Report Number: WUCSE-2010-31

2010

Multi-Tier Diversified Service Architecture for Internet 3.0: The Next Generation Internet

Subharthi Paul, Raj Jain, Jianli Pan, and Chakchai So-in

The next generation Internet needs to support multiple diverse application contexts. In this paper, we present Internet 3.0, a diversified, multi-tier architecture for the next generation Internet. Unlike the current Internet, Internet 3.0 defines a new set of primitives that allows diverse applications to compose and optimize their specific contexts over resources belonging to multiple ownerships. The key design philosophy is to enable diversity through explicit representation, negotiation and enforcement of policies at the granularity of network infrastructure, compute resources, data and users. The basis of the Internet 3.0 architecture is a generalized three-tier object model. The bottom tier... **Read complete abstract on page 2.**

Multi-Tier Diversified Service Architecture for Internet 3.0: The Next Generation Internet

Subharthi Paul, Raj Jain, Jianli Pan, and Chakchai So-in

Complete Abstract:

The next generation Internet needs to support multiple diverse application contexts. In this paper, we present Internet 3.0, a diversified, multi-tier architecture for the next generation Internet. Unlike the current Internet, Internet 3.0 defines a new set of primitives that allows diverse applications to compose and optimize their specific contexts over resources belonging to multiple ownerships. The key design philosophy is to enable diversity through explicit representation, negotiation and enforcement of policies at the granularity of network infrastructure, compute resources, data and users. The basis of the Internet 3.0 architecture is a generalized three-tier object model. The bottom tier consists of a high-speed network infrastructure. The second tier consists of compute resources or hosts. The third tier consists of data and users. The “tiered” organization of the entities in the object model depicts the natural dependency relationship between these entities in a communication context. All communication contexts, including the current Internet, may be represented as special cases within this generalized three-tier object model. The key contribution of this paper is a formal architectural representation of the Internet 3.0 architecture over the key primitive of the “Object Abstraction” and a detailed discussion of the various design aspects of the architecture, including the design of the “Context Router-” the key architectural element that powers an evolutionary deployment plan for the clean slate design ideas of Internet 3.0.

2010-31

Multi-Tier Diversified Service Architecture for Internet 3.0: The Next Generation Internet

Authors: Subharthi Paul, Raj Jain, Jianli Pan, Chakchai So-in

Corresponding Author: {spaul, jain,jp10,cs5}@wustl.edu

Web Page: http://www.cse.wustl.edu/~jain/

Abstract: The next generation Internet needs to support multiple diverse application contexts. In this paper, we present Internet 3.0, a diversified, multi-tier architecture for the next generation Internet. Unlike the current Internet, Internet 3.0 defines a new set of primitives that allows diverse applications to compose and optimize their specific contexts over resources belonging to multiple ownerships. The key design philosophy is to enable diversity through explicit representation, negotiation and enforcement of policies at the granularity of network infrastructure, compute resources, data and users. The basis of the Internet 3.0 architecture is a generalized three-tier object model. The bottom tier consists of a high-speed network infrastructure. The second tier consists of compute resources or hosts. The third tier consists of data and users. The “tiered” organization of the entities in the object model depicts the natural dependency relationship between these entities in a communication context. All communication contexts, including the current Internet, may be represented as special cases within this generalized three-tier object model. The key contribution of this paper is a formal architectural representation of the Internet 3.0 architecture over the key primitive of the “Object Abstraction” and a detailed discussion of the various design aspects of the architecture, including the design of the “Context Router-” the key architectural element that powers an evolutionary deployment plan for the clean slate design ideas of

Type of Report: Other

Multi-Tier Diversified Service Architecture for Internet 3.0: The Next Generation Internet

Subharthi Paul, Raj Jain, Jianli Pan, Chakchai So-in

Washington University in St. Louis
Department of Computer Science and Engineering
One Brookings Drive,
Saint Louis, MO 63130
{spaul, jain,jp10,cs5}@wustl.edu

