# A Vision of the Next Generation Internet: A Policy Oriented Perspective[1]

Subharthi Paul*, Raj Jain*, Jianli Pan*, Mic Bowman[†]

*Department of Computer Science and Engineering
Washington University in Saint Louis
*{pauls, jain, jp10}@cse.wustl.edu*

[†]Intel Systems Technology Lab
Intel Corporation
*mic.bowman@intel.com*

**The host centric design of the current Internet does not recognise data and end-users as integral entities of the system. The first generation of Internet has been very successful and yet business, organizations, governments are finding it difficult to enforce their policies on their networks with the same ease that they do other methods of communications and transport. Ad-Hoc solutions e.g. firewalls, NAT, middleboxes etc, that try to mitigate these issues end up providing localized myopic fixes which often hurt the basic underlying principles of the original design. We envision the future internet to be a dynamic, heterogeneous, secure, energy efficient ubiquitous network flexible enough to support innovations and policy enforcements both at the edge and the core. The first step towards the next generation is the redesign of naming and name binding mechanisms. We, therefore, propose a Policy Oriented Network Architecture (PONA) and an abstract two part protocol stack with a virtualization layer in between. We also introduce the concept of generalized communication end-points – hosts, users, data/services, instantiate the ideas with the Mapping and Negotiation layer and provide an integrated framework for the next generation Internet.**

*Keywords: Next Generation Internet, Policies, Virtualization, Naming, Name Binding, Mapping and Negotiation, Realms, Zones.*

## 1. INTRODUCTION

The original Internet design was host centric where it was believed that the network would be the infrastructure between two hosts wishing to communicate with each other. Also the original design was around a system of stationary end hosts in a friendly trust-all environment of universities and government agencies. It is obvious that the environment has now changed. Today, Internet is the primary means of communication inside and between organizations. The original academic endeavour is now the world's largest commercial communication infrastructure. It is a complete virtual world in itself – the biggest market, the biggest commercial transaction arena and the single largest source of information. The beauty of the original design was in its simplicity and elegance. With the increased pressure on its design and with the exponential rise in its popularity, the internet design had to accommodate quite a few standard and non-standard extensions. While few of these extensions were planned and hence properly researched and engineered, many extensions were ad-hoc and were undertaken with a myopic outlook to achieve quick results. Such planned and unplanned extensions have not only made the design complicated but also attributed to a huge chaos in trying to define the basic underlying building principles.

The current Internet usage is "data centric" as evidenced by the popularity of the peer-to-peer applications. Data centric view abstracts a data requestor from having to know where the data or service comes from. Also, the end-to-end paradigm of the transport layer becomes a problem for many mobile applications. The security paradigm too, has become one of the greatest concerns in the current internet. In our design of the next generation Internet we realize this evolution and make way for them to be incorporated into the basic architecture.

We advocate a two part abstract modelling of the communication stack as shown in Figure 1. The lower part is the infrastructure, responsible for actual physical connection between two communicating entities and the upper part is the end-to-end logical connection between the communicating entities. Between these two layers, there needs to be a hybrid layer that maintains the logical connections and maps them to physical connections. This layer acts as a virtualization layer, trying to realize any sort of virtual end-to-end connection over the infrastructure. The idea of the two part abstract protocol stack is based on the separation of concerns of the communication support system of

[1] This research was sponsored in part by a grant from Intel Research Council.

the infrastructure from the actual communication between the two entities. In a way, the current TCP/IP based protocol stack implements a similar idea wherein the IP layer and below is concerned with actual delivery and the TCP and above is concerned with the end-to-end data paradigm. However, in the present stack, the transport and upper layers are strongly bound to the identifiers in the IP layers, mostly IP addresses. This renders the separation in-effective. We propose a virtualization layer between the transport and network layers that realizes this separation. Apart from maintaining upper layer logical connections, the virtualization layer also allows for the realization of multiple virtual communication end-points as opposed to the host-only end-point idea of the present protocol stack. Details of this idea, its benefits and a high level instantiation of the ideas proposed are discussed in the rest of this paper.

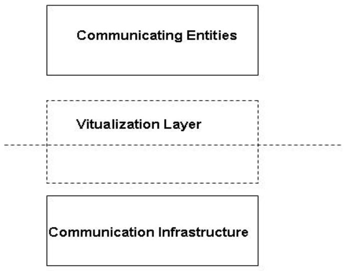

**Figure 1:** Two Part Abstract Model

The core of this proposal is the framework of a new naming architecture called "Policy Oriented Naming Architecture (PONA)". We show how we can achieve the above requirements and many more. Some of the new concepts proposed in PONA framework are a **hierarchy of realms**, which follow the organizational structure of commercial organizations. This way each realm can enforce its own policies on the traffic while also providing services to its members. PONA objects can designate proxies to represent them even when the object is away or sleeping (for **energy efficiency**). PONA objects have IDs that do not change when they move and so other objects can reach the **mobile objects** using their IDs. Separation of ID and addresses is not new but the hierarchical organization of IDs to match the **organization structure** and their use in providing services is unique. PONA distinguishes **network connectivity** from organizational ownership. Network service providers can enforce their own policies as the packets leave their network to other service provider or customer networks. This is possible by an address hierarchy and **zones**.

We begin with Identity/Locator split architecture in the lines of other such architectures proposed in the past [1, 37, 11, 31, 34]. However, unlike past efforts, we present an integrated and efficient approach taking into consideration all the implicit and explicit factors dictated by the commercial nature of the network applications. Also, in designing PONA framework, we make no assumptions about the structure of the Identifiers. It allows the co-existence of multiple Identifier types as relevant within its scope. We believe that the way to go forward with a new design is to ensure enough commercial motivation towards its realization and that the design be efficient and feasible at the same time. Also, unlike past efforts, we realize the importance of end users in the communication process as against the "end host" paradigm and make way for its presence explicit, in the architecture. We integrate the concepts of "host centric", "data centric" and "user centric" approaches in one integrated system architecture.

