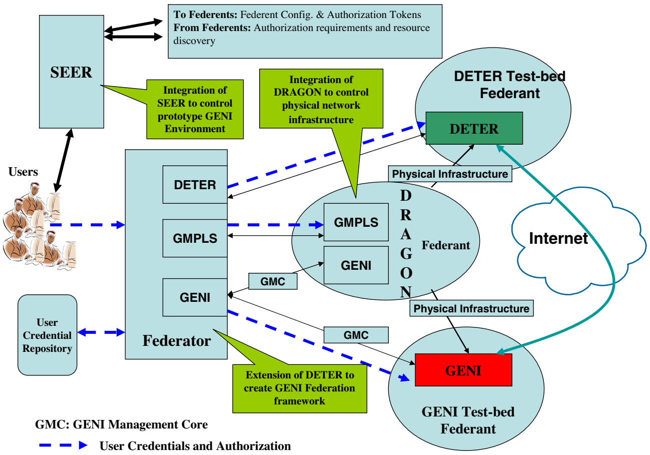
To Federents: Federent Config. & Authorization Tokens
From Federents: Authorization requirements and resource discovery
SEER
Integration of SEER to control prototype GENI Environment
Integration of DRAGON to control physical network infrastructure
DETER Test-bed Federant
DETER
Users
DETER
GMPLS
GENI
Federator
Physical Infrastructure
GMPLS
GENI
DRAGON
Federant
Internet
GMC
GMC
User Credential Repository
Physical Infrastructure
GENI
Extension of DETER to create GENI Federation framework
GENI Test-bed Federant
GMC: GENI Management Core
User Credentials and Authorization

Fig. 30. TIED federation mechanism.

stack. DETER handles this through VLAN (Virtual LAN) technology and switched Ethernet connectivity. The details of the exact architecture of the DETER testbed can be found at [19]. Thus, DETER supports a secure sliceabile capability which ensures strict isolation of experiments running on a common substrate [87]. Also, a federation model of DETER with other Emulab [109] based testbeds, such as WAIL [110], can be found at [49].

Federation architecture: TIED proposes a dynamic federation architecture through a federator module mediating between distributed researchers and distributed, diverse testbed environments. A user is supposed to specify his experimental requirements in some high-level constructs which are mapped to experiment topology, resource requirement, etc. by an experiment creation tool. The experiment creation tool may also have as inputs, the specific properties of testbeds in the federated environment. The experiment creation tools finally submit an "experiment representation" to the "Federator." The federator is responsible to set-up a coherent experiment across resources from multiple testbeds, abstracting the specific control and management heterogeneity from users. A diagrammatic representation of the TIED federation architecture can be seen in Fig. 30. SEER [182] is the Security Experimental Environment for DETER which comprises of various tools integrated to ease the configuration of security experiments by researchers, while DRAGON [125] allows inter-domain dynamic resource allocation across multiple heterogeneous networking technologies. Details of SEER and DRAGON are beyond the scope of the present discussion and an interested reader is encouraged to follow the references to know more about them.

As part of the spiral 1 prototype development effort, TIED undertakes the following activities [59] (1) Development and deployment of TIED component manager and clearinghouse packages, (2) operate a clearinghouse prototype for TIED, (3) provide GENI users access to TIED testbed. Thus, TIED allows GENI prototype developers to use TIED clearinghouse and component implementations in their own aggregate mangers leveraging the TIED federation architecture and also the secure and controlled experimental environment provided for security experiments in DETER.

Cluster B: PlanetLab control framework. "Cluster B" utilizes the Planetlab control framework. While the Planetlab control framework is extended to coalesce with the GENI control framework and realize the GENI design goals of federation and sliceability, the rest of the six projects are involved in designing substrate nodes with diverse capabilities for resource sharing and isolation, and their corresponding component managers.

Planetlab has already been discussed in Section 9.1. The "cluster B" prototype development effort enhances the control framework for Planetlab and extends it to be able to coherently federate all slice based architecture network substrates [169] such as PlanetLab, VINI, Emulab and GENI. The various enhancements are implemented through a GENI wrapper module [170] that bundles an aggregate manager, slice manager and a registry into the PlanetLab Control (PLC) and also a Component Manager to individual nodes (nodes in PlanetLab correspond to components of GENI [169]).

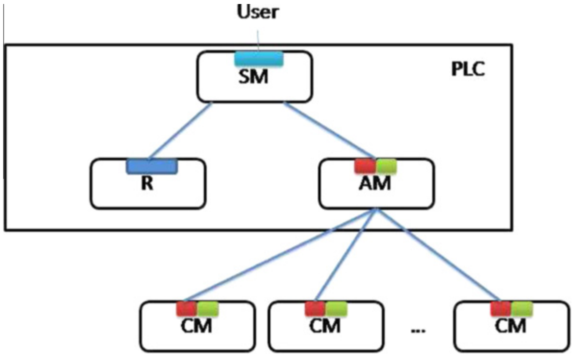

Fig. 31. Plain Vanilla implementation of GENI wrapper.

The plain Vanilla Planetlab implementation of the GENI wrapper is shown in Fig. 31. Users setup a slice by interacting with the slice manager (SM). The slice manager contacts the registry (R) to get the necessary credentials and then contact the slice manager interface of the aggregate manager (AM) to create and control the slice. The Aggregate manager communicates to the individual components through the component manager's (CM's) slice management interface.

Federation architecture: Based on the Vanilla PlanetLab implementation, federation with other slice based architectures may be architected as follows:

1. Alternative slice manager: As shown in Fig. 32 for the case of federation between PlanetLab and Emulab, the Emulab Slice Manager contacts the PlanetLab Registry to retrieve the credentials, then it contacts the PlanetLab Aggregate Manager to retrieve a ticket for each slice and finally it redeems those tickets directly with the PlanetLab nodes through the component managers.
2. Common registry: As shown in Fig. 33, A common registry is maintained between the federating entities, PlanetLab and Emulab, such that the credentials are commonly maintained at the PLC and Emulab may retrieve these credentials and use it to create slices purely on Emulab nodes through the Emulab aggregate manager.
3. Multiple aggregates: As shown in Fig. 34 for the case of Planetlab and VINI, PlanetLab Slice Manager retrieves credentials from the common registry and may use these credentials to create slices through the Aggregate managers of both PlanetLab and VINI. This results in a federation where users are allowed to

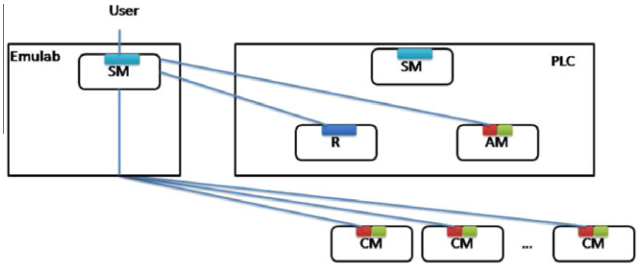

Fig. 32. PlaneLab Emulab Federation: alternative slice manager.

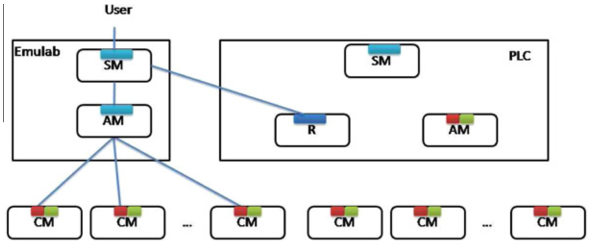

Fig. 33. PlaneLab Emulab Federation: common registry.

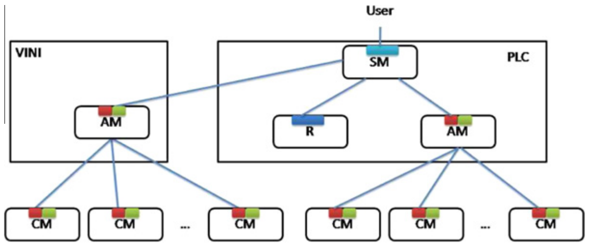

Fig. 34. PlanetLab VINI Federation: multiple aggregates.

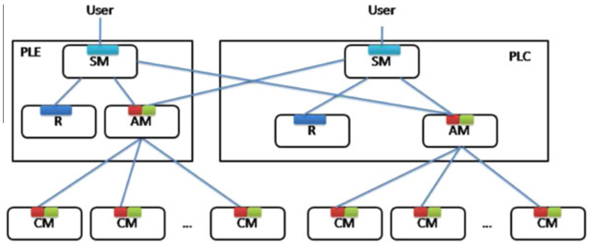

Fig. 35. PlanetLab PlanetLab-Europe Federation: full aggregation.

run their experiments spanning multiple diverse testbeds such that one of the testbeds (in this case VINI) not implementing any Registry or Slice management functionality.

4. Full aggregation: As shown in Fig. 35, full federation involves both the federating parties maintaining their own registries. This allows a "multiple aggregate" scenario wherein each federating party is functionally independent from each other, implementing it's own slice manager, aggregate manager and registry. Users belonging to one testbed may create and control components from both the testbeds.

Cluster C: ProtoGENI control framework. The control framework in ProtoGENI [126] is an enhanced version of the Emulab control software. The ProtoGENI clearinghouse [127] has been designed to allow it to be shared by all members of the ProtoGENI federation as shown in Fig. 36 and performs the following two functions: (1) Allows users to find components and (2) acts as a central point of trust for all members in the federation.