ABSTRACT

The next generation Internet needs to support multiple diverse application contexts. In this paper, we present Internet 3.0, a diversified, multi-tier architecture for the next generation Internet. Unlike the current Internet, Internet 3.0 defines a new set of primitives that allows diverse applications to compose and optimize their specific contexts over resources belonging to multiple ownerships. The key design philosophy is to enable diversity through explicit representation, negotiation and enforcement of policies at the granularity of network infrastructure, compute resources, data and users. The basis of the Internet 3.0 architecture is a *generalized three-tier object model.* The bottom tier consists of a high-speed network infrastructure. The second tier consists of compute resources or hosts. The third tier consists of data and users. The "tiered" organization of the entities in the object model depicts the natural dependency relationship between these entities in a communication context. All communication contexts, including the current Internet, may be represented as special cases within this generalized three-tier object model. The key contribution of this paper is a formal architectural representation of the Internet 3.0 architecture over the key primitive of the "Object Abstraction" and a detailed discussion of the various design aspects of the architecture, including the design of the "Context Router-" the key architectural element that powers an evolutionary deployment plan for the clean slate design ideas of Internet 3.0.

Keywords

Network Architectures, Next Generation Internet, Future Internet, Service Oriented Architectures, Multi-Tier Diversified Architecture, Overlay Networks, Diversified Network Architectures, Object Abstraction, Cloud Computing, Distributed Computing

I. INTRODUCTION

Internet 3.0 is an effort to define a new architectural basis for the future Internet. Leveraging years of experience with the current Internet design and related research efforts to modify/improve it, Internet 3.0 proposes a multi-tier diversified architecture that deviates from the existing "one suit fits all" paradigm of the current Internet model and proposes an architecture design inspired by the "requirement specific networking" philosophy being actively discussed in next generation networking communities [FIND] [GENI] [FIRE].

The current Internet, designed around the modest needs of file transfer and resource sharing applications fails to satisfy the diverse needs of modern distributed applications. All these years, the original design has been overlaid with external (and often architecturally ugly) incremental mechanisms to satisfy the requirements of specific contextual needs. These encroached mechanisms, solely designed for satisfying specific requirements often introduce inconsistencies and non-determinism into the overall architecture. Also, these mechanisms are severely restricted in their effectiveness owing to inherent constraints imposed by the original design. The Internet is rife with instances of such tussles, be it between P2P providers and ISP's [ROB08][BAN07][ON606][WAT05][AGG08][ON607][AGG07], NAT mechanisms and end-to-end protocols [HOL01][SIE00][HOL00][HAI00][SRI99][ABO04][STE00][MON00], policy

This work was supported in part by a grant from Intel Corporation and NSF CISE Grant #1019119.

control mechanisms and security mechanisms [LEH07][JOR07][SYV00], underlay routing policies and overlay routing requirements [AND01][LIU05][NAK03][LEE08], etc. The non-determinism manifests in the fact that a new internet-wide standardized mechanism can no longer be guaranteed to perform, as determined, across the whole system and also it could potentially break some of the existing mechanisms.

We attribute the cause of these tussles, broadly to two primary weaknesses in the current Internet design:

1. Lack of adequate diversity leading to selfish contextual innovations on a shared substrate.

2. Lack of adequate policy framework preventing policy expression and enforcement at the required level of granularity.

On the surface, these two causes might seem unrelated. That is why, existing proposals [TUR107][AKARI][FEDERICA] advocating the "requirements specific networking" theme, attack the first cause by enabling diversity in the architecture through virtualization - isolated sharing of resources amongst multiple co-existing contexts on a shared substrate. However, we argue that both these causes are in tandem, leading to the two *design philosophies* underlying the Internet 3.0 architecture [PAU10]:

- ***Diversity naturally follows ability to express and enforce policy at the required level of granularity:*** Mostly, although diversity is in the offing, choices can not be enabled owing to the lack of a proper policy framework allowing policy expression, enforcement and negotiations. As an example, years of research and multiple technically sound solutions later, QoS routing could not be widely deployed over the Internet. The reason can be traced to the lack of a proper business framework wherein multiple autonomous systems could negotiate their individual services and aggregate them to provide an end-to-end inter-domain QoS routing service to applications that need it. Also, the extent of diversity depends directly on the level of granularity of policy enforcements. An example can be cited in the per-flow and flow-class managements of Inteserv[BR94] and Diffserv[KN98], respectively.
- ***True diversity can be achieved only through the explicit separation of policy from functionality:*** An example in support of this design philosophy may be seen in the current Internet inter-domain routing where individual autonomous system (AS) policies can effect the global state of the distributed routing algorithm [TAN05a][TAN05][FEI06][FEA05][GAO00]. AS relationships govern routing quality. The reason can be attributed to the conflation of the routing state to represent both, reachability information as well as AS level policies.