In this proposal, we present the basic ideas of PONA and show that it is a very powerful architecture that provides many new features. This will open up opportunities for other researchers to design details of these new features. As a part of this proposal we plan to concentrate on designing the naming architecture.

The rest of the paper is organized as follows: Section 2 discusses the data-, user-, host-centric model that forms the basis of the endpoint generalization paradigm of PONA. In Section 3, we discuss the basic design principles underlying our proposed architecture followed by the details of the "Mapping and Negotiation" (MN) layer in Section 4. Section 5 deals with the "Policy Design Principles" followed by a discussion on how PONA helps realize some of the objectives for the future Internet in Section 6 and Section 7.

## 2. DATA-, USER- AND HOST-CENTRIC MODELS

PONA objects are classified as hosts, users, and data. Hosts are electronic computing entities, e.g., computers, palmtops, firewalls, routers, and network attached storage. Data objects represent information stored or transmitted in the form of bits, e.g., music, movies, and documents. Data objects reside on hosts and often multiple copies of the data are available from multiple hosts. Users are human objects or user agents. In order to communicate over the Internet, users need to connect to a host. Their connectivity to hosts changes frequently as the users move from one system to the next. User objects are part of a user realm. For example, John.Intel is a member of Intel realm (organization) and as a result has certain privileges and responsibilities that apply to Intel employees. Host objects are part of host realms. Data objects are part of data realms. It is possible although not necessary that user, host, and data realms are part of the same organization. Even if they are part of the same organization, the policies for managing users, hosts, and data are typically very different. There may be restrictions on which hosts are accessible to which users and what data can reside on them. Data objects have their own access control lists and so on.

Many applications require that the network be more data centric and that it should move away from its original host-centric design. Data centric view abstracts a data requestor from having to know where the data comes from. PONA provides a generic architecture which allows us to implement **data-centric**, **host-centric**, and **user-centric** Internet architecture. Figure 2 shows a simple dependency diagram among key players in a basic communication scenario. Most of the communication on the Internet can be characterized as “user wanting to access data”. For example, a user wanting to listen to a music file or accessing a web page, or downloading a file. Even user to user communication can be represented as one user supplying data to the network while the other receiving data from the network. Users connect to the network via Hosts. The data resides on hosts. Hosts have location. The principles of a “data centric”, “host centric”, and “user centric” network can all be realized by adding certain semantic meaning to this diagram. Here, solid black arrows represent dependencies and the dotted red arrows represent peer-to-peer associations.

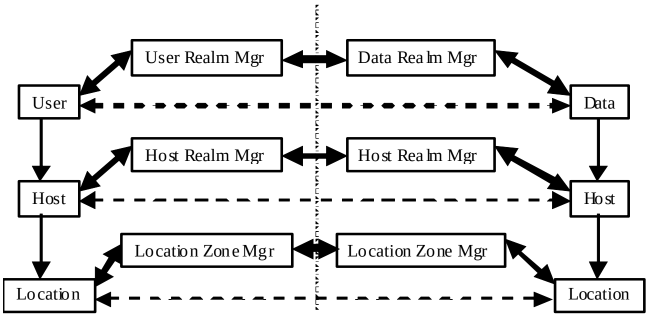

**Figure 2:** Dependency Diagram between User, Hosts, and Data

In a host-centric architecture, as in the current Internet, the host is the central player in all exchanges and all data is specifically directed to or retrieved from a specific host. Data is tightly coupled with a host. In this architecture, a possible request is that I want music file x.mp3 from host y. We have to resolve to a specific host. In practice, most users simply want the music file x.mp3 regardless of which host it comes from. It is difficult to make that request in the current architecture since data file x.mp3 is not considered an object and network understands only hosts not data. Data is not considered a separate entity and networks resolve to hosts rather than data.

The data centric approach vests more importance to data and tries to address data, with the network dynamically resolving the data to a publisher host. The data centric approach is considered better in the sense that computing and networking paradigms have changed over the years and today it does not matter where data comes from as long as it is available and reliable. Also, host centric approaches cannot deal with data mobility and data replication within the framework and need to depend on external roundabout means to support them.

These arguments extend to the “user centric” paradigm as well. Since end users are an integral part of a communication process, they should be realized in the mainstream architecture as well. In a user centric view, the communication terminates at the user and not at the host. So when the user moves from one host to the next the communication continues. As we shall see, user centric realization adds a huge amount of flexibility to the whole system. Basically, it adds the user’s perspective into the architecture making room for personalized services and hence promoting innovations.

PONA introduces a new Mapping and Negotiation Layer (MN Layer) between the transport and network layers of the current stack. The MN layer is responsible for making the contexts of the host, data and users explicit in the protocol stack, thus reducing the inter-layer coupling. The MN layer is in a way a hybrid layer between the IP infrastructure and the strictly end-to-end transport paradigm providing space for the existence of middleboxes and realizing and integrating them into the mainstream architecture. The discussion here shall seem more meaningful after we discuss the basic design principles behind PONA and hence we defer the explanations till then.

## 3. BASIC DESIGN PRINCIPLES

Before moving to the details of the proposed architecture, it is essential to consider some of the design principles and state the rationale behind those principles. In this section, we identify some of the key design principles of PONA and also discuss the issues which lead us to these findings.

