Elaborato finale in Reti di Calcolatori

Research approaches and testbeds for future Internet architectures

Anno Accademico 2011/2012

Candidato:

Geremia Longobardo

matr. N46/000654
Index

Introduction 1

Chapter 1. Global Environment for Network Innovations 3

  1.1  Spiral development 5
  1.1.1  Spiral 1 6
  1.2  Control Frameworks 7
  1.2.1  Successive Spirals (2009-today) 8
  1.2.2  Control Frameworks today 8
  1.3  GENI's Resources 9
  1.3.1  GENI Racks 10
  1.4  Planetlab 11
  1.4.1  Onelab 12

Chapter 2. Openflow and Slice-based Federated Architecture 13

  2.1  Resource virtualization 15
  2.2  Slice-based Federated Architecture 16
  2.2.1  Components 16
  2.2.2  Deepening on slivers and slices 17

Chapter 3. Future Internet Architectures (FIA) 20

  3.1  eXpressive Internet Architecture (XIA) 20
  3.2  Mobility First 23
  3.3  Nebula 25
  3.4  Named Data Network 27

Conclusions 30
Bibliography 31
List of Figures 34
Introduction

Internet as we know it, is more than 30 years old and is used for many more purposes than it was originally designed for:

- VoIP
- video streaming
- social networks

Internet base protocols were not meant to handle all this new functions, furthermore its dimension has become bigger by several orders of magnitude and hundreds of new protocols and extensions have been added, which makes management more complex every day.

Some researchers assert that all these new solutions can be seen as "patches" which cannot hold forever.[3]

A "clean slate" approach to the problem is needed and Research Institutes must work together for a redesign and a long term paradigm change.[6]

According to others, instead, radical changes are a threat to Internet stability and growth, today's Internet is completely scalable.

There was already an improvement with the transition to IPv6 that not only extends the address range but brings some advantages in the management of the new mobile/wireless services.

But IPv6 may be insufficient in the case new challenges show up.
IP has a “narrow waist” which means that the core architecture is hard to modify and upgrade, new functions have to be implemented through inefficient patches on top of the existing architecture.[21]

This flaw is clearly illustrated by the 15+ years deployment history of IPv6 and the difficulty of integrate functions needed to enhance the security.

Two extremes dominated telecommunication world so far:

- PHONE NETWORK, smart network, "dumb" endpoints
- INTERNET, "dumb" network, smart endpoints

How can we have the best of both?

How to make a better use of optical networks and the spectrum?

ISP must be allowed to provide value added services and new user-centric innovations must migrate from endpoints to network.

The two main initiatives funded by US and European Governments that aim to study new solutions for Future Internet, are Global Environment for Network Innovations (GENI) and Future Internet Research and Experimentation Initiative (FIRE).

Their researchers do not ask themselves how a new technology fits within existing Internet architecture, but how it can shape Future Internet in realizing its potential.
Chapter 1

Global Environment for Network Innovations

GENI is a virtual and distributed laboratory funded by the National Science Foundation. Its main scope is to experiment innovations in networks field and particularly to redesign or improve today’s Internet.

There are about 50 American Universities and some industrial partners, that participate this project[22], grouped into 4 main Working Groups:

- **Experiment Workflow & Service WG**
  The focal point is the developing of tools and services needed by researchers doing experiments on GENI for each Spiral cycle.

- **Control Framework WG**
  Obtaining and managing shared resources, interacting with other testbeds federated with GENI, access control within GENI. Definition of a GENI aggregated API for policies, scheduling and resource representation.

- **Operations Management, Integration & Security**
  Tracking new instruments (both Open and commercial), that can be adapted for large scale operations and GENI.
  Definition of security policies for assets, data and processes; long term operations that fit GENI infrastructure and production of documentation about concept of operations in various fields.

- **Instrumentation and Measurement**
  Provide GENI with data-gathering, analysis and archival instruments, removing the burden to become a system and network measurement infrastructure expert on the GENI researcher, so that he can better focus on his own experiments.
What are Internet problems that GENI and FIRE, the European corresponding, are currently studying and want to remedy?

- Difficulties to understand and foresee the behavior of large scale networks
- Difficulties to experiment on current architectures and technologies
- Issues in understanding when Internet is really safe and ensures the privacy, it was also originally designed for research, therefore a small group of trusted users.

Old Internet was based on connection between machines, Future Internet should be based on users and content.