Federation architecture: Each member of the ProtoGENI federation is an Emulab installation site and has to have a self-generated and self signed root certificate. This certificate becomes the identity of the federated site within ProtoGENI. Thus a "web of trust" is formed between all the members of the federation. A user is provided with an SSL certificate issued by its local Emulab instance which authenticates the user to the entire federation. Certification Revocation Lists (CRL) are sent by each member of the federation to the Clearinghouse, where they are combined and sent out to each member of the federation. The Aggregate Manager of ProtoGENI is implemented by placing the Component Manager API code on top of the Emulab software. Thus, this makes any site running the latest version of Emulab code to join the federation quite easily. It may be noted that the federation concepts of ProtoGENI is in contrast to that of the Planetlab federation concept that allows federation between any two testbeds that implement a slice based architecture.

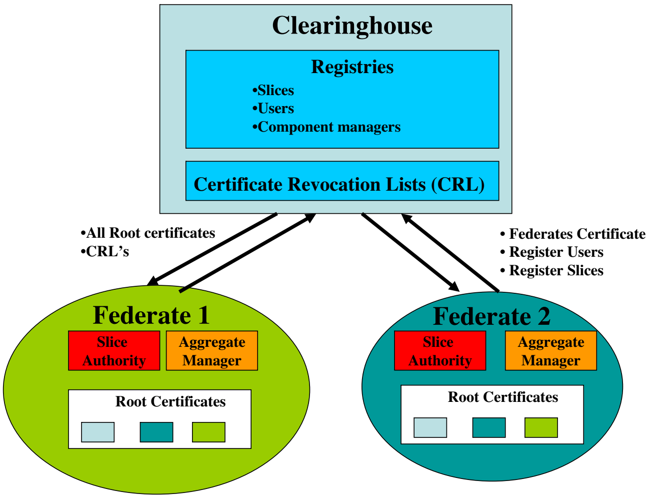

Fig. 36. ProtoGENI control framework.

Cluster D: Open Resource Control Architecture (ORCA) control framework. The "Cluster D" GENI prototype development plan involves the extension of ORCA (a candidate control framework for GENI) [128] to include the optical resources available in BEN (Breakable Experimental Network). ORCA is a control plane approach for secure and efficient management of heterogeneous resources [129]. ORCA is different from traditional resource management schemes based on middlewares operating between the host operating system supplying resources and the applications requesting them. ORCA defines a paradigm of resource management wherein the resource management of ORCA runs as an "underware" [37] below the host operating system. ORCA uses virtualization to allocate "containers" over which a resource requester may install its own environment. Hence, as shown in Fig. 37, the ORCA control plane may be viewed as an "Internet Operating System" supporting a diverse set of user environments on a common set of hardware resources.

Also, as shown in Fig. 38, the ORCA "underware" control plane allows federation of various heterogeneous underlying resource pools, each with their own set of resource allocation policies.

Federation architecture: The implementation of federation of diverse resource pools is architected through Shakiro [73] resource leasing architecture based on the SHARP [56] secure resource peer-

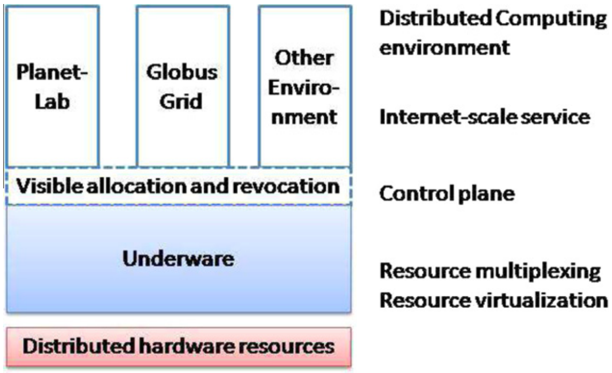

Fig. 37. ORCA control plane.

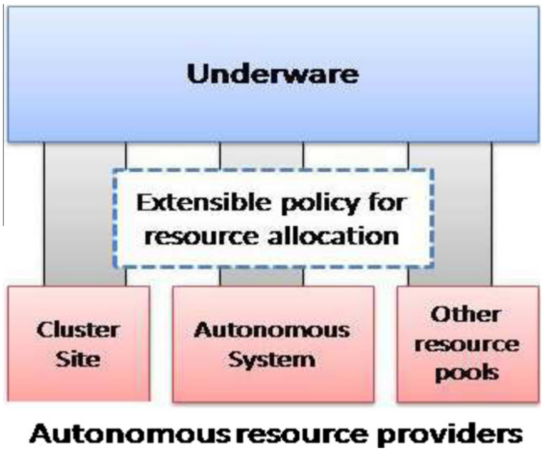
Underware
Extensible policy for resource allocation
Cluster Site
Autonomous System
Other resource pools
Autonomous resource providers

Fig. 38. ORCA underware.

ing framework. Each SHARP resource has a type with associated attributes and available quantity. As shown in Fig. 39, the site exports a leasing service interface. An application specific service manager may make resource request through the lease API to the broker. The broker matches the requirements and issues tickets for particular resource types, quantity and location. The service manager may then redeem the tickets with the site-leasing service interface which allocates resources and sets them up.

Cluster E: the ORBIT control framework. The "Cluster E" GENI prototype is based on the extension of control and management framework (OMF) [132] of ORBIT to suit the GENI compliant control framework. ORBIT [130,131] is a unique wireless networking testbed which comprises of (1) A laboratory based wireless network emulator for an initial, reproducible testing environment, and (2) Real-world testbed environment of wireless nodes (mix of 3G and 802.11 wireless access) for field validation of experiments.

The OMF is the control and management framework for ORBIT. As shown in Fig. 40, the user end has an "Experiment Controller" component that is responsible for controlling an user experiment, translating an experiment description to resource requirement and communicating with the resource manager for allocation and control or required resources. The OMF has three primary components: (1) The aggregate manager – responsible for the overall management of the testbed, (2) resource manager – exists on every Resource and manages various aspects of the resource, and (3) resource controller – communicates with an experiment controller to control the part of the resource committed to an experiment. Finally, a centralized database stores and retrieves experimental measurement data.

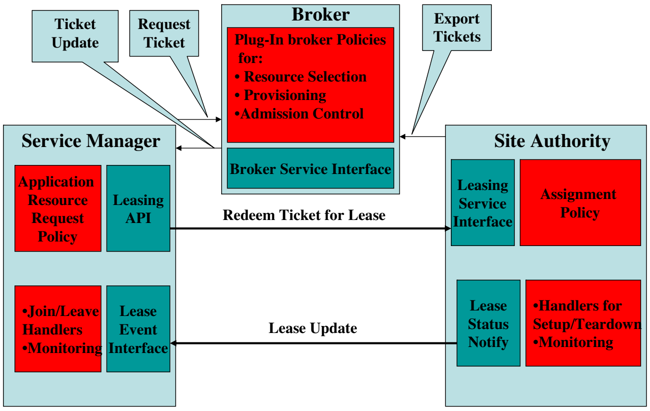

Fig. 39. ORCA federation mechanism.

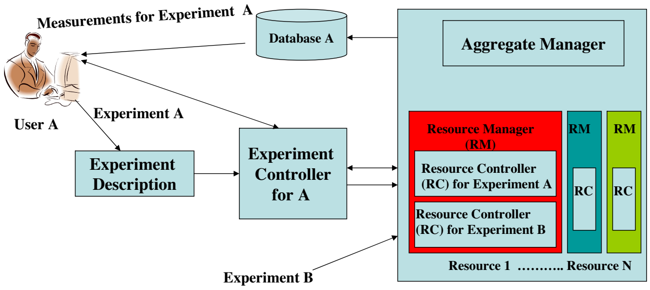
Measurements for Experiment A
Database A
Aggregate Manager
User A
Experiment A
Experiment Description
Experiment Controller for A
Resource Manager (RM)
Resource Controller (RC) for Experiment A
Resource Controller (RC) for Experiment B
RM
RC
RM
RC
Resource 1 ……….. Resource N
Experiment B

Fig. 40. ORBIT control and management framework (OMF).

The OMF is a candidate control framework for GENI and hence the OMF design is being extended to: (1) support multiple heterogeneous hardware, (2) support resource virtualization to support multiple experiments sharing a resource, (3) federate multiple testbeds, and (4) add dynamic steering of experimental control.

Federation architecture: As part of the Spiral 1 effort, OMF is being extended to support multiple heterogeneous testbeds in accordance with the GENI control framework [133]. Already, OMF has been extended to support mobile testbeds by defining methods to: (1) distribute experiment scripts to mobile nodes, (2) cache experimental measurement data locally on the node in cases of disconnection, and (3) perform experiment actions at predefined points in time. This extension of OMF is aimed to demonstrate the capability of OMF to support multiple heterogeneous testbeds and thus concur to the GENI design requirements.

9.3.2. FIRE testbeds

The counterpart of the GENI effort in the US is the Future Internet Research and Experimentation (FIRE) effort of the European Union.

A diverse set of testbeds for networking experimentation and testing, in various contexts of access technologies, engineering motivations and layered and cross-layered architecture validation, were developed as part of various past research efforts in Europe. The basis of most of this work relates back to the GEANT project [134] which was undertaken to connect 30 National Education and Research Networks (NREN's), spread across Europe through a multi-gigabit network dedicated specifically for research and educational use. The GEANT network thus provides the high bandwidth infrastructure to be shared among various research projects ranging from grid computing to real time collaborative experimentation support. Also, multiple cutting edge network services, such as IPv6, IP with QoS, multicasting, premium IP (prioritized IP based services), have been implemented and are available over GEANT. Hence, GEANT is not a testbed but a production level network infrastructure serving the research community in Europe, much in the spirit of the original NSFNet, LambdaRail [120], CSENET or Internet2 [121] networks in various other parts of the world.