### 3.1 Layer Independence

Layer Independence refers to minimization of inter-layer coupling between the protocol layers. In the present TCP/IP based Internet design the upper layers namely the application and the transport are very strongly coupled to the IP layer through their strong bindings with the IP addresses. Such strong coupling is responsible for the strictly host-centric approach of the current design. IP addresses indicate the location of the host and so when the hosts move, TCP has difficulty keeping the connection up. A number of location/Identity split architectures have recently been proposed that address this concern and advocate the use of host identifiers to identify hosts independent of their address in the IP forwarding infrastructure. We move a step forward and propose the need to establish the freedom of upper layer entities such as users, agents, data and services from being bound to a particular host. Balakrishnan et al [15] propose a similar idea of having independent names for each layer of the protocol stack. Our proposal is different from [15] in that we are talking about “object stack” which is different from the protocol stack. When we discuss protocol stack, we address the added concerns of layer and end-to-end paradigm violations.

### 3.2 Hybrid Layer

Transport layer of the present internet defines end-to-end semantics. However, we believe that rendering data handling to be always end-to-end is often extremely restrictive and inhibit the existence of models other than host-centric. The MN layer is a hybrid layer between the infrastructure and transport in that it provides intermediate “transfer points” to maintain the end-to-end semantics above it and thus make room for other types of network models, such as disconnected operation. The MN layer helps to realize end-points of the end-to-end paradigm generically as objects rather than the restrictive idea of endpoints being hosts in the original host-centric internet.

### 3.3 Policy Enforcement Points

TCP connections are end-to-end. The end-to-end paradigm is extremely beneficial in ensuring end-to-end flow control, reliable delivery and security. However, they render policy enforcement points on data to be illegal violation of the end-to-end semantics. The present TCP/IP stack provides no such interim legal points of policy enforcements on a transport connection except for the source and destination. Contrary to the initial design requirements, the present Internet is in need of such policy enforcement points as evident from the wide-scale deployment of middleboxes in the path of end-to-end connections. PONA realizes many levels of policy enforcements, user-to-data, host-to-host, infrastructure-to-infrastructure, user-to-host, data-to-host, etc. An example of user-to-data policy is which users can access a data object regardless of the data location or user location.

### 3.4 Realms and Zones

Realms refer to high level logical aggregations of objects based on organizational, administrative or commercial relationships. Zones refer to topological aggregations of the infrastructure. The realms and zones have managers that provide Identities to the network entities and act as the points of policy enforcements, negotiation boundaries and mobility anchors. The concept of realms and zones was originally designed for security and trust relationships. We extend their meaning to the present form. In a way, realms and zones provide for a legal space for the deployment of middleboxes.

### 3.5 Directives, not Addresses

In the current internet initial name resolutions through DNS results in IP addresses. This too is a result of frozen design principles of the original host-centric design. We realize, that such a scheme is restrictive and hindrance to generality. Hence, we propose that initial name resolution should result in a set of directives rather than a fixed IP address. A directive is a set of bindings of the desired data/service/user/host to a host address, a Host ID, Service ID, Data ID, etc. The lower layers will further refine this mapping to a specific instance of the desired object depending upon the initial choice. Such an approach leads to multiple benefits. Firstly, it provides backward compatibility to stacks that do not implement the Mapping and Negotiation Layer (Discussed in Section 4) and stick to the original host centric approach. Also, it provides backward compatibility to existing applications that reside over these stacks. Secondly, it supports late binding of objects to their locations and thus provides more dynamicity. Thirdly, it establishes the policy oriented paradigm by forcing a connection establishment to go through policy enforcement points.

### 3.6 Independent States of Application and Transport

Applications typically maintain end-to-end application states. The connection oriented transport protocols too maintain end-to-end transport layer states. State-full applications and transport connections are thus bound to physical hosts. The only way to provide mobility of application and transport connections over hosts is to provide for a mechanism for them to offload their state and restart on a different host based on the preserved state. Session layer protocols try to snoop application state from application packets and help mobile applications rebuild themselves. We believe that such methods are not effective and violate layered architecture. Applications and transport protocols cannot and should not maintain state in a generic way. They should be allowed to maintain state in their own proprietary way. The work of the protocol stack is just to provide means for them to offload their state if and when they want and deliver it to newer instances when asked for. PONA works around this principle by allowing applications and transport protocols a means to preserve their state across host-to-host mobility rather than trying to build state information on its own.

These six principles form the basis of the rest of our proposal and we believe that these principles shall be relevant for any design for the next generation Internet. PONA is just an instance of these design principles. PONA might very well be replaced by other schemes in the future which better realize the principles stated above.

## 4. A PROTOCOL STACK FOR THE NEXT GENERATION INTERNET

As indicated earlier, the network consists of an infrastructure consisting of a set of hosts on which data and services reside. Users use the network services via hosts. In a sense the network consists of two major parts: Infrastructure on the bottom and data and services on the top. The role of the infrastructure is to provide points of attachment to the network. These points of attachments are uniquely identified by locators (or addresses). The infrastructure has protocols to find the optimal path from one address to another. This is similar to what is done in the bottom 3 layers of the TCP/IP stack – namely IP, data link, and physical layer. Although we believe that certain changes in IP will make it more efficient, we do not want to dwell on that here since we want to concentrate on the higher part consisting of data and services.

The Services/Data part is responsible for sending/receiving data between the entities who are interested in it. In the current stack, the transport layer and application layer make up this layer. Some of the issues that crop up in the services layer are those of reliability, error correction, congestion management, and flow control etc. The problem with the present design is that all these layers are tightly coupled with the IP addresses making the design host centric and also unfit to support mobility of hosts and users and multiplicity of data.

We envision a new network stack design for the next generation Internet as shown in Figure 3. Above the infrastructure, we introduce a new layer called the "**Mapping and Negotiation Layer"** or "MN Layer" and we generalize the Transport Layer and call it the "**Transfer layer**".