GENI's goals are to build models which describe big networks, to comprehend the behaviors and the causes of failures, develop new architectures easier to evolve and extend for future networks (a better support to mobility for example), develop safer system more suited to protect privacy.[5]

GENI is basically a distributed, heterogeneous, programmable and virtualized network that is used by network researchers to test their new ideas for the future of networking: a TESTBED.[1]

- Trial of brand new architectures which may or not be compatible with current infrastructures
- Long running realistic experiments to gather real data
- Large scale growth of experiment for further testing
- Tight integration of physical layer with higher layers

**GENI facility: motivation**

![Diagram showing the need for large experimental facility/infrastructure and the maturity of different research stages](Figure 1.1)
1.1 Spiral development

GENI was built following the principles of Spiral development instead of the classic Waterfall model, because it is more suitable for large and complicated projects, and allows at the end of each cycle to estimate the general quality, the fulfillment of the requirements and the schedule.

This method combines the advantages of both top-down and bottom-up approaches and allows developers to re-evaluate the final goal of each cycle, following the changes of technology and allowing earlier releases.[2]

During each spiral there are the classic Waterfall phases but shorter

- **Planning**: the identification of objectives, alternatives, and constraints on the new iteration
- **Design**, choosing the software structure and how to meet these requirements
- **Build out**, the development and prototyping begins
- **Integration**, all pieces start to fit together, the testing begins
- **Use**, the software at this point should be functional and running
Early prototyping demonstrates the core GENI concepts, exposes their strengths and limitations, identifies critical risks, and lets operators understand what additional technologies are needed to progress.

At the same time, working prototypes help fuel community discussions of GENI’s scope and construction.

**1.1.1 Spiral 1**

GENI has yearly cycles and is actually approaching the end of the fourth one.

The main goal was to develop and operate very rudimentary, end-to-end working prototypes as soon as possible, it makes the community acquire experience and confidence with this new platform [8].

The creation of semi-independent and competing groups is also an important feature of spiral developing.

Spiral 1 contains tens of entities grouped according to the control framework in common (DETER, Planetlab, ProtoGENI, ORCA, ORBIT) into **Clusters** and chosen through an open competitive solicitation.

Every Control Framework contains some but not all of the functionality envisioned for a GENI control plane.

Their main scope will be to evolve their individual designs, and the conceptual design of GENI as a whole.
Since one intended use of the GENI suite is experimentation with non-IP data transport, researchers needed to create mechanisms for creating virtual topologies that can transport arbitrary packet formats: two mechanisms were selected for trial use in Spiral 1: IP tunnels and Ethernet VLAN.

1.2 Control Frameworks

Control frameworks are interfaces and tools available to the experimenter to create and manage GENI slices which are, in simple terms, virtual topologies that link together resources within different aggregates.

There are many different ways to create virtual topologies with modern networking technology, including IP Virtual Private Networks, Ethernet VLANs, Multi Protocol Label Switching, etc.

By accessing GENI through a control framework, the experimenter is provided with an API and/or GUI that makes easier his job, facilitating identification, reservation, configuration, and interconnection of various GENI resources granted to him.

Other functions include the provision an operational clearinghouse, Operations and Management and security functions and programmable interfaces for GENI tools which help researchers to find and use GENI resources.

Ideally, some software interfaces will be common across multiple control frameworks, laying the foundations for later GENI-wide integration.

The five control frameworks have different emphasis:

**CLUSTER A: TIED/DETER (Cyber Defense Technology Experimental Research)**

This control framework is focused on trust, federation and security, it aims to provide usability across multiple platforms, on-demand federation with GENI and non-GENI testbeds.

**CLUSTER B: PLANETLAB** (see par.1.4)

**CLUSTER C: ProtoGENI**, based on Emulab and focused on network control and management.

**CLUSTER D: ORCA (Open Resource Control Architecture)**, focused on sensor networks.

**CLUSTER E: ORBIT (Open-Access Research Testbed for Next-Generation Wireless Networks)** focused on wireless networks (especially WiMAX) and their integration within GENI.
1.2.1 Successive Spirals (2009-today)

Each cycle's main objectives are interoperability between control frameworks, meso-scale experimentation, improved integration, better instruments for measurements and security, the development and update of an aggregate manager (AM) interface which allows multiple control frameworks to be interoperable in resource discovery and reservation and allowing experimentation an even greater scale and very different architectures.[7]

1.2.2 Control Frameworks today

During these 3 years GENI is grown collaborating with newborn testbeds and changing the way the control framework can interact.

ORBIT was integrated together with VINI, another general purpose testbed, within Planetlab, and later ProtoGENI too, an aggregate manager or GENI API has been developed to make possible to all clusters (except ORCA for now) to communicate and lend resources to each other.