A discussion of GEANT was essential in the present context because the European effort for infrastructure development for the next generation Internet experimentation and testing is mostly focused on efforts towards the federation of diverse individual testbeds over the GEANT infrastructure facility. Federation is defined as "a union comprising a number of partially self-governing regions united by a central federal government under a common set of objectives" [159]. Fig. 41 shows various testbed development research projects that were undertaken as part of the Framework 6 program, most of which are either complete or almost reaching completion [135].

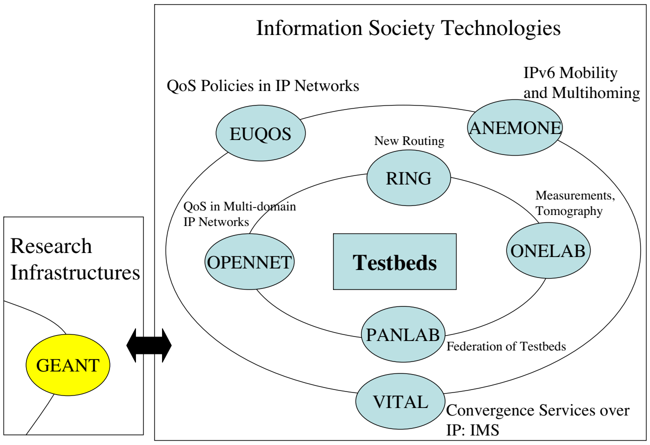
Information Society Technologies
QoS Policies in IP Networks
IPv6 Mobility and Multihoming
EUQOS
ANEMONE
New Routing
RING
QoS in Multi-domain IP Networks
Measurements, Tomography
Research Infrastructures
OPENNET
Testbeds
ONELAB
PANLAB
Federation of Testbeds
GEANT
VITAL
Convergence Services over IP: IMS

Fig. 41. Overview of FP6 projects.

These projects are expected to serve as the foundations for the FIRE facility with the projects such as Onelab 2 [112], Panlab II [136], VITAL++ [137], and WISEBED [138], exploring ideas for the federation of these facilities into a single large experimental facility. Another project, FEDERICA [139], is aimed at developing an end-to-end isolated experimental facility over dedicated high speed links provisioned over existing educational and research networks. FEDERICA has similar "sliceability" objectives as that of GENI. As shown in Fig. 42, while the other FIRE projects mainly concentrate on federation aiming to support experimentation on a diverse and rich set of underlying technologies, FEDERICA is more of a virtualization proposal aimed at allowing end-to-end disruptive innovations in architecture and protocol design.

In the rest of this section, we shall discuss the virtualization concepts of FEDERICA followed by the federation mechanisms of Onelab 2, PANLAB and PII, and WISEBED.

FEDERICA. FEDERICA [52,53] connects 12 PoPs (Point of Presence) using high speed (1 Gbps) dedicated circuit infrastructure provisioned via the education and research infrastructure of GEANT2 [140] and NRENs and virtualization techniques to create a "slice" consisting of virtual circuits, virtualizable and programmable switches and routers, and virtualizable computing resources. The "slice" is the fundamental unit of allocation for a user's experimental needs. The substrate just creates the necessary resource configuration and is completely agnostic about the protocol, services and applications running on them. For those users wishing to test a distributed application may request a set of virtual routers and hosts pre-configured with IP and those users wanting to test a novel routing protocol may request a set of virtual hosts and routers interconnected over Ethernet circuits forming a specified topology. In fact, the FEDERICA approach of resource sharing and end-to-end experiment isolation is very similar to the proposals of a diversified Internet architecture discussed in [204,7].

FEDERICA has four core sites and 8 on-core sites. The core sites are connected into a full mesh topology through high-speed (1 Gbps) GEANT2 infrastructure links. The core allows only direct dedicated channels between the core switches making the core highly resilient and efficient. The core also allows BGP peering with the global Internet subject to security restrictions. Non-core POP's do not have the strict requirement of direct connection. Hence, non-core POP's can connect to FEDERICA via the GEANT2 infrastructure, via NRENs or via the public Internet. Also there is another group of POP's called collaborative POPs. Collaborative POPs do not provide guaranteed resources to the infrastructure and also they are managed and controlled by their owners.

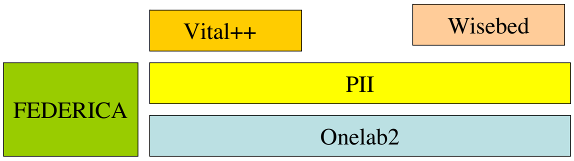

Fig. 42. Relationship amongst various FIRE projects.

A major difference between FEDERICA and other similar efforts such as GENI is that, FEDERICA is much more modest in terms of size and diversity. The only objective of FEDERICA is to develop an end-to-end isolated testing environment to be able to support innovative and disruptive network experimentation. As a result, FEDERICA will be available for researchers much sooner than any of the other similar testbed design efforts.

OneLab2. OneLab2 is an extension of OneLab and has a focus on research using open source tools and softwares. It is primarily non-commercial and hence the primary challenges for federation are technical rather than political. Also, as discussed in [160], an economic incentive based model needs to be developed to increase the resource contribution by each participating site. Resource provisioning in Planetlab currently follows a minimum fixed contribution rule wherein each site needs to contribute at least 2 nodes to be a part of the system. The allocation policies of Planetlab restrict each site from having at most 10 slices. However, since each slice has unrestricted access to resources irrespective of the number of nodes they contribute to the system, these allocation policies are not economic-centric in the sense that there does not exist enough incentive for a site to provision more resources for the system. To develop effective economic incentive models, wherein allocation is somehow related to contribution, the first step is to develop a metric for evaluating the value of a site through characterization of the resources offered. A suggestion [171] is to characterize resources based on three broad characteristics: (1) Diversity (technology, number of nodes, etc.), (2) Capacity (CPU, bandwidth, memory, etc.), and (3) Time (duration, reliability, etc.).

Federation mechanism: The present federation policies between Planetlab and Planetlab-Europe are that of "peering" wherein users from both facilities have the same access rights over the whole infrastructure and both facilities apply the same local policy. However, pairwise federation leads to the common full mesh "$n \times n$" scalability problems, with "n" being the number of federating sites. The problem worsens with plans to have large scale localized federations across heterogeneous networking contexts as discussed in Section 9.2.1. This calls for a hierarchical federation architecture in which an instance of PlanetLab central federates with various regional/local/personal PlanetLab instances which in turn federate with local testbeds [171]. Also, another model of federation could be based on Consumer-Provider relationship in scenarios wherein the users of one local federation form a subset of users of a larger federation. Hierarchical federation policies, however, introduce the added concerns of "Local Vs Global Policy" enforcements.

PANLAB and PII (Panlab II). PANLAB is the acronym for Pan European Laboratory and is mostly a consortium of telecom service providers across Europe. It was an effort to federate distributed test laboratories and testbeds across Europe to provide a diverse and heterogeneous facility for large scale networking experiments. It provides a realistic testing environment for novel service concepts, networking technologies and business models prior to their launching into production environments.

Federation mechanism: The main challenges for the creation of Panlab involve defining an architecture for diverse contextual platforms to be able to federate across a seamless homogenized platform accessible to its users. PANLAB takes an evolutionary approach, moving towards higher degree of automation in the management and control functions of the testbed. Fig. 43 shows the third and final level of this evolution. The three phases of evolution are:

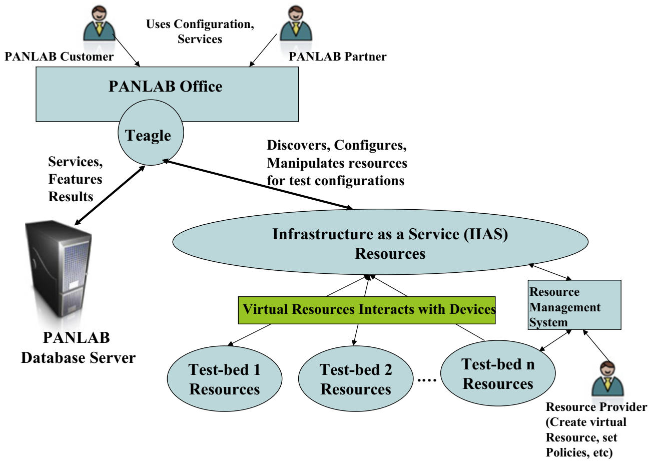

Fig. 43. PANLAB federation final phase: on-demand configuration approach.

1. **Centralized approach**: This is the first phase. Each partner site shall have to fill up a web form manually detailing the testbed descriptions and resources available for sharing. This form is provided by a web-based search services called Teagle. Users wishing to run an experiment submit the nature of the experimental requirements to Teagle. Teagle looks up the repository of testbed meta-data and tries to find a match.
2. **Manual configuration approach**: In this phase, the partner sites advertise the testbed meta-data by using a specialized middleware and expose a "Infrastructure as a Service (IaaS)" interface [Fig. 43]. Teagle will search the repository as well as query this service for required resources. In this phase, resources are virtualized and, hence, the IaaS may hide the actual location of a resource from the user providing infrastructure from one or more partner sites.
3. **On-demand configuration approach**: In this final phase of evolution shown in Fig. 43, Teagle will establish an on-demand testbed according to the user requirement by directly interacting with the virtualized resources. Teagle provides a best effort configuration and the users need to directly access the resources for complex configurations.

PANLAB also proposes the use of IMS (Internet Protocol Multimedia subsystem) to support the control plane of the federation. PII or PANLAB II is an extension of PANLAB and includes a federated testbed of four core innovative clusters and three satellite clusters [136]. PII takes a more holistic view of federation by considering the breadth of technological, social, economical and political considerations of the federation.