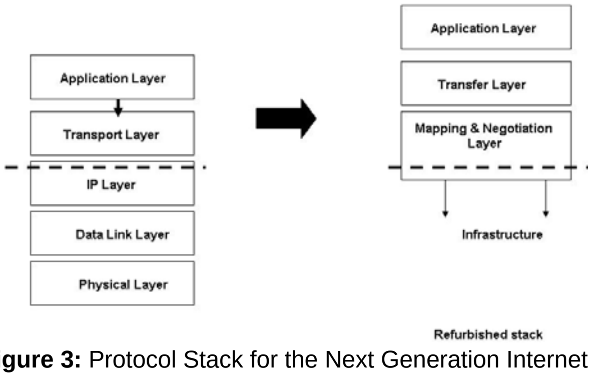

**Figure 3:** Protocol Stack for the Next Generation Internet

The MN Layer bridges between the Infrastructure and Services part of our two-part abstract model and incorporates the design requirements and solutions identified in the Data-Host-User model. This layer is actually an aggregation of three sub-layers: the user/data identity layer, the host identity layer and the zone identity layer. As indicated above, the host identity layer is a distributed layer over a host realm principally responsible for maintaining the host identity and locator mapping at all times to allow host mobility over the Infrastructure.

The user identity layer (or data identity layer) is also a distributed layer over the user realm (or data realm) principally responsible for maintaining user (or data) independence and hence mobility from one host to another.

The zone identity layer maintains zone ID's and their relationships with zone managers. The zone ID and zone manager ID's represent subscription of the communication endpoints with the infrastructure. The communication end-points in this case may be any of users, hosts or data/services. The zone ID layer is responsible for authentication, authorization and accounting of infrastructure usage, authentication of infrastructure end points and intermediate relay points, mobility of communication endpoints over multiple zones with business partnerships and providing differentiated QoS based infrastructure services.

The Transfer layer is the generic incarnation of today's transport layer [8] but one that can bind to a user/data/service ID, host ID or a zone ID/Locator/IP. The present TCP is a member of this generic set as one which can bind only to end-host's IP address. The "Transfer Layer" connections are thus still end-to-end with the difference that the end-points need no longer be IP addresses of hosts.

An end host, as in any network-able device that can be addressed over the infrastructure, has a Host ID. When a user logs into the device, he/she registers his or her User Id and Host Id with the User Realm manager. Similarly, data objects register their existence on a host with the data realm manager. Note that registrations use host Ids and not their addresses. The translation of the host ID to addresses is the responsibility of the Host realm manager. The translation changes as the host moves. Thus, our model allows for independent movements of users, data, and hosts. The relationship between users (or data) and hosts can be one-to-many allowing a user (or data) to be available on multiple hosts. Each user can have multiple Ids as authorized by the realm manager including an "anonymous ID" in which case the communicating parties are not made aware of the real identity of the user. However, this "anonymous ID" may be extremely restricted in privileges.

The term realm manager, till now, has been used to mean the object which maintains the mapping between the identifier in its layer and the horizontal layer beneath it, e.g., UID.URID → HID.HRID. Here URID is the id of the realm in which user ID "UID" exists. Similarly, "HRID" is the host realm ID in which Host ID "HID" exists. The primary function of the realm mangers is to maintain the mappings and update the mappings as and when needed. This function is the basis for the network supporting all kinds of mobility for the users, hosts and data/services. However, realm managers can and shall have much greater function than just this. The realm manager may enforce security functions in the form of user, data or host authentication. Realms may be organized in any such manner which suit the purpose they serve. The association between different realm managers leads to negotiations which might refer to security negotiations, transfer agent negotiations etc. A user may belong to multiple realms and have multiple ID's. Each Id is expressed as UID.URID. Thus, the realm can provide the user a middlebox service in which URID object acts as a "transfer point" (virtual end point) for all external communications.

It is completely on the realm designer as to how he organizes the realm structure. However, for purposes of interoperability of realms, each realm needs to have an ID. Objects within the realm have ID's which are local to that realm. Object ID's within the realm may not carry any meaning to any other object outside the realm but may be semantically overloaded within the realm to reflect some organizational ordering. A group of realms may form some sort of security association wherein all or some resources may be shared. Note that realms really represent an organizational entity and so have features that most organizations have. We may go on talking about all the things that can be done with realms and the negotiations between them. This really makes our future network a hot seat of innovations.

It is to be noted that the Mapping and Negotiation Layer represents a "layer" in the protocol stack and not a particular protocol instantiation. It is expected to host a family of protocols that abide to the broad specifications of a virtualization layer requirement, the details of which are beyond the scope of the current paper. Also, the M&N layer is not an end to end layer and it does not come in the path of the data. It is more of a virtual provisioning layer that provisions the maintenance of heterogeneous virtual network protocols to exist within it and end-to-end layers above it may choose the best virtual network that suits their purpose. As an example, let us suppose that a new session layer protocol has been designed to support disconnected operation. Such a session layer may aim itself to be deployed on top of a virtual network that provides the best support to its design goals by defining intermediate service points that act on behalf of a host with interrupted connectivity.

To summarize, the MN layer is not just an Id-Locator Mapping layer but acts as a virtualization layer which realizes generic object connections and allows heterogeneous topologies depending on the realm structure over a common infrastructure. In this way, PONA is effective in an organizational stringent policy oriented and homogeneous scenario as well as the more general heterogeneous and loose policy enforced scenario of the public internet.

## 5. PONA POLICY DESIGN PRINCIPLES

One of the key contributions of the PONA architecture is its intrinsic support for policy enforcements on data, services, users and hosts. Based on the architecture defined in Section 5, PONA supports a layered policy enforcement scheme in which the hierarchical ID's in each layer intrinsically define the policy enforcement points.
Some of the policy design principles on which PONA is based can be stated as follows.