The native AM protocol in ORCA relies on features that are not yet available in the GENI AM API, therefore it has not been integrated yet.
Today it is possible, for an ORBIT user, to request slices on the other testbeds, for example, and the same applies to the others.

TIED/DETER adopted a security mechanism called **Attributed-Based Access Control (ABAC)** which allows control frameworks to share security information within a single control framework, as well as with each other that will be probably adopted by the others.

Other projects joined GENI, such as Lone-star Education And Research Network (LEARN) and iGENI which connects GENI with Starlight, **VISE** an actuator/sensor network, all integrated with ORCA; **PrimoGENI** a large-scale, real-time network simulator integrated in ProtoGENI; **COGRADIO** a stand-alone cognitive radio system (ORBIT) and **GENIcloud** which, as the name suggests, is focused on cloud computing. Planetlab, **CRON**, a testbed focused on optical networks and network emulation.

Generally, the number of projects affiliated with GENI doubles every Spiral, and new regional networks keep joining every year together with many Campus' networks thanks to Openflow and WiMax.

### 1.3 GENI's Resources

GENI's experiments run between endpoints, these endpoints are mainly nodes belonging to Regional and Campus networks that joined the community during the time, but how are all these networks connected?

The project has been supported since the beginning by two organizations, Internet2 and National Lambda Rail, that granted their infrastructures, the most important are the backbones that reach each zone of the United States, with a certain amount of dedicated bandwidth only for experimenters.

During Spiral Four phase, the deployment of Openflow-enabled backbone nodes by these two providers, will allow a custom control of routing on backbone level.
1.3.1 GENI Racks

During Spiral 4 began the deployment of specific GENI hardware to support the exponential growth of the network and enhance the control: two types of rack have been developed since now, **Exogeni** ones with the collaboration of IBM and **InstaGeni** ones with the collaboration of HP, both designed for meso-scale deployment and provided with GENI AM API.

**ExoGENI** - A high performance, highly viable and adaptable solution for virtual networks that support Openflow providing an excellent support for multi-site cloud applications, these ones are usually installed internally in a campus network, and each rack can run approximately 100 VM.

**InstaGENI** - A mid-range, modular GENI Racks solution that will be deployed at a great number of campuses, allowing Internet cloud applications support, along with Openflow and VLAN networking, but outside a site firewall.
1.4 PlanetLab

Among all the projects running on GENI platform, one of the most interesting and important is PlanetLab whose purpose is to establish a control framework that combines GENI components and user-level services into a coherent system through a control framework.[4]

Planetlab was born in 2002 as an independent project and later expanded and integrated within GENI.

Its assets are very heterogeneous, for example clusters, backbone nodes, enterprise-level nodes, and wireless nodes all of them independently controlled by international and corporate partners.

Besides it provides GENI users the access to its 1000+ nodes and creates and deploys component manager packages and clearinghouse packages.

The **GENI Clearinghouse** is the name given to a collection of related services supporting federation among experimenters and aggregates.

Planetlab is also federated with other major testbeds such as Onelab (European Planetlab), Korean and Japanese Planetlab.
1.4.1 OneLab

In Europe, the research for new solutions for Future Internet are conducted by the Future Internet Research Experimentation Initiative (FIRE) which needs as well a testbed the experiments.

Planetlab is a worldwide geographically extended testbed, but main decisions are taken in the USA, what could Europe do if its researchers want to test a different technology? US Universities will always have higher priority.

OneLab is the extension of the old Planetlab Europe, from a wired network to an hybrid one.

A high performance wired-only network does not represent faithfully the real Internet, instead, Planetlab purpose is to mix new technologies as they become available, for further testing.

- Wimax
- UMTS
- Wireless ad hoc networks (mesh network)
- Multihomed nodes
- Emulated nodes
Chapter 2

OpenFlow and Slice-based Federated Architecture

OpenFlow is an open standard, created at Stanford University as part of Clean Slate Program, and it was used to connect GENI to campus networks, it is compatible with both Planetlab and ProtoGENI control frameworks.

It enables researchers to run experiment on commercial switches, routers and access points.[9]

Commercial switches and routers do not, generally, neither offer an open software platform, nor provide the means to virtualize their software or hardware.

The standardized external interfaces are limited, (they serve only for packet forwarding), and all of the switch’s internal structure is hidden and it is unlikely that it will be similar to the one inside different branded switches, penalizing the researchers who seek for a standard platform to experiment with new ideas.

Further, network equipment vendors are understandably nervous about showing what is inside their devices: they spent years deploying and setting their own protocols and algorithms, and they are afraid that new experiments could bring new issues to the networks and of course, open platforms lower the market entry barrier for new competitors.