9.3.3. WISEBED

The WISEBED project [213] is aimed at federating large scale wireless sensor testbeds to provide a large, diversified, multi-level infrastructure of small-scale heterogeneous devices. An Open Federation Alliance (OFA) is defined that develops open standards for accessing and controlling the federation. WISEBED classifies the diverse testbeds into two categories: (1) Fully integrated: The testbed defines a full range of services as defined by the OFA and (2) semi integrated: Provides sunset of the service defined in the OFA. Another classification based on the access to the testbed also consists of two categories: (1) Fully Accessible: users can access the testbed data and also re-program the testbed devices and (2) semi accessible: Users are only permitted to extract experimental data from the testbed.

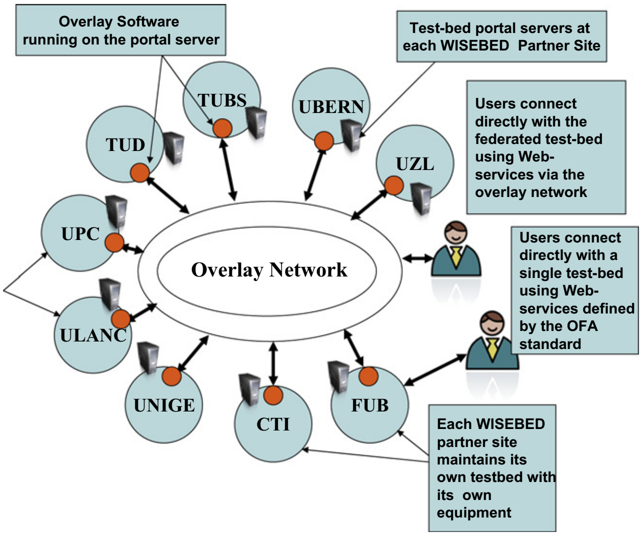

Fig. 44. Overall architecture of WISEBED testbed federation.

Federation mechanism: As shown in Fig. 44, WISEBED federates multiple wireless sensor node testbeds comprising of a diverse range of hardware and software technologies. The federation mechanism of WISEBED consists of a hierarchy of layers, with each layer comprising of one or more peers. The bottom layer consists of a network of wireless sensor nodes belonging to diverse hardware and software technologies. Each testbed exposes a web-based portal through which users may deploy, control and execute experiments. These portal servers form the second layer of the WISEBED federation architecture. The third and final layer is an overlay of the portal servers. Each portal server exposes its services through an identical interface allowing the federation to expose an unified virtual testbed to the users. Each site participating in the federation needs to expose Open Federation Alliance (OFA) standardized interfaces for accessing and controlling the testbed.

Fig. 45 presents a high-level view of the portal servers. The portal servers are responsible for the control, management and measurements of a single site. The inner layer consists of services that can communicate with hardware sensor devices through gateways to the wireless networks. User commands are translated into a generic binary packet format that can be understood by the wide and diverse wireless substrate technologies of the testbed. Also, each portal server is connected to one or more, local data stores for storing measurement data. The "outer layer" exposes service interfaces for users to access the testbed through the portal server.

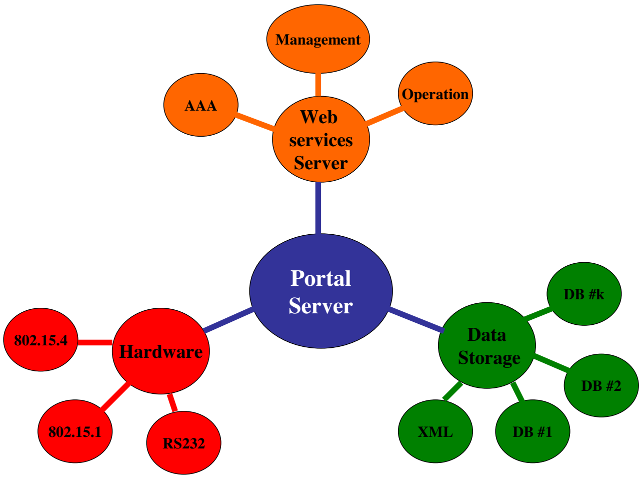

Fig. 45. WISEBED: high-level view of portal servers.

Either these portal servers or a separate overlay node running client services to the portal server in its "inner layer" and exporting portal server interface in its outer layer, run an overlay software to form the federate with other sites. An user requiring uses federated resources may connect using OFA standard web services through the overlay.

10. Conclusions

A number of industry and government funding agencies throughout the world are funding research on architecture for future networks that are "clean-slate" and are not bound by the constraints of the current TCP/IP protocol suite. In this paper, we have provided an overview of several such projects. National Science Foundation (NSF) in the United States started a "future Internet design (FIND)" program which has funded a number of architectural studies related to clean-slate solutions for virtualization, high-speed routing, naming, security, management, and control. It also started the Global Environment for Network Innovations (GENI) program that is experimenting with various testbed designs to allow the new architectural ideas to be tested.

The Future Internet Research and Experimentation (FIRE) program in Europe is also looking at future networks as a part of the 7th Framework program of the European Union (FP7). Another similar study is the AKARI program in Japan.

In addition to the above, Internet 3.0 is an industry funded program that takes a holistic view of the present security, routing, and naming problems rather than treating each of them in isolation. Isolated clean-slate solutions do not necessarily fit together, since their assumptions may not match. Internet 3.0, while clean-slate, is also looking at the transition issues to ensure that there will be a path from today's Internet to the next generation Internet.

NSF has realized the need for a coherent architecture to solve many related issues and has recently announced a new program that will encourage combining many separate solutions into complete architectural proposals.

It is yet to be seen whether the testbeds being developed today, which use TCP/IP protocol stacks extensively, will be able to be used for future Internet architectures that have yet to be developed.

In this paper, we have provided a brief description of numerous research projects and hope that this will be a useful starting point for those wishing to do future network research or simply to keep abreast of the latest developments in this field.

11. List of abbreviations

4D	Data, discovery, dissemination and decision
AKARI	"a small light in the dark pointing to the future" in Japanese
ANA	Autonomic network architecture
AS	Autonomous system
ASRG	Anti-Spam Research Group (of IRTF)
BGP	Border Gateway protocol
CABO	Concurrent Architectures are Better than One
CCN	Content Centric Networking
CDN	Content Distribution Network
CONMan	Complexity Oblivious Network Management
CTS	Clear to send
DAN	Disaster day after networks
DFT	Delay/fault tolerant
DNS	Domain name system
DONA	Data Oriented Network Architecture
DTN	Delay/disruption tolerant network
FEDERICA	Federated E-infrastructure Dedicated to European Researchers Innovating in Computing network Architectures
FIND	Future Internet design
FIRE	Future Internet Research and Experimentation
FP6	6th Framework Program
FP7	7th Framework Program
GENI	Global Environment for Network Innovations
GROH	Greedy routing on hidden metrics
HIP	Host Identity Protocol
HLP	Hybrid link state path-vector inter-domain routing
ID	Identifier
IIAS	Internet in a slice
INM	In-Network Management
IP	Internet Protocol
IRTF	Internet Research Task Force
ISP	Internet service provider
LISP	Locater ID Separation Protocol
MILSA	Mobility and Multihoming supporting Identifier Locater Split Architecture
NGI	Next generation Internet
NGN	Next generation network
NNC	Networking Named Content
NSF	National Science Foundation
OMF	ORBIT Control and Management Framework
ORBIT	Open-access Research Testbed
ORCA	Open Resource Control Architecture
PANLAB	Pan European Laboratory
PI	Provider Independent
PIP	Phoenix Interconnectivity Protocol
PLC	PlanetLab Control
PONA	Policy Oriented Networking Architecture
PTP	Phoenix Transport Protocol
RANGI	Routing Architecture for Next Generation Internet
RCP	Routing Control Platform
RTS	Ready-to-Send
SANE	Security Architecture for Networked Enterprises
SCN	Selectively Connected Networking
SLA	Service Level Agreement
SLA@SOI	Service Economy with SLA-aware Infrastructures
SMTP	Simple Mail Transfer Prototocol
SOA	Service-Oriented Architecture
SOA4ALL	Service-Oriented Architectures for All
SPP	Supercharged PlanetLab Platform
TIED	Trial Integration Environment with DETER
UML	User Mode Linux
WISEBED	Wireless Sensor Network Testbeds

References

[1] V. Aggarwal, O. Akonjang, A. Feldmann, Improving user and ISP experience through ISP-aided P2P locality, in: Proceedings of INFOCOM Workshops 2008, New York, April 13–18, 2008, pp. 1–6.

[2] New Generation Network Architecture AKARI Conceptual Design (ver2.0), AKARI Architecture Design Project, May, 2010, <http://akari-project.nict.go.jp/eng/concept-design/AKARI_fulltext_e_preliminary_ver2.pdf>.

[3] M. Allman, V. Paxson, K. Christensen, et al., Architectural support for selectively-connected end systems: enabling an energy-efficient future Internet, NSF NeTS FIND Initiative. <http://www.nets-find.net/Funded/ArchtSupport.php>.

[4] M. Allman, K Christenson, B. Nordman, V. Paxson, Enabling an energy-efficient future internet through selectively connected end systems, HotNets-VI, 2007.

[5] M. Allman, M. Rabinovich, N. Weaver, Relationship-Oriented Networking, NSF NeTS FIND Initiative. <http://www.nets-find.net/Funded/Relationship.php>.

[6] S. Androutsellis-Theotokis, D. Spinellis, A survey of peer-to-peer content distribution technologies, ACM Computing Surveys 36 (4) (2004).

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

[8] T. Anderson, L. Peterson, S. Shenker, et al., GDD-05-02: Report of NSF Workshop on Overcoming Barriers to Disruptive Innovation in Networking, GENI Design Document 05-02, January 2005. <http://groups.geni.net/geni/attachment/wiki/OldGPGDesignDocuments/GDD-05-02.pdf>.