### 5.1 Layered Policies

All communication scenarios typically follow a layered structure. As evident from the discussion on data, user, and host centric models in Section 2, there is an inherent layered orientation between the various objects involved in an electronic communication system such as the Internet. The realization of this inherent layered-ness in the policy framework is thus necessary. We believe that policy enforcements are most effective if applied between peer entities of a communication process in a layered fashion. PONA, thus, applies its policies between user-data, source host-destination host, home zone – visited zone and source host zone – destination host zone, etc.

### 5.2 Hierarchical Identities

Hierarchical ordering, in any scenario, represents a distribution of responsibilities. Same is true for the framework of policy oriented network architecture. Identifiers at the various levels necessarily need to consist of hierarchical semantics, which explicitly represent the ordering of policy enforcement points. PONA advocates the use of such hierarchical IDs at the various layers in the form of [Object-id.Realm-Manager-id] structure of its ID's.

### 5.3 Local and Global Enforcements

A policy aware network stack needs to be able to define global and local enforcement points. While global enforcement points can be distributed, such as realms as discussed above with regards to PONA, local enforcement points may be at the layers of the stack themselves. As an example, in PONA, the User/Data Id layer may enforce a policy of allowing only users/data belonging to certain realms to register themselves on them. Similarly, the host id layer may enforce a policy of allowing only some particular <host id.host.realm ids> to register themselves on it. As an example, all hosts in XYZ Corporation may have the policy of not allowing any ID's to be installed on them except for <some host id.XYZ corporation id>.

### 5.4 Hard and Soft Policies

Hard policies refer to policies whose lapses are intolerable. Soft policies refer to policies whose lapses are tolerable and hence negotiable. An example of hard policy may be authentication. Authentication lapses are generally considered intolerable in any communication environment. On the other hand authorization lapses may often be negotiable. In a layered architecture as in PONA, such negotiation is generally strictly top-down. As an example, suppose a user is authenticated and authorized to get some data by the data realm. However, the host realm of the user's machine, though authenticated, is not authorized to access the particular data server host on which the data is hosted. Such a scenario may be quite relevant in an organizational setup where the CEO needs some data to which he is authorized but he may be accessing it from a host which is not. Such scenarios, call for negotiations if possible. One basis of negotiation may be the sensitivity of the data. Data below a certain level of sensitivity may be allowed to override host level authorizations. However it should be noted, that such negotiations are only possible top-down, that is, if a the user is not authorized to get the data, no negotiations should be possible. Another interesting example, very relevant to a peer-to-peer scenario could be when a user is authorized to access some data by the owner of the data, but the users host is not allowed to access the peer host on which the data is hosted. Such a scenario calls for the P2P system to look for replicated data on other hosts. This leads to a whole new area of research which we prefer to call "Policy Enforced Anycasting".

The above discussion tries to formalize some of the Policy design principles that we think are essential for any policy oriented architecture for the future Internet. Once again, we emphasize that PONA is just am instance that tries to imbibe these principles within it. These principles can be looked upon as the blue-print of future design endeavours of this nature.

## 6. HOW PONA SUPPORTS FEATURES OF THE NEXT GENERATION INTERNET

The future Internet is expected to support various exciting and innovative features. Frozen design decisions of the present Internet design pose a hindrance to such innovations, both at the edge as well as the core. In PONA, we see the potential of providing a generic protocol level support to such innovations and many more. In this section, we briefly explain how PONA features described above help support some of the intuitive requirements identified for the next generation Internet.

### 6.1 Mobility

Separation of IDs and addresses in PONA clearly helps objects move freely. The proposed architecture together with newly introduced MN layer provides an elegant support for mobility of end users with session manageability, mobility of data and services among different hosts and mobility of end hosts across various locator zones. The control algorithm which shall be a part of the MN layer shall ensure scalable and efficient mobility by separating the data plane from the control plane. Hence, unlike other mobility solutions as in Mobile IP [5, 6, 7], I3 [18] etc., our architecture does not suffer from the problem of data triangulation under normal circumstances.

### 6.2 Security

PONA already has 3 features that work together to provide mechanisms for implementing strong security: well-defined bounded context, separation of management, control and data plane, and policy servers.

First, PONA has a concept of boundary with well-defined gates in a realm or zone. The administrative policies of the realm or zone are enforced at these points. This is similar to what is done currently in organizations and countries. Each organization has a well defined set of entry points manned by security guards or receptionists who check the credentials of all persons entering or leaving. Once a person enters a building, he/she has authorization to move in certain areas but is again subject to further verification if they want to enter sensitive areas. Similar effect is achieved in PONA by realms.

Second, one key reason for the insecurity of current Internet is that data, control, and management planes are intermixed. A host can easily send a message that looks like a routing message. Telephone networks, on the other hand, are considered more secure because the control lines used for communication between switches are physically separate from the data lines used for transmission of voice packets from the customers. Unlike the phone network, PONA does not require a physically separate control network. Rather the separation is logical so that the packets on one plane cannot penetrate the other.

Third, the concept of servers allows security to be achieved much easier than that in the current Internet architecture. An authentication server in a realm can help authenticate all realm members and verify authentication of other correspondent. Even low power devices like personal digital assistants (PDAs) and palm-tops can use totally authenticated communication. In the absence of strict boundaries provided by the realms and the separation of control and data planes, this type of authentication service is not possible because some outsider can easily pose as an authentication server.

The arguments given above for authentication servers also apply to other security services, such as, encryption, privacy, anonymity and to other administrative policies such as resource usage and priorities.
Having a bounded trust domain in the form of a realm also allows PONA to use sophisticated security mechanisms such as very large keys, physical tokens, and biometrics, etc.