Due to the inflexibility, the poor customizability of commercial solutions and low performance and limited fanout of researchers ones, which are too general , it is necessary a compromise to have a certain degree of switch flexibility and some other characteristics:

- High-performance and low-cost implementations.
- Capacity to support a wide range of research activities.
- Isolation of experimental traffic from normal traffic.
- Satisfy vendors’ necessity for closed platforms.
It is a software that allows to customize the behavior of a device dividing the functions of "packet forwarding" and "path decision", without uncovering the inner architecture, the first one still resides on the device (Switch), while the latter is moved to a separate device (Controller), typically a standard server.[10]

The basic idea is simple: Openflow exploits the fact that the most modern Ethernet switches and routers use flow-tables for NAT, QoS, firewall, to collect statistics.

While each vendor’s flow-table is different, Openflow developers have identified a common set of functions available in many switches and routers, therefore they have decided to exploit them.

The OpenFlow Switch and Controller communicate via Openflow protocol which defines various types of messages.

The data path of an OpenFlow Switch is an abstraction of the clean flow table, whose entries contains a group of packet fields, and an action (such forwarding it from a port, modification of a field, or dropping the packet)

Figure 2.1

When an OpenFlow Switch receives a packet with no matching flow entries, therefore belonging probably to a new flow, it sends it to the controller whose role is to handle it in some ways: dropping it or adding a new entry in the Flow Table, telling the switch how to forward similar packets in the near future.
Since the Controller is programmable this standard allows an easy deployment of custom routing or switching protocols, security models, addressing schemes and even alternatives to IP. 

OpenFlow can easily provide users with their own isolated network, just as Virtual LANs do. 

The simplest approach is to statically declare a set of flows which indicates the ports accessible by traffic on a given VLAN ID. 

Traffic identified as coming from a single user (for example, originating from specific switch ports or MAC addresses) is tagged by the switches with the proper VLAN ID. 

A more complex and dynamic approach might use a controller to manage authentication of users and use the knowledge of their locations for tagging traffic at runtime. 

Additional features:
- Rewrite headers
- Map to queue/class
- Encrypt
- Support of multiple controllers for load balancing and reliability

### 2.1 The virtualization of the resources

Planetlab-like testbeds are made up of end-hosts, the routing process is normally done by Internet routers, these end-hosts are virtualized to avoid cross-experiment influence. 

First of all, the user that gained access to Planetlab/Onelab, gains a Slice[11]. 

A Slice is a set of allocated resources distributed across Planetlab (or any other testbed), for most users which means some number of virtual servers on Planetlab nodes, their function is to support many experiments in parallel and isolate them from each other. 

A Sliver is a slice running on a specific node, the user can use UNIX Shell to login a sliver. 

To enact virtualization, the user's sliver shares (equally) CPU cycles and network bandwidth with other users' slivers on the same node, running on Fedora virtual machines created on demand. 

This is important also from a security point of view, while the Node OS (based on Fedora Linux) isolates slivers, PlanetLab Central (PLC) remotely manages the nodes and grants slices if possible, checking nodes health and traffic.
The virtualization of a wireless link is much more difficult because there can be noise or obstacles that make it unstable, and choices about transmission power and channels must be made.

A successful experiment with a small slice allows the researcher to ask for a bigger slice and so on, in this way there is no waste of resources and only successful projects are tested on a very large scale.

Of course, there can be hundreds of slivers for each node, and thousands of slices scattered through the testbed.

2.2 Slice-based Federated Architecture

All actual testbeds have something in common, the SFA, which is the general mechanism for the management of shared resources.

SFA has different implementations according to the control framework, but the general structure is the same for everyone.[12]

The SFA recognizes four key types of entities operating through the control framework[13]:

- **Owners**, they control a portion of the network substrate, therefore they are responsible for externally visible behavior and establish high-level policies for how their portion is utilized.

- **Operators**, most of the times they work for owners, their job is to keep the platform up and running and prevent malicious or damaging activities, they also manage the hardware, its updates and faults.

- **Researchers**, the ones who effectively use the platform for experiments and measure aspects of the platform, they create, populate the slices and allocate resources to them.

- **Identity anchors** control authorization declaring the roles of other entities, they are also called Identity Providers. For example, an IdP might declare that a given user is a Principal Investigator (PI) representing a research organization that can authorize individual researchers to access the facility.
To mediate these activities the SFA defines three main actors:

A **management authority** is responsible for a subset of components, providing operational stability, ensuring that they behave according to the use policies established by the owner, and granting resource allocation within the specified limits.