[9] (Online) Anti-spam techniques wiki web page. <http://en.wikipedia.org/wiki/Anti-spam/techniques>.

[10] (Online) ASRG: Anti-Spam Research Group, Internet Research Task Force (IRTF) working group. <http://asrg.sp.am>.

[11] (Online) AVANTSSAR: Automated Validation of Trust and Security of Service-oriented Architecture, European Union 7th Framework Program. <http://www.avantssar.eu>.

[12] B. Awerbuch, B. Haberman, Algorithmic foundations for Internet architecture: clean slate approach, NSF NeTS FIND Initiative. http://www.nets-find.net/Funded/Algorithmic.php.

[13] (Online) AWISSENET: Ad-hoc personal area network and WIreless Sensor SEcure NETwork, European Union 7th Framework Program, http://www.awissenet.eu.

[14] H. Ballani, P. Francis, Fault Management Using the CONMan Abstraction, IEEE INFOCOM 2009, Rio de Janeiro, Brazil, April 2009.

[15] H. Ballani, P. Francis, CONMan: A Step Towar Network Manageability, ACM SIGCOMM 2007, Kyoto, Japan, August 2007.

[16] E. Bangeman, P2P responsible for as much as 90 percent of all 'Net traffic', ars Technical, September 3rd, 2007. <http://arstechnica.com/old/content/2007/09/p2p-responsible-for-as-much-as-90-percent-of-all-net-traffic.ars>.

[17] A. Bavier, N. Feamster, M. Huang, et al., In VINI veritas: realistic and controlled network experimentation, in: Proceedings of the ACM SIGCOMM 2006, Pisa, Italy, September 11–15, 2006, pp. 3–14.

[18] S.M. Bellovin, D.D. Clark, A. Perrig, et al., GDD-05-05: Report of NSF Workshop on A Clean-Slate Design for the Next-Generation Secure Internet, GENI Design Document 05-05, July 2005. <http://groups.geni.net/geni/attachment/wiki/OldGPGDesignDocuments/GDD-05-05.pdf>.

[19] T. Benzel, R. Braden, D. Kim, et al., Experience with DETER: A Testbed for Security Research, in: Proceedings of Tridentcom, International Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities, Barcelona, Spain, March 1–3, 2006.

[20] Y. Rekhter, T. Li, S. Hares, (Eds.), A Border Gateway Protocol 4 (BGP-4), RFC 4271, January 2006.

[21] Bobby Bhattacharjee, Ken Calvert, Jim Griffioen, Neil Spring, James Sterbenz, Postmodern Internetwork Architecture, NSF-FIND proposal, ITTC Technical Report ITTC-FY2006-TR-45030-01, The University of Kansas, February 2006.

[22] D.J. Blumenthal, J.E. Bowers, C. Partridge, GDD-05-03: Report of NSF Workshop on Mapping a Future for Optical Networking and Communications, GENI Design Document 05-03, July 2005. <http://groups.geni.net/geni/attachment/wiki/OldGPGDesignDocuments/GDD-05-03.pdf>.

[23] D. Boneh, D. Mazieres, M. Rosenblum, et al., Designing Secure Networks from the Ground-Up, NSF NeTS FIND Initiative, <http://www.nets-find.net/Funded/DesigningSecure.php>.

[24] M. Buchanan, 10 Percent of Broadband Subscribers Suck up 80 Percent of Bandwidth But P2P No Longer To Blame, Gizmodo, 22 April. <http://gizmodo.com/382691/10-percent-of-broadband-subscriber-suck-up-80-percent-of-bandwidth-but-p2p-no-longer-to-blame>.

[25] S. Burleigh, M. Ramadas, S. Farrell, et al., Licklider Transmission Protocol – Motivation, IETF RFC 5325, September 2008.

[26] M. Caesar, D. Caldwell, N. Feamster, et al., Design and implementation of a routing control platform, in: Proceedings of the 2nd Conference on Symposium on Networked Systems Design and Implementation (NSDI 2005), Berkeley, CA, May 02–04, vol. 2, 2005, pp. 15–28.

[27] Z. Cai, F. Dinu, J. Zheng, A.L. Cox, T.S. Eugene Ng, The Preliminary Design and Implementation of the Maestro Network Control Platform, Rice University Technical Report TR08-13, October 2008.

[28] R. Canonico, S. D'Antonio, M. Barone, et al., European ONELAB project: Deliverable D4B.1 – UMTS Node, September 2007. <http://www.onelab.eu/images/PDFs/Deliverables/d4b.1.pdf>.

[29] R. Canonico, A. Botta, G. Di Stasi, et al., European ONELAB project: Deliverable D4B.2 – UMTS Gateway, February 2008, <http://www.onelab.eu/images/PDFs/Deliverables/d4b.2.pdf>.

[30] A. Carzaniga, M.J. Rutherford, A.L. Wolf, A routing scheme for content-based networking, IEEE INFOCOM 2004, Hong Kong, China, March 2004.

[31] A. Carzaniga, A.L. Wolf, Forwarding in a content-based network, ACM SIGCOMM 2003, Karlsruhe, Germany, August 2003, pp. 163–174.

[32] A. Carzaniga, M.J. Rutherford, A.L. Wolf, A routing scheme for content-based networking, Technical Report CU-CS-953-03, Department of Computer Science, University of Colorado, June 2003.

[33] A. Carzaniga, A.L. Wolf, Content-based networking: a new communication infrastructure, in: NSF Workshop on an Infrastructure for Mobile and Wireless Systems. In Conjunction with the International Conference on Computer Communications and Networks ICCCN, Scottsdale, AZ, October 2001.

[34] A. Carzaniga, A.L. Wolf, Fast forwarding for content-based networking, Technical Report CU-CS-922-01, Department of Computer Science, University of Colorado, November, 2001 (Revised, September 2002).

[35] M. Carbone, L. Rizzo, European ONELAB project: Deliverable D4E.3 – Emulation Component, February 2008. <http://www.onelab.eu/images/PDFs/Deliverables/d4e.3.pdf>.

[36] V. Cerf, S. Burleigh, A. Hooke, et al., "Delay-Tolerant Network Architecture," IETF RFC 4838, April 2007.

[37] J. Chase, L. Grit, D. Irwin, et al., Beyond virtual data centers: toward an open resource control architecture, in: Proceedings of the International Conference on the Virtual Computing Initiative (ICVCI 2007), Research Triangle Park, North Carolina, May 2007.

[38] K. Claffy, M. Crovella, T. Friedman, et al., GDD-06-40: Community-Oriented Network Measurement Infrastructure (CONMI) Workship Report, GENI Design Document 06-40, December 2005. <http://groups.geni.net/geni/attachment/wiki/OldGPGDesignDocuments/GDD-06-40.pdf>.

[39] (Online) CORDIS website, European Union 7th Framework Program. <http://cordis.europa.eu/fp7/ict/programme/challenge1_en.html>.

[40] John Day, Patterns in Network Architecture: A Return to Fundamentals, Edition I, Prentice Hall, ISBN-10: 0-13-225242-2, December 2007, pp. 464.

[41] B. Donnet, L. Iannone, O. Bonaventure, European ONELAB project: Deliverable D4A.1 – WiMAX component, August 2008. <http://www.onelab.eu/images/PDFs/Deliverables/onelab14a1.pdf>.

[42] D. Dudkowski, M. Brunner, G.Nunzi, et al., Architectural principles and elements of in-network management, in: Mini-conference at IFIP/IEEE Integrated Management symposium, New York, USA, 2009.

[43] (Book) Jeff Dyke, User Mode Linux, Prentice Hall, April 2006.

[44] (Online) European Network of Excellence in Cryptology II, European Union 7th Framework Program. <http://www.ecrypt.eu.org>.

[45] T.S. Eugene Ng, A.L. Cox, Maestro: an architecture for network control management, NSF NeTS-FIND Initiative. <http://www.nets-find.net/Funded/Maestro.php>.

[46] K. Fall, A delay-tolerant network architecture for challenged Internets, in: Proceedings of SIGCOMM 2003, Karlsruhe, Germany, August 25–29, 2003, pp. 27–34.

[47] K. Fall, S. Farrell, DTN: an architectural retrospective, IEEE Journal on Select Areas in Communications 26 (5) (2008) 828–836.

[48] S. Farrell, M. Ramadas, S. Burleigh, Licklider Transmission Protocol – Security Extensions, IETF RFC 5327, September 2008.

[49] T. Faber, J. Wroclawski, K. Lahey, A DETER Federation Architecture, in: Proceedings of the DETER Community Workshop on Cyber Security Experimentation and Test, August 2007.

[50] N. Feamster, H. Balakrishnan, J. Rexford, et al., The case for separating routing from routers, in: ACM SIGCOMM Workshop on Future Directions in Network Architecture (FDNA), Portland, September, 2004, pp. 5–12.

[51] N. Feamster, L. Gao, J. Rexford, CABO: Concurrent Architectures are Better Than One, NSF NeTS FIND Initiative. <http://www.nets-find.net/Funded/Cabo.php>.

[52] P. Szegedi, Deliverable JRA2.1: Architectures for virtual infrastructures,new Internet paradigms and business models, Version 1.6, FEDERICA project, European Union 7th Framework, October 2008.

[53] Deliverable DSA1.1: FEDERICA Infrastructure, Version 7.0, FEDERICA project, European Union 7th framework, October 2008.

[54] C. Foley, S. Balasubramaniam, E. Power, et al., A framework for in-network management in heterogeneous future communication networks, in: Proceedings of the MACE 2008, Samos Island, Greece, September 22–26, vol. 5276, 2008, pp. 14–25.