### 6.3 Energy Efficiency

The bounded trust domain provided by realms and concept of servers help PONA achieve energy efficiency. PONA objects can delegate any part of their responsibility to servers in the realm. This proxy is simply an entry with proper notation in the realm registry for ID to address translation. An extension of the proxy server concept allows PONA objects to go to sleep. The objects can wake up or be awakened to handle networking tasks as necessary. Many of the energy efficiency concepts from sensor network research and wireless research can be extended for application to wired devices as well.

### 6.4 Representation of Organizational Structure and Enforcement of Organizational Policies

This should be obvious by now that the key driving force behind PONA design is our requirement to represent organizational structure via realms. Each realm represents an organization entity and the realm hierarchy represents the organizational hierarchy. The network connectivity represented by zones may or may not be the same as the organizational hierarchy.

### 6.5 Non-Electronic End-Systems

PONA objects do not have to be computers or electronic devices. A human being is a valid PONA object and will have a PONA name and ID. User IDs are dynamically mapped to Host IDs which are dynamically mapped to Host addresses. The person is linked to its computers, PDAs, and phones via visual or auditory links. The recipient can indicate to the realm manager his preference for devices to which the incoming connection requests should be directed to. In fact, the person itself may be a realm with different devices in/on his body being members of the realm.

### 6.6 Location Transparency

The basis of a location transparency scheme is indirection. The method is to employ some proxy to receive data on the user's behalf and relay it to the user. Location transparency comes for the price of data triangulation. Our architecture can support a more efficient location Transparency than proposed in I3 [18]. Such a solution is possible with the active support of the user's realm managers. The idea is that, suppose a host belongs to certain realm Verizon.St_Louis and moves to Paris. It may elect a host on its behalf at the Verizon.London realm and have all data triangulated to it through the host in the Verizon.London realm which may be in London at that time, thus reducing the effect of triangulation. For users (and data/services) a similar method may be applied wherein the user's ream manager maps the user to a trusted host id which can proxy for the user's host Id.

## 7. ADDITIONAL FEATURES

Here we list certain additional features which should be relatively easy to support based in the design of PONA. However, the exact details of how they will be realized are open to future research efforts.

### 7.1 Generic Transport Layers

The introduction of the MN layer makes the transport selection more explicit. What this means is that each data (or user) realm manager may now dynamically install a transfer mechanism which is optimized for its kind. For example, when a user declares its wish to watch some movie X in realm Netflix, the realm manager of movie X, in addition to resolving the host ID of the movie X, may also indicate the transfer protocol module to be installed for this data type. Realm manager of the user may similarly indicate the specific transfer module characteristics for the user. This will help in proper data translation (different screen sizes) and presentation of the movie. The point is that host ID (and hence address) is only one of the several characteristics that is resolved by the realm managers.

### 7.2 Transport Level Gateways

The architecture also supports transport layer gateways by entering into such associations between the realm managers. The realm managers may negotiate or choose some third party transport layer gateways to which both resolve their host ID's to and force all data to pass through these gateways. This helps in realizing heterogeneity in the mainstream design wherein it should be possible to reach an offbeat mote network from an Internet host without the mote having to implement the standard transport layer.

### 7.3 Delay Tolerant Networking

The key solution to hosts with intermittent connectivity is to choose another host as its caretaker host and download data from the source to the caretaker host [25]. When the original host is again available, the data is offloaded to it. In our architecture, the realm managers can easily appoint caretakers. Also, the delay tolerant networking is equally feasible at the user level as well as the host level.

### 7.4 User Session Transfer

User mobility entails user session transfer for supported applications wherein a user may move to a different host and have his session transferred to that host. This would enable revolutionary and innovative services like transferring a live music session from the user's laptop at home dynamically to his smart phone in the car and to his office PC in his office. Of course, such dynamicity can be achieved only with highly refined presence protocols and ubiquitous networking environment, but the point is that the proposed architecture can support such and many more extravagant efforts.

### 7. 5 Defined Business Motivations

The proposed architecture provides clear and well defined business motivations. The infrastructure providers as the owners of the underlying physical infrastructure and the Internet Service providers, providing addressing and forwarding over the physical medium generate their revenues on bits of data that they deliver. The data realm and host realms too may be commercial organizations whose basic service is to provide mapping functions to support user, data and host mobility, and may generate revenues on the value added services as in security, disconnected connectivity, location transparency, outsourced secondary storages etc.

### 7. 6 Effect on Locator design

The proposed architecture shall simplify the numbering of locators (addresses), making them simpler and more efficient than those in the current Internet. Currently, the IP addresses are used as locators as well as identifiers. IP addresses however suffer from the problem of non-synchronization between its administrative class hierarchy and its functionality as a topological locator. This leads to non aggregated mapping tables leading to inefficiency. The concept of ID's and realms frees the infrastructure from its administrative role and now the host locators can be numbered to imply only topological locators and be highly aggregatable.

The list of such features and more is long and is only bounded by one's imagination. In summary, the basic design principles behind the PONA architecture are flexible and generic enough to support and realize a wide set of paradigms within the basic architecture itself rather than rendering them to be disparate, non-standard ad-hoc solutions.

## 8. RELATED WORK

Naming and addressing is a fundamental aspect of network architecture and so the number of papers in this area is enormous as listed in the references. The idea of overloaded IP addresses and hence the need for a locator/identity split was first proposed by Saltzer [24]. We have prepared a survey paper that provides a summary and categorization of this past work [62]. In this section, we focus on some of the recent naming and identity/locator splitting proposals.