In Internet 3.0, the implementation of these design philosophies is realized through; 1) explicit separation and representation of network infrastructure, host, data and user resources, 2) explicit representation of resource ownership and ownership-linked policies on resources, 3) explicit representation of application contexts as a federation of resources belonging to multiple ownerships, 4) explicit policy enforcement and negotiation plane that allows federation of multi-ownership resources to map the requirements of specific application contexts, and 5) a management and control plane that co-ordinates the interaction of the policy plane with the functional plane implementing functional diversity such as novel forwarding paradigms, dissemination topologies etc, optimizing specific application contexts.

Traditionally, "overlay" mechanisms have been proposed over the Internet "underlay" to enable application specific functional diversity. However, the design primitives of the Internet such as the unicast forwarding paradigm, end-to-end to design principles, single-path routing, etc while contributing to the simplicity and hence large-scale success of the Internet, are optimized to serve a specific communication context. The Internet design is thus inherently non-optimized to serve as the underlay for multiple diversified application contexts. More recent proposals on Overlay Hosting Platforms (OHPs) [TUR207] advocate third party overlay service providers to host multiple overlay services implementing application specific diversity. OHPs shall allow multiple packet processing contexts to co-exist over the Internet through deployment of interposed specialized OHP nodes. These OHP nodes shall lease general purpose compute resources and programmable packet processing resources to hosted application contexts. Distributed OHP nodes shall be connected through pre-provisioned network links. The network of OHP nodes is a single ownership fixed network and application contexts hosted over it are restricted by the

distribution (number of nodes, density, diversity, etc) and topology of the underlying OHP network. On the contrary, Internet 3.0 proposes a diversified Internet architecture over distributed ownership. Internet 3.0 federates resources leased from multiple ownerships (infrastructure domains or Autonomous Systems, cloud computing platforms, data domains, etc) through explicit policy negotiations to dynamically spawn specific application contexts.

The primary primitive of the Internet 3.0 architecture is the **Object Abstraction.** Data, host and infrastructure are the broad classification of resources that constitute a networking context and are established as **"entities"**. These entities are organized in tiers, representing the natural dependency among the resources representing them. The bottom tier represents high speed networking infrastructure. The middle tier represents hosts and the top tier represents data. The entities belonging to the different tiers are overlaid with the ownership framework of **"realms."** Realms advertise realm-specific services through **"Objects."** Objects encapsulate the complexities of resource allocation, resource sharing, policy enforcements etc and expose a standard interface representing capabilities (in terms of standardized abstract parameters) and fixed or negotiable policies.

"Services," within the Internet 3.0 architecture refer to aggregated objects belonging to same or multiple ownerships, composed over a **policy negotiation plane**, to exhibit a common set of attributes and policies. Thus, object composition in Internet 3.0 is a non-trivial function and lies at the basis of the policy and security framework of the architecture.

In the rest of the paper, we shall first discuss the generalized three-tier object model in Section II that lies at the basis of the Internet 3.0 architecture followed by the formalisms of the object abstraction (Section III) and object composition principles (Section IV). The architecture of Internet 3.0 is discussed in Section V followed by a survey of related efforts in the past in Section VI. We finally conclude in Section VII.

II. GENERALIZED THREE-TIER OBJECT MODEL

Internet 3.0 is a "*communication-paradigm*" based architecture as opposed to the "*communication-system*" based architecture of the current Internet. The difference between these two architectural approaches being that in the "communication-system" based architecture the underlying communication primitives evolve largely independent of the specific needs of the application context that is installed over it, while the "communication-paradigm" based architecture ideally allows the communication primitives to evolve per the specific requirements of application contexts. The three-tier object model (Figure 1) represents the reference framework for the "*generalized communication-paradigm*" over which the Internet 3.0 architecture is based.