[55] P. Francis, J. Lepreau, Towards Complexity-Oblivious Network Management, NSF NeTS-FIND Initiative. <http://www.nets-find.net/Funded/TowardsComplexity.php>.

[56] Y. Fu, J. Chase, B. Chun, et al., SHARP: an architecture for secure resource peering, SIGOPS Operation System Review 37 (5) (2003) 133–148.

[57] GENI-SE-SY-RQ-01.9: GENI Systems Requirements, Prepared by GENI Project Office, BBN Technologies, January 16, 2009. <http://groups.geni.net/geni/attachment/wiki/SysReqDoc/GENI-SE-SY-RQ-02.0.pdf>.

[58] GENI-SE-CF-RQ-01.3: GENI Control Framework Requirements, Prepared by GENI Project Office, BBN Technologies, January 9, 2009. The 4D Project: Clean Slate Architectures for <http://groups.geni.net/geni/attachment/wiki/GeniControlFrameworkRequirements/010909b>.

[59] GENI-FAC-PRO-S1-OV-1.12: GENI Spiral 1 Overview, Prepared by GENI Project Office, BBN Technologies, September 2008. <http://groups.geni.net/geni/attachment/wiki/SpiralOne/GENIS1Ovrvw092908.pdf>.

[60] P. Brighten Godfrey, Igor Ganichev, Scott Shenker, Ion Stoica, Pathlet Routing, in: Proc. ACM SIGCOMM, Barcelona, Spain, August 2009.

[61] A.G. Prieto, D. Dudkowski, C. Meirosu, et al., Decentralized in-network management for the future internet, in: Proceedings of IEEE ICC'09 International Workshop on the Network of the Future, Dresden, Germany, 2009.

[62] T. Griffin, F.B. Shepherd, G. Wilfong, The stable paths problem and interdomain routing, IEEE/ACM Transaction on Networking 10(1), 232–243.

[63] T.G. Griffin, G. Wilfong, On the correctness of IBGP configuration, ACM SIGCOMM 2002, Pittsburgh, PA, August 19–23, 2002.

[64] A. Greenberg, G. Hjalmtysson, D.A. Maltz, et al., A clean slate 4D approach to network control and management, ACM SIGCOMM Computer Communication Review 35 (5) (2005).

[65] A. Greenberg, G. Hjalmtysson, D.A. Maltz, et al., Refactoring network control and management: a case for the 4D architecture, CMU CS Technical Report CMU-CS-05-117, September 2005.

[66] J. Guare, Six Degrees of Separation: A Play, Vintage Books, New York, 1990, 120 p.

[67] (Online) Hiroaki Harai, AKARI Architecture Design Project in Japan, August 2008. <http://akari-project.nict.go.jp/eng/document/asiafi-seminar-harai-080826.pdf>.

[68] R. Moskowitz, P. Nikander, P. Jokela, Host Identity Protocol (HIP) Architecture, IETF RFC4423, May 2006.

[69] M. Ho, K. Fall, Poster: Delay Tolerant Networking for Sensor Networks, in: Proceedings of the First IEEE Conference on Sensor and Ad Hoc Communications and Networks (SECON 2004), October 2004.

[70] (Online) INTERSECTION: INfrastructure for heTErogeneous, Resilient, SEcure, Complex, Tightly Inter-Operating Networks, European Union 7th Framework Program. <http://www.intersection-project.eu/>.

[71] IPv6, <http://www.ipv6.org/>.

[72] H. Iqbal, T. Znati, Distributed control plane for 4D architecture, Globecom, 2007.

[73] D. Irwin, J. Chase, L. Grit, et al., Sharing networked resources with brokered leases, in: Proceedings of USENIX Technical Conference, Boston, Massachusetts, June 2006.

[74] Computer Business Review Online, ITU head foresees internet balkanization, November 2005.

[75] V. Jacobson, Content Centric Networking, Presentation at DARPA Assurable Global Networking, January 30, 2007.

[76] Van Jacobson, Diana K. Smetters, James D. Thornton, Michael F. Plass, Nicholas H. Briggs, Rebecca L. Braynard, Networking Named Content, CoNext, Rome, Italy, 2009.

[77] S. Jain, K. Fall, R. Patra, Routing in Delay Tolerant Network, in: Proceedings of SIGCOMM 2004, Oregon, USA, August 2004.

[78] R. Jain, Internet 3.0: Ten Problems with Current Internet Architecture and Solutions for the Next Generation, in: Proceedings of Military Communications Conference (MILCOM 2006), Washington, DC, October 23–25, 2006.

[79] T. Koponen, M. Chawla, B. chun, et al., A data-oriented (and beyond) network architecture, ACM SIGCOMM Computer Communication Review 37 (4) (2007) 181–192.

[80] F. Kaashoek, B. Liskov, D. Andersen, et al., GDD-05-06: Report of the NSF Workshop on Research Challenges in Distributed Computer Systems, GENI Design Document 05-06, December 2005. <http://groups.geni.net/geni/attachment/wiki/OldGPGDesignDocuments/GDD-05-06.pdf>.

[81] C. Kim, M. Caesar, J. Rexford, Floodless in seattle: a scalable ethernet architecture for large enterprises, in: Proceedings of the ACM SIGCOMM 2008 Conference on Data Communication (Seattle, WA, USA), August 17–22, 2008.

[83] D. Krioukov, K. Claffy, K. Fall, Greedy Routing on Hidden Metric Spaces as a Foundation of Scalable Routing Architectures without Topology Updates, NSF NeTS FIND Initiative. <http://www.nets-find.net/Funded/Greedy.php>.

[84] C. Labovitz, A. Ahuja, A. Bose, F. Jahanian, Delayed Internet routing convergence, IEEE/ACM Transaction on Networking 9 (3) (2001) 293–306.

[85] C. Labovitz, A. Ahuja, F. Jahanian, Experimental study of Internet stability and wide-area network failures, in: Proceedings of the International Symposium on Fault-Tolerant Computing, 1999.

[86] C. Labovitz, R. Malan, F. Jahanian, Origins of Internet routing instability, in: Proceedings of IEEE INFOCOM, New York, NY, March 1999.

[87] K. Lahey, R. Braden, K. Sklower, Experiment Isolation in a Secure Cluster Testbed, in: Proceedings of the CyberSecurity Experimentation and Test (CSET) Workshop, July 2008.

[88] T. Leighton, Improving performance in the Internet, ACM Queue 6 (6) (2008) 20–29.

[89] D. Farinacci, V. Fuller, et al., Internet Draft: Locater/ID Separation Protocol (LISP), draft-farinacci-LISP-03, August 13, 2007.

[90] T. Li, Internet Draft: Design Goals for Scalable Internet Routing, IRTF draft-irtf-rrg-design-goals-01 (work in progress), July 2007.

[91] T. Li, Internet Draft: Preliminary Recommendation for a Routing Architecture, IRTF draft-irtf-rrg-recommendation-00, February 2009.

[92] H. Luo, R. Kravets, T. Abdelzaher, The-Day-After Networks: A First-Response Edge-Network Architecture for Disaster Relief, NSF NeTS FIND Initiative. <http://www.nets-find.net/Funded/DayAfterNet.php>.

[93] Z.M. Mao, R. Govindan, G. Varghese, R. Katz, Route flap damping exacerbates Internet routing convergence, in: Proceedings of ACM SIGCOMM, Pittsburgh, PA, August 19–23, 2002.

[94] D. Massey, L. Wang, B. Zhang, L. Zhang, Enabling Future Internet innovations through Transitwire (eFIT), NSF NeTS FIND Initiative <http://www.nets-find.net/Funded/eFIT.php>.

[95] MASTER: Managing Assurance, Security and Trust for sERvices, European Union 7th Framework Program. <http://www.master-fp7.eu>.

[96] M. Caesar, D. Caldwell, N. Feamster, et al., Design and implementation of a routing control platform, in: Second Symposium on Networked Systems Design and Implementation (NSDI'05), April 2005.

[97] B. Mathieu, D. Meddour, F. Jan, et al., European ONELAB project: Deliverable D4D1 – OneLab wireless mesh multi-hop network, August 2007. <http://www.onelab.eu/images/PDFs/Deliverables/d4d.1.pdf>.

[98] N. McKeown, T. Anderson, H. Balakrishnan, et al., OpenFlow: Enabling Innovation in Campus Networks, OpenFlow Whitepaper, March 2008. <http://www.openflowswitch.org/documents/openflow-wp-latest.pdf>.

[99] D. Menasche,A. Rocha, B. Li, D. Towsley, A. Venkataramani, Content availability and bundling in peer-to-peer swarming systems, in: ACM Sigcomm International Conference on Emerging Networking Experiments and Technologies (CoNEXT), December 2009.

[100] D. Meyer, L. Zhang, K. Fall, Report from IAB workshop on routing and addressing, IETF RFC 4984, September 2007.

[101] S. Milgram, The small world problem, Psychology Today 1 (1967) 61–67.

[102] (Online) MOBIO: Mobile Biometry, Secured and Trusted Access to Mobile Services, European Union 7th Framework Program. <http://www.mobioproject.org>.

[103] G. Neglia, G. Reina, H. Zhang, D. Towsley, A. Venkataramani, J. Danaher, Availability in BitTorrent systems, Infocom, 2007.

[104] (Online) Networked European Software and Services Initiative, a European Technology Platform on Software Architectures and Services Infrastructures. <http://www.nessi-europe.eu/Nessi/>.

[105] H. Schulzrinne, S. Seetharaman, V. Hilt, NetSerV – Architecture of a Service-Virtualized Internet, NSF NeTS FIND Initiative, <http://www.nets-find.net/Funded/Netserv.php>.

[106] P. Nikander et al., Host Identity Indirection Infrastructure (Hi3), in: Proceedings of The Second Swedish National Computer Networking Workshop 2004 (SNCNW2004), Karlstad University, Karlstad, Sweden, November 23–24, 2004.