Host Identity Protocol (HIP) [37] is one of the identifier locator split designs by the Internet Engineering Task Force (IETF). HIP introduces a new public keys based namespace. Mobility and multihoming are also under development in some drafts. However, there is no concept of organizational structure of name spaces or organizational policies. HIP based proposals generally advocate processing on flat ID's as has been proposed in many DHT based peer-to-peer schemes[3, 4, 16, 20, 29, 41].Internet Indirection Infrastructure (I3) [18] adds an overlay indirection infrastructure above the routing network to provide better multicast, anycast, and mobility. The mapping from identifier to address is called "trigger" stored in the overlay servers. I3 also introduces a globally unique flat namespace for the identifiers and does not address the separation of organizational membership and connectivity of objects.HI3 [1, 34] proposes a scheme based on using HIP based Identifiers as triggers in I3. Shim6 [12] uses IPv6 addresses as the Upper Layer Identifier (ULID). It doesn't introduce any new namespace and allows multihoming. Location/ID Separation Protocol (LISP) [11] uses IP-in-IP tunneling to split identifiers from locators. It requires using provider independent (PI) address as identifiers, which may limit the scalability of the routing system. GSE [30] divides IPv6 address into identifier and locator parts, and uses a NAT-like style to manage the network. Multihoming is not considered in GSE. MILSA [21] proposes an innovative scheme of having a SIP-like [22, 39] control plane between the IP and TCP layers using URI like IDs. FARA [10], Forwarding directive, Association and Rendezvous Architecture presents an abstract meta architecture of a naming scheme based upon the decoupling of end system identity from IP addresses thus resolving the problem of IP overloading.

The semantics of names has been discussed in several works. The most successful such scheme which has seen world-wide adoption has been the DNS [33]. Some of the other schemes worth mentioning are the Uniform Resource Name (URN) [26, 27] and the Uniform Resource Locator (URL). [35] proposes a service naming scheme for large scale multi-domain networks and [32] discusses a name space model for locating services. Our present work does not make any assumptions about the semantics of the names. As a design principle PONA allows any naming scheme as long as the semantics are understood by the concerned entities.

The IP address depletion problem was addressed by the Address Lifetime Expectations Working Group [17]. Network Address Translation (NAT) proposed the most popular ad-hoc solution to circumvent the address depletion problem. However the lack of standardization of such an ad-hoc solution led to the development of different flavours of NAT rendering un-interoperability. Moreover NAT introduced the problem of isolated private address spaces, host within which could instantiate a communication session but could not be reached directly by external hosts. A few schemes to remove this weakness were proposed [23] but could never be standardized owing to the non-standardized nature of the problem itself. A more permanent solution to this problem was proposed in the form of 128 bit IP addresses in IPv6 [38]. IPv6 also proposes methods to support mobility circumventing the problem of data triangulation. However, wide scale deployment of IPv6 is still very much in the future.

The NSF funded GENI [14] program is one of the biggest initiative towards design of the next generation internet. Some of the other noteworthy efforts towards the future evolution of the present internet may be referred to in [2, 9, 13, 14, 19, 28, 36]

Anycast Name Resolution (ANR) [42] is a NSF/FIND funded project. It proposes a data centric approach in which data is a named entity and the network plays an active role in resolving the present location of data through a routing method based on flat labels. Our approach supports data centric approach too, but it is more general and supports host centric as well as user centric approaches. We do not freeze a method for how data is discovered and managed. Our approach is to provide support for whatever method might suit a particular commercial context. It might be through table lookups, transitive trust based pathways based on distributed trust reachability and routing methods or virtual ID based routing on an overlay network. Also, in our design we make the contexts of “data”, “user” and “host” explicit to be able to address and introduce policies explicitly on each of these entities. We believe this has important connotations in design of an economic and administrative model which shall support the architecture. A method of “anycasting” in the application layer is discussed in [13].

## 9. SUMMARY

In this paper, we have identified that a proper naming and name binding mechanism is the key to attain de-coupling between the communication processes and the communication infrastructure. The naming solution proposed in this work can form the basis of upper layer virtualization where communicating entities are made explicit and shall connect over a virtual framework rather than having a defined physical connection. We believe that such an approach would open up the Internet design for immense innovations and shall also provide an effective standard solution to most of the problems being faced at present. This paper provides an outline to the key ideas behind this philosophy and also presents an instantiation of the ideas through the design of the Policy Oriented Network Architecture.

## REFERENCES

[1] A. Gurtov, D. Korzun and P. Nikander, “Hi3: An Efficient and Secure Networking Architecture for Mobile Hosts”, HIIT Techinal Report, June 2005.

[2] A. Perrig, D. Clark, and S. Bellovin, Editors, "Secure Next Generation Internet”, NSF Workshop Report, July 2005, http://www.nsf.gov/cise/cns/geni/ngsi.pdf.

[3] A. Rowstron and P. Druschel, “Pastry: Scalable, decentralized object location and routing for large-scale peer-to-peer systems”, Proceedings of the 18th IFIP/ACM International Conference on Distributed Systems Platforms (Middleware 2001), Heidelberg, Germany, November 2001.

[4] B. Y. Zhao et al. , “Tapestry: A Resilient Global-Scale Overlay for Service Deployment”, IEEE Journal on selected area in communications, Volume 22, No. 1, pp. 41-53, January 2004.

[5] C. Perkins, “IP Mobility Support for IPv4”, RFC 3220, January 2002.

[6] C. Perkins, Ed., “IP Mobility Support for IPv4”, RFC 3344, August 2002.

[7] C. Perkins, “IP Mobility Support”, RFC 2002, October 1996.

[8] DARPA Internet Program, “Internet Protocol”, RFC 791, September 1981.

[9] D. Clark, et al., "New Arch: Future Generation Internet Architecture”, Technical Report, Air Force Research Laboratory, Rome, NY, December 31, 2003, 76 pp., http://www.isi.edu/newarch/iDOCS/final.finalreport.pdf.