The key elements of the three-tier object model are:

A. ENTITIES: Data, Users, host (or compute resources) and infrastructure represents entities. Entities are broad classification of resource types. Any communication instance is implicitly organized as an interaction between these entities. However, the conflated design of the current Internet neither allows these interactions to be explicitly associated with the individual entities nor allows these entities to specifically enforce their policies. The core of the multi-tier diversification architecture lies in making inter-entity interactions explicit and designs a framework for active negotiation of policies between them.

B. REALMS: Realms overlay entities with a discreet ownership framework. Ownership entails related administrative and management responsibilities. In the "Three-tier Object Model" (Figure 1), the bottom tier infrastructure is owned by multiple infrastructure owners. The second tier of hosts is owned by individual users or different organizations such as DoE, DARPA, Amazon, etc. The third tier of users and data may belong to specific organizations or individual users. Thus, realms represent logical divide of entities into multiple ownership and management domains. Each realm is managed by a "Realm Manager." Explicit representation of ownership simplifies the design of the policy framework through more natural representation and enforcement of policy rather than conflating them with functionality as in

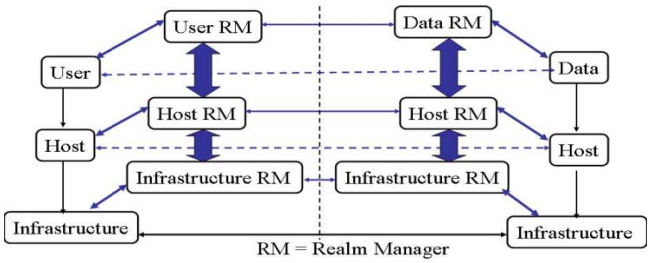

Figure 1. Generalized Three-tier Object Model

the current Internet. As already mentioned, an example of such conflated policy representation can be seen in the inter-domain routing function of the present Internet where a route advertisement to a specific prefix by a routing domain (or autonomous system) represents an implicit agreement on the part of the routing domain to provide transit for packets destined for that prefix. This in turn indirectly conflates the inter-AS commercial relationship such as provider-customer relationships or peering relationships. While, this implicit representation is sufficient for a single (best-effort) service network with an unnaturally enforced valley-free routing policy, it is not sufficiently expressive to allow multiple service types (such as QoS, etc) or policies (security, accountability, financial agreements, etc).

C. REALM HIERARCHY: Realms are organized hierarchically. The highest level of the hierarchy, the root realm, represents ownership. The concept of ownership also entails the highest administrative and management authority for objects lying within this realm. Subsequent levels of this hierarchy represent more specific administrative and management boundaries. Note that the hierarchy is not a binary tree since a realm can have two or more parents, i.e., an organization can be part of several higher-level organizations and can have several lower level sub organizations.

The realm hierarchy is of primary importance to the "abstraction" concept of objects as they represent different abstraction boundaries of objects. More concretely, in a specific scenario, the lowest tier realm hierarchy could be responsible for managing the resource allocation to objects and abstracting them as generic capability parameters across the realm boundary. The next higher level realm may introduce certain security features and policy parameters on the object across its realm boundary. Additional functionalities and policies may be added as the object moves up the realm hierarchy. Finally, at the root realm, the object is abstracted as a set of abstract capability and policy parameters implemented at the different realm hierarchy boundaries.

An example would make the discussion clearer. A data center represents a pool of compute resources owned by an enterprise. At the lowest level (level 0) of the hierarchy, this resource pool is divided amongst multiple controlling realms. These realms are responsible for allocating resources to objects from their pool such that they satisfy certain abstract capability parameters. We call these *basic objects*. The level-0 realm hides the mechanisms it uses to allocate resources from the level-1 realm. At the next level of the hierarchy (level 1), these basic objects may be enhanced with certain fault tolerance properties. We call this *level-1 enhanced object*. Thus, the level 1 realm is responsible for ensuring these fault tolerance properties, transparently, to the higher level realm. This could be implemented in different ways such as reassigning a different basic object to the level-1 enhanced object, leasing a basic object from another level-1 realm, etc. The level-1 realm enhanced object is presented as a level-1 basic object to the next higher level, level-2 realm. In the level-2 realm, the level-1 basic object could be enhanced with mobility guarantees. This could mean that the level-2 realm undertakes the responsibility to ensure that the computation state of the object at any given instance could be transferred to some other object (with different state of attributes) within the same root realm or outside the root realm. Similarly, at each level in the realm hierarchy new services could be defined to enhance the capabilities of the object. However, each realm needs to incorporate some basic security and energy-efficiency mechanisms into its service creation. Also, objects may advertize special security and energy efficiency parameters as explicit capabilities. For example, objects with special security features could be used as part of a banking application context. Similarly energy-efficient objects with comparable performance guarantees are more desirable considering cost and environmental issues.