A **slice authority** is responsible for one or more slices, naming and registering them and enabling users to access and control their own.

The slice authority must check information about the slice and respond to any misbehavior by the slice, at the same time, management authorities have the right to select what SAs have the authorization to create slices on their controlled resources.

A **user** is a generic entity that can have different roles, a researcher that intends to run an experiment or service in a slice, an operator that takes care of the maintenance, or an owner that contributes new resources to a facility.
2.2.1 Components

Components are the primary elements of the architecture, a component might be for example an edge computer, a customizable router or an access point.

Each component encapsulates a set of physical resources (hardware), logical resources (metadata, port numbers, etc) or synthetic resources (paths, links) and are grouped in aggregates under the control of the same Aggregate Manager, which exports a well-defined remotely accessible interface defining the operations available to the users.

The management authority establishes the policies about how the resources are granted to users, following the owner's will.

2.2.2 Deepening on slivers and slices

It is possible to grant a single component use to multiple users in two ways: virtualizing it, where each user acquires a virtual copy of the component's resources, or by partitioning it into distinct resource sets (where each user acquires a physical partition of the component's resources).

Both the virtual copy and the partition can be considered slivers.
Each Slice is composed by a set of slivers, it has also an associated group of users which are allowed to access its slivers and it is registered in the context of a Slice authority (SA), a principal that takes responsibility for the behavior of the slice in case it provokes some damages.

From a researcher's point of view, a slice is a distributed laboratory capable of running an experiment, while for the operator, slices are the primary abstraction for accountability.

Resources are acquired and used by slices, and external programs behavior can be associated to the relative slice.

Multiple experiments can run on a single slice at the same time, for each run, the experimenters may change parameters but leave the slice configuration unchanged (instantiation), or they may change either the set of components or the resources assigned on those components, or both.

Figure 2.4
Chapter 3

Future Internet Architecture (FIA)

These initiatives mentioned so far are not the only ones that are trying to give a new shape to the Internet.

In 2010, NSF funded four projects, each one with a different scope, as part of FIA program to increasing the efforts to discover new solutions for the future.

3.1 eXpressive Internet Architecture (XIA)

The main goals of this project are

- Extended support for enhanced security mechanisms

- Support for long term evolution of usage models, the network should be easily extended in case new entities add up in the future, avoiding the errors made in the past with IP

- Support for long term technology evolution, to exploit the dramatic increase of computational capabilities of both endpoints and network.

- Support explicit interfaces between actors, that allow these actors to function efficaciously. This principle can be applied both to the interfaces between users (applications) and the network, and between the providers that could offer new services.
XIA Components and Interactions

Figure 3.1

The core of the XIA is XIP, Expressive Internet Protocol, which supports communication between various types of principals.

XIA has classes of “locators”, which are called “identifiers”, or XIDs (160 bit), that make use of cryptographic hash technology.

**CID:** identifier for content C, is used if the router "knows" where the content is

**SID:** identifier for service S, which can retrieve the content

**HID:** identifier for host H, that is located the same place as the content

**NID:** identifier of a network N, where the host is located

Different XIDs trigger different **Per Hop Behavior (PHB)**, in the routers.

Through the XIDs is possible to evolve the "narrow waist" of networking, IP.

There is more than one kind of XID in each packet (they are also ordered), if the router does not have a matching PHB for the first falls back on the second on the list and so on, at the last fallback the service must be guaranteed.
This mechanism supports the future introduction of new classes of PHB, and the nesting of IDs makes routing more efficient.

NID PHB is very similar to today's routing.

The implementation of backward compatibility is realized with the 4ID mechanism, which is a XIA principal type that represents the IPv4 address, the connection of two or more XIA-enabled networks connected by an IPv4 network is realized with the encapsulation of the XIA packet with an IP header, but unlike the ordinary tunneling, the configuration is made by the network itself.

XIA is intrinsically secure because through the hashing of the content returns an ID, if it matches the requested one, the user received the correct content, this same principle can be applied to the other levels of the stack.[14]

![Diagram](image)

**Figure 3.2**

Probably among the four projects it is the easiest to develop and it presumes a classic approach to routing and according to the latest tests its performance can be compared to IP.

Nowadays XIA is in a prototyping stage and with an early release available to the public.

This early release includes the mechanisms to set up a XIA network on a private network or GENI testbeds, C & Python socket APIs for XIA (Xsocket API) needed for the development of new XIA applications.