[107] A. de la Oliva, B. Donnet, I. Soto, et al., European ONELAB project: Deliverable D4C.1 – Multihoming Architecture Document, February 2007. <http://www.onelab.eu/images/PDFs/Deliverables/d4c.1.pdf>.

[108] A. de la Oliva, B. Donnet, I. Soto, European ONELAB project: Deliverable D4C.2 – Multihoming Mechanisms Document, August 2007. <http://www.onelab.eu/images/PDFs/Deliverables/d4c.2.pdf>.

[109] (Online) University of Utah, the Emulab Project, 2002. <http://www.emulab.net>.

[110] (Online) University of Wisconsin, The Wisconsin Advanced Internet Laboratory, 2007. <http://wail.cs.wisc.edu>.

[111] (Online) PlanetLab. <http://www.planet-lab.org>.

[112] (Online) OneLab. <http://www.onelab.eu>.

[113] (Online) VINI. <http://www.vini-veritas.net/?q=node/34>.

[114] (Online) User Mode Linux. <http://user-mode-linux.sourceforge.net>.

[115] (Online) XORP: eXtensible Open Router Platform. <http://www.xorp.org>.

[116] (Online) AKARI Project. <http://akari-project.nict.go.jp/eng/index2.htm>.

[117] (Online) OpenFlow Project. <http://www.openflowswitch.org/>.

[118] (Online) GENI: Global Environment for Network Innovations. <http://www.geni.net>.

[119] (Online) FIRE: Future Internet Research and Experimentation. <http://cordis.europa.eu/fp7/ict/fire/>.

[120] (Online) National LambdaRail. <http://www.nlr.net/>.

[121] (Online) Internet 2. <http://www.Internet2.edu/>.

[122] (Online) GENI Spiral 1. <http://groups.geni.net/geni/wiki/SpiralOne>.

[123] (Online) DETERlab Testbed. <http://www.isi.edu/deter>.

[124] (Online) TIED: Trial Integration Environment in DETER. <http://groups.geni.net/geni/wiki/TIED>.

[125] (Online) DRAGON: Dynamic Resource Allocation via GMPLS Optical Networks. <http://dragon.maxgigapop.net/twiki/bin/view/DRAGON/WebHome>.

[126] (Online) ProtoGENI. <http://groups.geni.net/geni/wiki/ProtoGENI>.

[127] (Online) ProtoGENI ClearingHouse. <http://www.protogeni.net/trac/protogeni/wiki/ClearingHouseDesc>.

[128] (Online) ORCA. <http://groups.geni.net/geni/wiki/ORCABEN>.

[129] (Online) BEN: Breakable Experimental Network. <https://geni-orca.renci.org/trac>.

[130] (Online) ORBIT. <http://groups.geni.net/geni/wiki/ORBIT>.

[131] (Online) ORBIT-Lab. <http://www.orbit-lab.org>.

[132] (Online) OMF: Control and Management Framework. <http://omf.mytestbed.net>.

[133] (Online) Milestone ORBIT: 1a Extend OMF to support multiple heterogeneous testbeds. <http://groups.geni.net/geni/milestone/ORBIT%3A%201a%20Extend%20OMF%20to%20support%20multiple% 20heterogeneous%20testbeds>.

[134] (Online) GEANT. <http://www.geant.net>.

[135] (Online) FP6 Research Networking Testbeds. <http://cordis.europa.eu/fp7/ict/fire/fp6-testbeds_en.html>.

[136] (Online) Panlab. <http://www.panlab.net>.

[137] (Online) Vital++. <http://www.ict-vitalpp.upatras.gr>.

[138] (Online) WISEBED: Wireless Sensor Network Testbeds. <http://www.wisebed.eu>.

[139] (Online) FEDERICA. <http://www.fp7-federica.eu>.

[140] (Online) GEANT2. <http://www.geant2.net>.

[142] (Online) CacheLogic, Home Page: Advanced Solutions for P2P Networks Home Page. <http://www.cachelogic.com>.

[143] (Online) AKAMAI, AKAMAI to enable Web for DVD and HD video, August 31, 2007. <http://www.akamai.com/dl/akamai/Akam_in_Online_Reporter.pdf>.

[144] (Online) Napster Home Web Page, <http://www.napster.com>.

[145] (Online) Hyperconnectivity and the Approaching Zettabyte Era, White Paper, Cisco Visual Networking Index(VNI), Jun 02, 2010. <http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/VNI_Hyperconnectivity_WP.html>.

[146] (Online) BitTorrent. <www.bittorrent.com>.

[147] (Online) P4P working group. <http://www.openp4p.net>.

[148] (Online) P2PNext Project. <http://www.p2p-next.org>.

[149] (Online) KaZaa, KaZaa, <http://www.kazaa.com>.

[150] (Online) Gnutella. <http://en.wikipedia.org/wiki/Gnutella>.

[151] (Online) InterPlaNetary Internet Project, Internet Society IPN Special Interest Group. <http://www.ipnsig.org/home.htm>.

[152] (Online) Delay Tolerant Networking Research group, IRTF. <http://irtf.org/charter?gtype=rg&group=dtnrg>.

[153] (Online) CENS: Center for Embedded Networked Sensing. <http://research.cens.ucla.edu>.

[154] (Online) SeNDT: Sensor Networking with Delay Tolerance. <http://down.dsg.cs.tcd.ie/sendt>.

[155] (Online) DTN/SN Project, Swedish Institute of Computer Science. <http://www.sics.se/nes/Projects/DTNSN.html>.

[156] (Online) The 4D Project: Clean Slate Architectures for Network Management. URL: <http://www.cs.cmu.edu/∼4D/>.

[157] (Online) IBM Corporation, Autonomic computing – a manifesto, 2001. <www.research.ibm.com/autonomic>.

[158] (Online) Autonomic Network Architecture (ANA) Project. <http://www.ana-project.org>.

[159] OneLab2 Whitepaper: "On Federations…, January 2009. <http://www.onelab.eu/index.php/results/whitepapers/294-whitepaper-1-on-federations.html>.

[160] P. Antoniadis, T. Friedman, X. Cuvellier, Resource Provision and Allocation in Shared Network Testbed Infrastructures, ROADS 2007, Warsaw, Poland, July 11–12, 2007.

[161] J. Pan, S. Paul, R. Jain, et al., MILSA: a mobility and multihoming supporting identifier locater split architecture for next generation Internet, in: Proceedings of IEEE GLOBECOM 2008, New Orleans, LA, December 2008. <http://www.cse.wustl.edu/jain/papers/milsa.htm>.

[162] J. Pan, S. Paul, R. Jain, et al., Enhanced MILSA Architecture for Naming, Addressing, Routing and Security Issues in the Next Generation Internet, in: Proceedings of IEEE ICC 2009, Dresden, Germany, June 2009. <http://www.cse.wustl.edu/jain/papers/emilsa.htm>.

[163] J. Pan, S. Paul, R. Jain, et al., Hybrid Transition Mechanism for MILSA Architecture for the Next Generation Internet, in: Proceedings of the Second IEEE Workshop on the Network of the Future (FutureNet II), IEEE Globecom 2009, Honolulu, Hawaii, 30 November–4 December, 2009.

[164] J. Pan, R. Jain, S. Paul, et al., MILSA: A new evolutionary architecture for scalability, mobility, and multihoming in the future Internet, Journal of IEEE on Selected Area in Communications (JSAC) 28 (8) (2010).

[165] S. Paul, S. Seshan, GDD-06-17: Technical Document on Wireless Virtualization, GENI Design Document 06-17, September 2006. <http://groups.geni.net/geni/attachment/wiki/OldGPGDesignDocuments/GDD-06-17.pdf>.

[166] S. Paul, R. Jain, J. Pan, et al., A vision of the next generation internet: a policy oriented view, in: Proceedings of British Computer Society conference on Visions of Computer Science, September 2008, pp. 1–14.

[167] S. Paul, R. Jain, J. Pan, Chakchai So-in, Multi-Tier Diversified Architecture for Internet 3.0: the Next Generation Internet, WUSTL Technical Report, WUCSE-2010-XX, June 2010, 25 pp.

[168] L. Peterson, A. Bavier, M.E. Fiuczynski, et al., Experiences building planetlab, in: Proceedings of the 7th symposium on Operating Systems Design and Implementation (OSDI 2006), Berkeley, CA, 2006, pp. 351–366.

[169] L. Peterson, S. Sevinc, J. Lepreau, et al., Slice-Based Facility Architecture, Draft Version 1.04, April 7, 2009. <http://svn.planet-lab.org/attachment/wiki/GeniWrapper/sfa.pdf>.

[170] L. Peterson, S. Sevinc, S. Baker, et al., PlanetLab Implementation of the Slice-Based Facility Architecture, Draft Version 0.05, June 23, 2009. <http://www.cs.princeton.edu/geniwrapper.pdf>.

[171] (Presentation) Panayotis Antoniadis et al., The Onlab2 Project and research on federations, Kassel, March 2009. <http://www.onelab.eu/images/PDFs/Presentations/onelab_pa_kivs09.pdf>.

[172] R. Ramanathan, R. Hansen, P. Basu, et al., Prioritized Epidemic Routing for Opportunistic Networks, in: Proceedings of ACM MobiSys Workshop on Mobile Opportunistic Networking (MobiOpp 2007), San Juan, Puerto Rico, USA, June 11, 2007.

[173] A. Ramachandran, A. das Sarma, N. Feamster, Bitstore: an incentive compatible solution for blocked downloads in Bittorrent, NetEcon, 2007.