It must be noted that the idea of hierarchical management and administration is not new. In-fact, it is the standard systems practice for handling scalability and managing complexity. Our effort is to provide the

formal constructs and the basic platform such that these mechanisms can be incorporated into the architecture. Simply expressed, we try to ensure that the hierarchical realm framework and the object abstraction lie at the base of every communication context, allowing specific contexts to define their own implementations.

III. The Object Abstraction

The "object abstraction" is the basis of the multi-tier diversification architectural framework of Internet 3.0. In this section we define some of the basic terms associated with the "object abstraction". At this point, a minimal set of formal constructs are presented to model the basic principles of object composition.

In all formalisms presented in this proposal, the following symbols are defined as:

$\cup$: Set operator U nion; $\cap$: set operator Intersecti on;

- : Operator between 2 set operands representi ng the set Difference operation; η : set cardinal number

3.1 Object Abstraction: Building Blocks

3.1.1 Entities: The formal definition of "entity" (Section 2.A) is given by (1).

Formal Representation:

$$\text{Entity } E = \{e\}$$
$$e: \text{Entity tier number; } e = 1 \text{ for Infrastructure tier, } e = 2 \text{ for Host tier, } e = 3 \text{ for User/Data tier} \quad \textbf{(1)}$$

3.1.2 Realms: The formal definition of realms (Section 2.B, 2.C) is given by (2).

Formal Representation:

$$\text{Realm } R = < \text{rid}, \{\text{rid}_p\}, \{\text{pol}_h\}, \{\text{pol}_y\}, \{\text{pol}_h^p\}, \{\text{pol}_v^p\} >; \text{ where,} \quad \textbf{(2)}$$

rid : realm ID; $\{\text{rid}_p\}$: set of parent realms; $\text{rid}_p = \varphi$ indicates "no parent"

$\{\text{pol}\}$: set of policies enforced by realm. Peering conditions or restrictions owing to security, social, political or other such reasons, service extension conditions, pricing, etc are some examples of policies.

$\{\text{pol}^p\}$: set of policies inherited from realm's parent

$\text{pol}^p = \varphi$, represents no parent policies inherited.

3.1.3 Objects: An object is a logical instantiation of an entity, in a specific networking context. Objects encapsulate the complexities of resource allocation, resource sharing, policy enforcements etc. and expose a standard interface representing capabilities (in terms of standardized abstract parameters) and fixed or negotiable policies. Objects are owned and managed by realms and represent the responsibilities and policies pertaining to its realm membership.

Formal Re-presentation:

1. Object Identity $\text{OID} = < \text{id}, r \in R >$; id : " id" of object wit hin realm ' r' **(3)**
2. Object $o \in O = < \{\text{oid} \in \text{OID}\}, \{\text{cap}\}, \{\text{pol}_h\}, \{\text{pol}_v\}\, e_n \in E >$ **(4)**

$\{\text{cap}\}$: set of capabiliti es advertised by the object

$\{\text{pol}_h\}, \{\text{pol}_v\}$: set of object policies defined for horizontal and vertical composites , respective ly

$\varphi \in O; \quad \eta(\varphi.\text{id}) = 0$

Discussion

3. Object Placeholde r (OP) defines a space for placeholde r objects. Placeholde r objects $op \in OP$ has a one-to-one correspond ence with the objects in the object space O. Unlike, objects in the object space O. placeholde r objects are not actual instantiat ion of objects **(5)**

Object policies may represent object requirements specific to their entity levels. As an example, "connection" for infrastructure objects refers to actual physical connections whereas in host objects or user/data objects "connection" refers to logical connections.

3.1.4 Object Capabilities: Object capabilities, called *"Singular capabilities,"* represent the set of capabilities exposed by an object. Each object capability needs to be a parameter-value pair with "hints for capability negotiation". These hints are meant to support interoperability in the capability definition

language and also the maximal allowance for deviation from the mean value stated in the parameter-value list.