The XIA Project presented a demo of the XIA prototype at the 12th GENI Engineering Conference (GEC) in Kansas City on November 2nd, 2011.
The demo used four ProtoGENI nodes, and the XIA Team used content retrieval, video streaming and live services to demonstrate various capabilities enabled by XIA, including the transitioning from IPv4 to XIP.

3.2 Mobility First

The Mobility First project is founded on the assumption that the Internet is approaching an historic change, with mobile devices and applications replacing the fixed-host/server model.

The major design goals of this architecture are mobility extended to all, an host is no more bound to the point-of-access of a network but it has its own address assigned dynamically, robustness taking into account the problems and merits of use of the wireless medium, trustworthiness in the form of improved security and privacy measures for both mobile networks and wired infrastructure.[16]

The design is also conditioned by technological factors, for example, radio spectrum scarcity and wired bandwidth abundance.

Mobility First has, like XIA, classes of identifiers: content, service, host, network called Global unique Ids (GUID), routers have separate PHB for each class.

The address has two levels, GUID and NID (or NAddress), the GUID must be routable within the network of NID, this implies separation between names(ID) and network (NA) which is used for a "fast path" if available, with a fallback on GUID if needed.

  - Sender gets GUID from a name service.

  - **Global Name Resolution Service (GNRS)** maps GUID to current NID, the receiver must keep this mapping current.

  - As the packet is forwarded, if network N does not know about the GUID, it can re-query the GNRS to get updated location.
The routing is made segment-by-segment with storage routers and not end-to-end like TCP, introducing resilience against variable wireless links and disconnections.

It supports multicasting, anycasting and multihoming and takes advantage of cheap storage in the network (storage-aware routing), so if the requested content is in the router's cache the advantages are huge.

While XIA relies on nested addresses for routing, Mobility First is trying to explore new ways such as Distributed Hash Tables.

Another important feature is the Dynamic Spectrum Assignment (DSA), given that the majority of end-user devices are wireless, Internet spectrum should be assigned on demand so an algorithm coordinates neighboring access points.
One of Mobility First's ambitions is to connect vehicles with vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) modes, which require features such as location services, georouting, and efficient multicast.

Mobility First too, uses a classic approach to routing and the user has little control on it.

As for XIA, Mobility First group released a prototype working with Linux devices, with its own MF-socket API and it has been tested on ProtoGENI testbed during 2011, with edge-components relying on both WiMax and Wifi.[15]

The tests were done using very different devices such as vehicular nodes, Android clients, Linux laptops, sensors and with some of them multihomed.

### 3.3 Nebula

The Nebula's research is on network support for Cloud Computing with a particular focus on high-assurance application and fine-grain control of forwarding path by all actors.[17]

Every potential actor is represented by a domain with its own behavior **Per Domain Behavior (PDB)**, which is arbitrary.

A policy is a sequence of domains through which the packet must pass, all actors must agree that this sequence of PDBs is acceptable and the order must be respected.

**Nebula Data Plane (NDP)** forwards the packet according to a pre-set policy, all actors must specify what can offer and services constraints so **Nebula Virtual And Extensible Network Techniques (NVENT)**, can compute the policies obtaining an acceptable sequence of PDBs.[18]

- Before a sender can send a packet, it consults the NVENT system.
- NVENT calculates the route through the domains, and returns it to the sender, it is called the “proof of consent” (PoC). Each domain must give its consent, for this operation to succeed, and it is NVENT's duty to check their disposability.
- As the packet is forwarded by the NDP, each domain monitors that it performed its service, computes the proof, which is put into one of packet's fields the “proof of path” (PoP).
- Through the use of advanced cryptographic techniques, each domain can inspect the PoP to that point and confirm that the previous steps actually happened.
The computation of policies certainly brings a consistent overhead and packets are 20% larger.

The **NEBULA Core (NCore)** is a mechanism that periodically interconnects enterprise data centers containing replicated data with ultra-high availability next-generation core routers developed in collaboration with Cisco.

Nebula is very difficult to implement but it is the most detached from the past, that's why it is currently behind XIA and Mobility First, but the need for more advanced hardware is also important.

The first prototype of NVENT and NDP will be implemented using the RapidNet declarative networking system, one of the interesting opportunities presented by integrating this system with NDP is the opportunity to perform a variety of analysis and verification security policies at runtime and prior to deployment.
3.4 Named Data Network

NDN is another architecture that sees content as the core around which everything else should be built on.

As stated in [23]: "Today’s applications are typically written in terms of what information they want rather than where it is located, then application-specific middleware is used to map between the application model and the Internet’s.

With NDN the application’s what model can be implemented directly, removing all the middleware and its associated configuration and communication inefficiencies".