[10] D. Clark et al., “FARA:Reorganizing the Addressing Architecture”, ACM SIGCOMM 2003 Workshops, Karlsruhe, Germany, August 25-27, 2003.

[11] D. Farinacci et al., “Internet Draft: Locator/ID Separation Protocol (LISP)”, draft-farinacci-LISP-03, Aug 13, 2007.

[12] E. Nordmark, M. Bagnulo, “Internet Draft: Shim6: level 3 multihoming Shim protocol for IPv6”, draft-ietf-shim6-proto-09, October, 2007.

[13] E. W. Zegura et al., "Application-layer anycasting: a server selection architecture and use in a replicated Web service”, IEEE/ACM Transactions on Networking (TON), Volume 8, Issue 4, August 2000, pp. 455 - 466.

[14] National Science Foundation, "Global Environment for Networking Innovation”, http://www.nsf.gov/cise/geni/

[15] H. Balakrishnan et al., “A Layered Naming Architecture for the Internet”, Proceedings of ACM SIGCOMM’04, Volume 34, Issue 4, Portland, Oregon, USA, Aug. 30–Sept. 3, 2004.

[16] H. Balakrishnan, Scott Shenker, and Michael Walfish, “Peering Peer-to-Peer Providers”, 4th International Workshop on Peer-to-Peer System (IPTPS ’05), February 2005.

[17] Minutes of Address Lifetime Expectations Working Group. Proceedings of 29th IETF Meeting, Seattle, April 1994.

[18] I. Stoica et al., “Internet Indirection Infrastructure”, ACM SIGCOMM ’02, Pittsburgh, Pennsylvania, USA, 2002.

[19] J. Crowcroft et al., "Plutarch: an argument for network pluralism”, Proceedings of the ACM SIGCOMM 2003 workshop on Future directions in network architecture, Karlsruhe, Germany, pp. 258 - 266.

[20] J. Eriksson, M. Faloutsos, and S. Krishnamurthy, “PeerNet: Pushing Peer-to-Peer Down the Stack”, Proceedings of the 2003 International Workshop on Peer-To-Peer Systems (IPTPS ’03), Volume 2735, pp. 268-277, 2003.

[21] J. Pan et al., MILSA: A Mobility and Multihoming Supported Identifier Locator Split Architecture for Next Generation Internet, March 2008, submitted for publication, available from authors.

[22] J. Rosenberg, H.Scheulzrinne, G.Camarillo, et al., “SIP: Session Initiation Protocol”, RFC 3261, June 2002.

[23] J.Rosenberg, J. Weinberger, C Huitema, R Mahy, “STUN - simple traversal of user datagram protocol(UDP) through network address translators(NAT), March 2003, RFC 3489.

[24] J. Saltzer, “On the naming and binding of network destinations”, RFC 1498, August 1993.

[25] K. Fall, "A Delay-Tolerant Network Architecture for Challenged Internets”, Computer Communication Review, 2003, Vol. 33; Part 4, pages 27-36.

[26] L. Daigle, et al, “Uniform Resource Names (URN) Namespace Definition Mechanisms”, RFC 3406, Oct 2002.

[27] L. Daigle, et al, “URN Namespace Definition Mechanisms”, RFC 2611, June 1999.

[28] L. Kleinrock, "A vision for the Internet", ST Journal for Research, Vol.2, No.1, pp. 4-5, November 2005. http://www.lk.cs.ucla.edu/PS/STMJrnl2005.pdf

[29] M. Caesar et al, “ROFL: routing on flat labels”, ACM SIGCOMM ’06, Pisa, Italy, Sept. 11-15, 2006.

[30] M. O’Dell, “Internet Draft: GSE - an alternate addressing architecture for IPv6”, draft-ietf-ipngwg-gseaddr-00, February 24, 1997.

[31] M Walfish, H Balakrishnan, “The location/identity split is useful for middleboxes, too”, Proceedings of Workshop on HIP and Related Architectures, November 2004.

[32] N. Hinds and C. V. Ravishankar, “Name space models for locating services”, Proceedings of the 1991 conference of the Centre for Advanced Studies on Collaborative research (CASCON '91), October 1991.

[33] P. Mockapetris, “Domain Names- Implementation and Specification”, RFC 1035, November, 1987.

[34] P. Nikander, J. Arkko, and B. Ohlman, “Host Identity Indirection Infrastructure (Hi3)”, Proceedings of Second Swedish National Computer Networking Workshop, 2004.

[35] R. Ahmed, R. Boutaba, F. Cuervo, et al, “Service naming in large-scale and multi-domain networks”, IEEE Communications Surveys & Tutorials, Volume 7, Issue 3, Third Quarter 2005.

[36] 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.

[37] R. Moskowitz, P. Nikander, "Host Identity Protocol (HIP) Architecture”, Internet RFC 4423, May 2006, 21pp., http://www.ietf.org/rfc/rfc4423.txt

[38] S. Deering and R. Hinden, “Internet Protocol Version 6 (IPv6) Specification”, RFC 2460, December 1998.

[39] S. Guha, Y. Takeday, P. Francis, “NUTSS: A SIP-based approach to UDP and TCP connectivity” , SIGCOMM 2004 Workshops, August 2004.

[40] S. Paul, J. Pan, R. Jain, “A Survey of Naming Systems: Classification and Analysis of the Current Schemes Using a New Naming Reference Model”, WUSTL Tech Report, March 2008, to be submitted for publication, available from authors.

[41] S. Ratnasamy et al. , “A Scalable Content-Addressable Network”, ACM SIGCOMM’01, San Diego, California, USA, August 27-31, 2001.

[42] S. Shenker, I. Stoica, “A New approach to Internet Naming and Name Resolution”, http://www.nets-find.net/Funded/NameArchitecture.php,