Formal Representation: $\text{Capability cap} = < \text{Parameter}, \text{Value}, \{\text{negotiation hints}\} >$ **(6)**

3.1.5 Map: A map may be considered to be a type of dynamic "workflow" [ROS05][SHA05][DAN07] [HOH06][YU05]. It represents requirement abstractions that drive object composition. It is a set of requirement specifications defining a particular "requirement specific" networking context. The map presents different levels of abstraction, with different sets of parameters at each entity level and moves top-down through the different entity levels.

The requirements are specified as "local requirements" and "end-to-end (e2e) requirements." A local requirement relates to parameters that can be satisfied by individual objects while end-to-end requirements are spawned when the individual objects are composed into groups. The map initially starts off with a few local and end-to-end requirement parameters defined over placeholder objects (see definition [5]) at the application specification level. This highly abstract specification mostly provides a top level description of the desired networking context. The context is refined and the abstract service parameters instantiated with actual object capabilities as the map moves downwards.

The key idea is to map the top level context specific requirements into discrete individual object capabilities. The requirements are prioritized at each level. Thus, at each step of the map's descent, it initiates a horizontal composition of objects (described next in Section 4) that satisfy a subset of the map's local requirements. The composition also spawns a new set of end-to-end requirements. The end-to-end requirements of objects belonging to the same entity level are recursively satisfied by factoring an end-to-end requirement as local requirements and initiating a horizontal composition. The horizontal composites are stacked downwards along a descending priority order. Finally, when all the local and end-to-end requirements are satisfied, the composite represents the requirement specific context that was set to be defined.

Formal Representation:

$\text{Map m} = < \text{map\_id}, \text{app\_id}, e_i, \{\text{local\_req\_specs}\}, \{\text{e2e\_req\_specs}\} >$ **(7)**

map_id: map identifier local to the context of an application

app_id: identity of top level application that has created the map

e_i: instantiation of the map at a particular entity level

{local_req_specs}: set of singular requirement specifications in terms of parameters of level e_i

{e2e_req_specs}: set of end-to-end requirement specifications in terms of parameters of level e_i

$\text{local\_req\_specs} = < \text{req\_id}, \text{specification}, \text{op} \in \text{OP} >$ **(8)** (see **(5)** for definition of OP)

req_id: id of the requirement local to the map

oid: object id

specification: construct representing the requirement

$\text{e2e\_req\_specs} = < \text{req\_id}, \text{op}_1 \in \text{OP}, \text{op}_2 \in \text{OP}, \text{specification}, e_i, \text{priority} >$ **(9)**

req_id: id of the requirement local to the map

$\text{oid}_1, \text{oid}_2$: object ids of objects in the end-to-end group. Composites with more than one object are factored into ${}^{n}C_2$ binary end-to-end groups

specification: construct representing the requirement

e_i: entity level at which the end-to-end specification is to be realized

Priority: End-to-end specifications are expanded to a new map consisting of a new set of singular and group specifications. Within each entity level, when multiple such group specifications are made, the priority determines sequence of this expansion. Thus, (e_i, priority) dictate the order of top-down vertical composition (discussed in Section 4)

Functions on Maps:

1. *Translate:* Replaces the placeholder objects in local and end-to-end requirements with actual instantiated objects.

$$\text{Translate } \chi(m) = m' \quad \textbf{(10)}$$

$\forall op_i$, m.local_req_specs.$op_i \in$ OP is replaced by m'.e2e_req_specs.$o_i \in O$

$\forall (op_i, op_j)$, m.e2e_req_specs.$op_i \in$ OP, m.e2e_req_specs.$op_j \in$ OP is replaced by m'.e2e_req_specs.$o_i \in O$, m'.e2e_req_specs.$o_j \in O$, respectively

2. *Prune:* Prunes a translated map to get rid of the local requirements that have been satisfied by the object compositions in that level.

$$\text{Prune } \gamma(m') = m'' \quad \textbf{(11)}$$

$m''.\ local\_req\_specs = \varphi$ (Assumption: At each level of object composition, all the local requirements for that level are satisfied)