The Data (or content chunks), becoming an independent entity, are named instead of the their location as happens with IP, names can follow a hierarchy allowing the routing to be scalable.

This innovation allows a different security method, securing the content itself instead of the transmission channel between host and server which is no more required, besides routers can cache content (because now it can be uniquely identified) enabling faster delivery to the end-users.[20]

Two kinds of packets: interests and data, **NDNI and NDND**.

NDNI: contains a **Content Name, CN**.

NDND: contains the CN, the effective content, and a signature.

Here is an example on how NDNI and NDND are treated by the routers:

- An host requests some content sending an NDNI
- As the NDNI arrives at a router:
  - The router checks the NDNI and if the requested content, the NDND, is there, in the Content Store (a buffer memory). If so, it returns it.
  - In a local table, called Pending Interest Table, it records the NDNI and its source path.
  - Forwards the NDNI to the next router.

- An NDND arrives at a router.
  - The router sees if there is a matching NDNI stored.
  - If so, forwards the NDND along the path, if not, it drops it

Each Data packet is handled independently from the routers: they do not take into account the source or destination to decide whether to cache its content to satisfy possible future requests, which means that there is no flow state.
That is the reason NDN does not have a separate transport layer, because it moves its functions directly up into application, the names are opaque to the routers so that applications can "choose" their naming scheme. [19]

An NDN network is able to operate above unreliable packet delivery services, avoiding the connectivity problems of mobile networking, guaranteeing the delivery of content chunks as opposed to IP. Interest Packets that are not satisfied within a reasonable period of time are retransmitted by the requesting application if it still wants the data.

The routing protocol is an **extension of OSPF for NDN** called "OSPF-N" (intra-domain), using new types of Opaque Link State Advertisements (LSA) to carry NDN routing information.

NDN is efficient in the delivery of popular content, through the caching of NDND packets and is resilient against most DDoS attacks, because the attacked can drop unwanted NDND and the caching will diffuse the attack on many routers.

![Diagram](image.png)

**Figure 3.6**

NDN routing relies on packet naming therefore avoids the four main problems that addresses cause in the IP architecture: **limited address space**, is no more a problem since namespace is boundless, **NAT traversal** because address is no more needed in order to offer content, **mobility**, that requires switching addresses in IP no longer interrupts communication since data names remain the same, **scalable address management** is no longer required in local networks, which is particularly useful for sensor networks.

This last point brings some issues because it is difficult to maintain under control routing table size due to the quantity of content on the Internet.
Currently, there is an early prototype of NDN running on 5 Planetlab nodes: the main concepts of NDN Project are implemented on each node (including **Data Caches and Pending Interest Tables**), that can send and receive Interest and Data Packets through ad hoc UDP Tunnels. The NDN Applications Team continues to develop and test different applications working over an NDN network, it verifies what the ones which make a better use of these new technology are: one of the early examples is **ccnchat** a simple chat tool often used to test connections and testbed management.

Testers can be connected from everywhere in the world supposing that firewalls do not block tunneling.
Conclusions

The aim of this work is the evaluation of how the research of new solutions for a more efficient and reliable Internet is progressing.

I considered both the new means that are made available to the researchers to experiment with and new architectures itself that are on that are on the way to an almost complete implementation.

My activity started gathering information on actual testbeds, how the major ones are being built and how their development works, the set of physical resources they own and the different objectives they have been created for.

The next step was to understand how these networks are controlled and provide their services to the operators, the policy behind the assignment of resources and how a heavily heterogeneous set of hardware is virtualized.

The core of this study on the way Internet is going to evolve in near future is, of course, the introduction to the new architectures that will replace the actual ones when they will be too obsolete or too inefficient to handle the vast amount of content on the Internet, that piles up every year.

Surprisingly, even being strongly detached from the past and not a simple improvement of current technologies, these architectures are already in a testing phase and other ones are designed by other Research Institutes allowing for a future healthy competition or for the adoption of more them for different purposes.
Bibliography


URL: http://groups.geni.net/geni/attachment/wiki/SpiralOne/GENIS1Ovrvw092908.pdf


URL: http://www.docstoc.com/docs/87635779/FINDing-a-GENI-in-a-CCCastle


URL: http://groups.geni.net/geni/attachment/wiki/SpiralOne/GENIS1Ovrvw092908.pdf

URL: http://groups.geni.net/geni/attachment/wiki/SpiralTwo/GENIS2Ovrvw060310.pdf

[10] Nick McKeown; "OpenFlow (Or: Why can’t I innovate in my wiring closet?)" The Stanford Clean Slate Program
URL: www.openflow.org/documents/OpenFlow.ppt