[174] M. Ramadas, S. Burleigh, S. Farrell, Licklider Transmission Protocol – Specification, IETF RFC 5326, September 2008.

[175] D. Raychaudhuri, M. Gerla, GDD-05-04: Report of NSF Workshop on New Architectures and Disruptive Technologies for the Future Internet: The Wireless, Mobile and Sensor Network Perspective, GENI Design Document 05-04, August 2005. <http://groups.geni.net/geni/attachment/wiki/OldGPGDesignDocuments/GDD-05-04.pdf>.

[176] Y. Rekhter, T. Li, S. Hares, A Border Gateway Protocol 4 (BGP-4), IETF RFC 4271, January 2006.

[177] J. Rexford, J. Wang, Z. Xiao, et al., BGP routing stability of popular destinations, in: Proceedings of the 2nd ACM SIGCOMM Workshop on Internet Measurement, Marseille, France, November 6–8, 2002.

[178] J. Rexford, A. Greenberg, G. Hjalmtysson, et al., Network-Wide Decision Making: Toward A Wafer-Thin Control Plane, in: Proceedings of HotNets III, November, 2004.

[179] M. Robuck, Survey: P2P sucking up 44% of bandwidth, CED Magazine, 25 June 2008. <http://www.cedmagazine.com/P2P-44-percent-bandwidth.aspx>.

[180] J. Sanjuas, G. Iannaccone, L. Peluso, et al., European ONELAB project: Deliverable D3A.2 Prototype Passive Monitoring Component, January 2008. <http://www.onelab.eu/index.php/results/deliverables/252-d3a2-passive-monitoring-component.html>.

[181] (Online) Internet Research Task Force Routing Research Group Wiki page, 2008. <http://trac.tools.ietf.org/group/irtf/trac/wiki/RoutingResearchGroup>.

[182] S. Schwab, B. Wilson, C. Ko, et al., SEER: A Security Experimentation EnviRonment for DETER, in: Proceedings of the DETER Community Workshop on Cyber Security Experimentation and Test, August 2007.

[183] K. Scott, S. Burleigh, Bundle Protocol Specification, IETF RFC 5050, November 2007.

[184] T. Wolf, Service-Centric End-to-End Abstractions for Network Architecture, NSF NeTS FIND Initiative. <http://www.nets-find.net/Funded/ServiceCentric.php>.

[185] S. Seshan, D. Wetherall, T. Kohno, Protecting User Privacy in a Network with Ubiquitous Computing Devices, NSF NeTS FIND Initiative. <http://www.nets-find.net/Funded/Protecting.php>.

[186] L. Sha, A. Agrawala, T. Abdelzaher, et al. GDD-06-32: Report of NSF Workshop on Distributed Real-time and Embedded Systems Research in the Context of GENI, GENI Design Document 06-32, September 2006. <http://groups.geni.net/geni/attachment/wiki/OldGPGDesignDocuments/GDD-06-32.pdf>.

[187] N. Shenoy, Victor Perotti, Switched Internet Architecture, NSF NeTS FIND Initiative. <http://www.nets-find.net/Funded/SWA.php>.

[188] E. Nordmark, M. Bagnulo, Internet Draft: Shim6: level 3 multihoming Shim protocol for IPv6, IETF RFC 5533, June 2009.

[189] SHIELDS: Detecting known security vulnerabilities from within design and development tools, European Union 7th Framework Program. <http://www.shieldsproject.eu>.

[190] G. Rouskas, R. Dutta, I. Baldine, et al., The SILO Architecture for Services Integration, Control, and Optimization for the Future Internet, NSF NeTS-FIND Initiative. <http://www.nets-find.net/Funded/Silo.php>.

[191] Empowering the Service Economy with SLA-aware Infrastructures, European Union 7th Framework Program. <http://sla-at-soi.eu>.

[192] A.C. Snoeren, Y. Kohno, S. Savage, et al., Enabling Defense and Deterrence through Private Attribution, NSF NeTS-FIND Initiative. <http://www.nets-find.net/Funded/EnablingDefense.php>.

[193] Service Oriented Architectures for ALL, European Union 7th Framework Program. <http://www.soa4all.eu>.

[194] I. Stoica, D. Adkins, et al., Internet Indirection Infrastructure, in: Proceedings of ACM SIGCOMM 2002, Pittsburgh, Pennsylvania, USA, 2002.

[195] L. Subramanian, M. Caesar, C.T. Ee, et al., HLP: a next generation inter-domain routing protocol, in: Proceedings of SIGCOMM 2005, Philadelphia, Pennsylvania, August 22–26, 2005.

[196] SWIFT: Secure Widespread Identities for Federated Telecommunications, European Union 7th Framework Program. <http://www.ist-swift.org>.

[197] TAS3: Trusted Architecture for Securely Shared Services, European Union 7th Framework Program. <http://www.tas3.eu>.

[198] TECOM: Trusted Embedded Computing, Information Technology for European Advanced (ITEA2) Programme. <http://www.tecom-itea.org>.

[199] C. Thompson, The BitTorrent Effect, WIRED, Issue 13.01, January 2005.

[200] J. Touch, Y. Wang, V.Pingali, A Recursive Network Architecture, ISI Technical Report ISI-TR-2006-626, October 2006.

[201] J.Touch, V.K. Pingali, The RNA Metaprotocol, Proc. IEEE ICCCN (Future Internet Architectures and Protocols track), St. Thomas, Virgin Islands, August 2008.

[202] J. Turner, GDD-06-09: A Proposed Architecture for the GENI Backbone Platform, Washington University Technical Report WUCSE-2006-14, March 2006. <http://groups.geni.net/geni/attachment/wiki/OldGPGDesignDocuments/GDD-06-09.pdf>.

[203] J. Turner, P. Crowley, J. DeHart, et al., Supercharging PlanetLab – a High Performance, Multi-Application, Overlay Network Platform Multi-Application, Overlay Network Platform, in: Proceedings of ACM SIGCOMM, Kyoto, Japan, August 2007.

[204] J. Turner, P. Crowley, S. Gorinsky, et al., An Architecture for a Diversified Internet, NSF NeTS FIND Initiative. <http://www.nets-find.net/Funded/DiversifiedInternet.php>.

[205] M.Y.S. Uddin, H. Ahmadi, T. Abdelzaher, R.H. Kravets, A low-energy, multi-copy inter-contact routing protocol for disaster response networks, Sensor, Mesh and Ad Hoc Communications and Networks, in: 6th Annual IEEE Communications Society Conference on SECON'09, June 2009, pp. 1–9.

[206] K. Varadhan, R. Govindan, D. Estrin, Persistent route oscillations in inter-domain routing, Computer Networks 32 (1) (2000) 1–16.

[207] A. Venkataramani, D. Towsley, A Swarming Architecture for Internet data transfer, NSF NeTS-FIND Initiative. <http://www.nets-find.net/Funded/Swarming.php>.

[208] VMWare. <http://www.vmware.com/.

[209] VServer. <http://linux-vserver.org/Welcome_to_Linux-VServer.org>.

[211] Y. wang, H. Wu, F. Lin, et al., Cross-layer protocol design and optimization for delay/fault-tolerant mobile sensor networks (DFT-MSN's), IEEE Journal on Selected Areas in Communications 26 (5) (2008) 809–819.

[212] J.W. Han, F.D. Jahanian, Topology aware overlay networks, in: Proceeding of IEEE INFOCOM, vol. 4, March 13–17, 2005, pp. 2554–2565.

[213] WISEBED: Grant Agreement, Deliverable D1.1, 2.1 and 3.1: Design of the Hardware Infrastructure, Architecture of the Software Infrastructure and Design of Library of Algorithms, Seventh Framework Programme Theme 3, November 30, 2008. <http://www.wisebed.eu/images/stories/deliverables/d1.1-d3.1.pdf>.

[214] L. Wood, W. Eddy, P. Holliday, A bundle of problems, in: IEEE Aerospace Conference, Big Sky, Montana, 2009.

[215] Xen. <http://www.xen.org/>.

[216] H. Xie, Y.R. Yang, A. Krishnamurthy, Y.G. Liu, A. Silberschatz, P4p: provider portal for applications, SIGCOMM Computer Communication Review 38 (4) (2008) 351–362.

[217] H. Xie, Y.R. Yang, A. Krishnamurthy,Y.G. Liu, A. Silberschatz, Towards an ISP-compliant, peer-friendly design for peer-to-peer networks, in: Proceedings of the 7th International Ifip-Tc6 Networking Conference on Adhoc and Sensor Networks, Wireless Networks, Next Generation internet, Singapore, May 05–09, 2008. Also Available at: A. Das, F.B. Lee, H.K. Pung, L.W. Wong, (Eds.), Lecture Notes In Computer Science, Springer-Verlag, Berlin, Heidelberg, pp. 375–384.

[219] H. Yan, D.A. Maltz, T.S. Eugene Ng, et al., Tesseract: A 4D Network Control Plane, in: Proceedings of USENIX Symposium on Networked Systems Design and Implementation (NSDI '07), April 2007.

[220] R. Yates, D. Raychaudhuri, S. Paul, et al., Postcards from the Edge: A Cache-and-Forward Architecture for the Future Internet, NSF NeTS FIND Initiative. <http://www.nets-find.net/Funded/Postcards.php>.

[221] X. Yang, An Internet Architecture for User-Controlled Routes, NSF NeTS FIND Initiative> <http://www.nets-find.net/Funded/InternetArchitecture.php>.

[222] Z. Zhang, Routing in intermittently connected mobile ad hoc networks and delay tolerant networks: overview and challenges, IEEE Communications Surveys and Tutorials 8 (1) (2006).

[223] J. Zien, The Technology Behind Napster, About, 2000. <http://Internet.about.com/library/weekly/2000/aa052800b.htm>.