$m''.\ e2e\_req\_specs = m'.\ e2e\_req\_specs$

3. *Remap:* Draws a new map from translated and pruned map. The group of end-to-end requirement specifications of the pruned map, having the highest priority is remapped to spawn new <local_req_specs, e2e_req_specs> in terms of objects $op_i, \in OP$, that satisfy the end-to-end service required for these objects. The end-to-end requirements not part of this group are redefined in terms of this remapping.

$$\text{Remap } \rho(m'') = m''' \quad \textbf{(12)}$$

$\forall\ m''.e2e\_req\_specs$ belonging to highest priority group

If, (m''.e2e_req_specs.$o_i \in O$, m''.e2e_req_specs.$o_i \in O$) spawns

- 'n' local requirement specifications in terms of objects $(op_1 \in OP, .. op_n \in OP)$
- $\lceil {}^{n}C_2 + 2 \rceil$, end-to-end requirement specifications $\{(o_i \in O, op_1 \in OP),$ $(op_1 \in OP, op_2 \in OP) \ldots (op_n \in OP, o_j \in O)\}$

For all other m''.e2e_req_specs defined over $(o_i \in O, o_j \in O)$ are redefined in terms of $\lceil {}^{n}C_2 + 2 \rceil$, end-to-end requirement specifications $\{(o_i \in O, op_1 \in OP), (op_1 \in OP, op_2 \in OP) \ldots (op_n \in OP, o_j \in O)\}$

Discussion: Initially to start with, single and group requirements are specified in terms of application level place-holder objects belonging to the placeholder object space. At the first level of object composition, when the place-holder object's local requirements are instantiated over actual objects belonging to the object space, the map is translated. After this the map is pruned and then remapped to guide the next level of object composition. Thus, the process of object composition involves multiple cycles of *(translate, prune, remap)* of the original map till 'prune' returns a map with empty local and end-to-end requirement specification sets.

Figure 2, is an example of the map rendering procedure along the different service levels, discussed thus far. The highest level application layer represents requirements between a data source and the data sink. An intermediate set of data processing objects processes the data produced by the data source. Between the first two data processing host objects, a delay tolerant service needs to be interposed that provides capability to store the data till a forwarding link is available. Finally all these objects map to infrastructure objects for actual transmission. The interposed host entity level services introduce packet transmission delays between the data-source and data-sink, thus requiring the "transit objects" over the infrastructure to vary in their capabilities. Also, as shown in the figure, at each downward step, the local requirements are instantiated on real objects and the end-to-end requirements are rendered into multiple placeholder objects.

IV OBJECT ABSTRACTION: PRINCIPLES OF OBJECT COMPOSITION

In this section, we define some of the underlying principles governing object compositions. The objects defined in Section 3.1 are called "Simple Objects" to distinguish them from "Composite Objects". Composite objects are formed by the aggregation of more than one simple object. However, the object

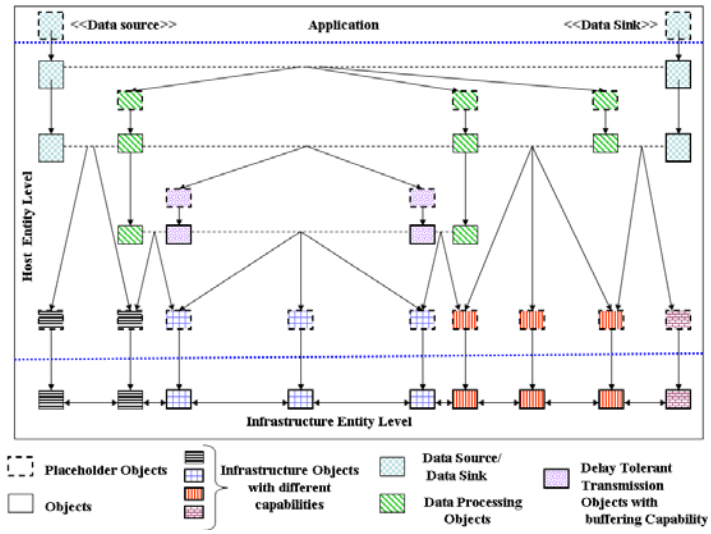

Figure 2. Example Scenario of "Rendering the Map"

abstraction does not treat composite objects any different from simple objects. Composite objects expose similar interfaces as that of simple objects. Thus, once successfully composed, composite objects abstract composite functionality of multiple simple objects. *However, composite objects are distinguished from simple objects : η{oid Є OID} > 1 for composite objects, η{oid Є OID} = 1 for simple objects.*