URL: http://groups.geni.net/geni/attachment/wiki/SliceFedArch/SFA2.0.pdf


URL: http://www.icsy.de/studium/seminar/ws1112/presentations/XIA.pdf


[16] D. Raychaudhuri; "MobilityFirst: A Robust and Trustworthy Mobility-Centric Architecture for the Future" Internet PIMRC Keynote, Sept 13, 2011

URL: http://mobilityfirst.winlab.rutgers.edu/documents/PIMRC_keynote_MF_911.pdf


URL: http://nebula.cis.upenn.edu/NEBULA-WP.pdf

[18] "Security research in the NEBULA Future Internet Architecture" Cornell University

URL: http://nebula.cis.upenn.edu/FIA_PI_Public.pdf


URL: http://infq.lab.imtlucca.it/_pdf/slides/Future_Internet


[22] http://www.geni.net/

URL: http://www.named-data.net/techreport/TR001ndn-proj.pdf
List of Figures

- **Figure 1.1** "GENI Facility: Motivation"  Ellen W. Zagura; "FINDing GENI in a CCCastle". Reinventing the Internet (and more) Conference, January, 2007 pag 21
- **Figure 1.2** "GENI Prototyping Plan"  Chip Elliot: "GENI, Exploring Networks of the Future" March 30, 2009 pag.21
- **Figure 1.3** "GENI Spiral Two Project Map"  The GENI Project Office; "GENI Spiral 2 Overview” June 3, 2010 pag.8
- **Figure 1.4** "GENI Control Frameworks Structure"  Chip Elliot: "GENI, Exploring Networks of the Future" March 30, 2009 pag.37
- **Figure 1.5** "Map of NLR Points of Presence"  http://www.telepresenceoptions.com/2010/02/nlr_internet2_to_develop_unifi/
- **Figure 2.1** "Openflow"  Nick McKeown; "OpenFlow (Or: Why can’t I innovate in my wiring closet?)" The Stanford Clean Slate Program pag.8
- **Figure 2.2** "Control functions of SM and OM"  "GENI System Architecture: Overview and Narrow Waist"Larry Peterson John Wroclawski 1st GENI Engineering Conference Oct 10,2007 pag.24
- **Figure 2.3** "SFA Management Structure"  Ellen W. Zagura; "FINDing GENI in a CCCastle". Reinventing the Internet (and more) Conference, January, 2007 pag 37
- **Figure 2.4** "SFA Interface among heterogeneous testbeds"
- **Figure 3.1** "XIA Components and Interactions"  "XIA: An Architecture for a Trustworthy and Evolvable Internet" Peter Steenkiste Dave Andersen, David Eckhardt, Sara Kiesler, Jon Peha, Adrian Perrig, Srinij Seshan, Marvin Sirbu, Hui Zhang Euro-NF Course on Future Internet Architecture Kaiserslautern, March 8, 2012 pag.9
- **Figure 3.2** "XIA Actors and Communication"  "XIA: An Architecture for a Trustworthy and Evolvable Internet" Peter Steenkiste Dave Andersen, David Eckhardt, Sara Kiesler, Jon Peha, Adrian Perrig, Srinij Seshan, Marvin Sirbu, Hui Zhang Euro-NF Course on Future Internet Architecture Kaiserslautern, March 8, 2012 pag.8
- **Figure 3.3** "Mobility First Actors and Communication"  "Content Discovery and Delivery in MobilityFirst" F. Zhang, K. Nagaraja, T. Nguyen, D. Raychaudhuri,Y. Zhang pag.6
- **Figure 3.4** "Mobility First Segmented Routing"  MobilityFirst Future Internet Architecture Project Project Update – Part I Oakland FIA Meeting, May 2011 pag 10
- **Figure 3.5** "Nebula Packet Structure"  "The NEBULA Future Internet Architecture" pag.13
- **Figure 3.6** "Internet Hourglass and NDN Hourglass"  A Survey of the Research on Future Internet Architectures" Jianli Pan, Subharthi Paul, and Raj JainA pag.4
- **Figure 3.7** "NDNI and NDND Packet Structure"  "Named Data Networking (NDN) Project" October 31, 2010 Lixia Zhang, Deborah Estrin, and Jeffrey Burke Van Jacobson, James D. Thornton, and Diana K. Smetters Beichuan Zhang Gene Tsudik Dan Massey Christos Papadopoulos Tarek Abdelzaher Lan Wang Patrick Crowley Edmund Yeh pag.13