Policy negotiations are not simple and may require lengthy representations. Specific functions for specific types of policies may need to be defined. Also, success and failure of a policy negotiation is dependent on specific context of the policy and the defined success and failure conditions. However, in all formalisms presented in this paper, for the sake of simplicity, all policy negotiation functions have been simplified using the **'τ'** operator returning a binary result. Thus,

$\tau(pol_i , pol_j) = 0$; even if a single negotiation fails.
$\tau(pol_i , pol_j) = 1$; indicate a successful negotiation.

Principle #1: *Objects represent singular capabilities. These singular capabilities can only satisfy "local requirement specifications" in the map. Thus, "end-to-end requirement specifications" of a map has to be factored into singular object capabilities that can be satisfied by object composition. Such composition shall spawn new end-to-end requirements and re-definition of existing end-to-end requirements.*

Example 1: Suppose G_1 = {Object a, Object b, Service S_1, $e_i = 3$, Priority = 1} and G_2 ={Object a, Object b, Service S_2, $e_i = 3$, Priority = 2} represent two group requirements between objects 'a' and 'b'.
Since, Service S_1 has a higher priority (lower the better metric), S_1 has to be factored into $\{s_1 \ldots s_n\}$ singular object capabilities. These singular capabilities are instantiated on objects $(o_1 .. o_n)$ spawning additional end-to-end (e2e) requirements potentially between each pair of $(o_1 .. o_n)$. Also, any lower priority group requirement between the same objects, such as G_2, has to be redefined to (G_2 (a, o_1), G_2 (o_1, o_2) … G_2 (o_n, b)).

Principle #2: *Derived from Principle #1, if re-defining an end-to-end requirement owing to higher priority end-to-end requirement initiated refactoring, causes the original end-to-end service to be rendered impossible, then the map is considered invalid.*

Example 2: As presented in Example 1, suppose G_2 represents an e2e requirement between 'a' and 'b' which fixes the maximum transmission delay to 10 ms. And, G_1 initiates a refactoring for a store and forward service between 'a' and 'b' consisting of 5 intermediate store-forward hops, each storing a packet for at-least 2 ms before forwarding it. Such a scenario shall cause rendering G_2 impossible, thus invalidating the map.

Principle #3: *Strict top-down ordering of object composition: The object composition is guided by the top-down movement of the map and hence enforces a strict top-down order in object composition.*

Principle #4: *Horizontal Composite: Simple/Composite objects belonging to the same entity level may be composed together to form a horizontal composite object, if, 1) each object satisfies at-least one singular requirement specification of the map through its set of capabilities, 2) their realm policies can be negotiated, and 3) their object policies for horizontal composite formation can be negotiated.*

The resultant horizontal composite object instantiates new end-to-end requirement specifications on the object pair of the horizontal composite and exposes a single unified interface for policies, capabilities aggregated over its constituent objects.

Formal Representation: Horizontal composite aggregation operator $(\Lambda_{(m \in map)})$ under map m: returns an aggregate object formed by an horizontal composite operation on the operand objects and an updated map instance generated from the input map and updated to reflect the results of object composition.

$$\Lambda_{m\,(e_i \in E,\, priority)}\,(o_i \in O, o_j \in O) = o_k \quad \text{if} \tag{13}$$

1. $o_j.e_j == o_i.e_i$; 2. $\tau((oid_j.r.pol \cup oid_j.r.pol^P), (oid_i.r.pol \cup oid_i.r.pol^P)) == 1;$

3. $\tau(pol_i^h, pol_j^h) == 1;$

Definition: 4.1 $(m_{(e_i \in E,\, priority)}.local\_req\_specs) \cap (cap_i \cup cap_j) \geq 1;$

if, $\eta(oid_i.id) > 1$ or $\eta(oid_j.id) > 1$; or, $\eta(oid_i.id) == 0$ or $\eta(oid_j.id) == 0;$

4.2 $(m_{(e_i \in E,\, priority)}.local\_req\_specs) \cap (cap_i \cup cap_j) \geq 2$; if $\eta(oid_i.id) == 1$ and $\eta(oid_j.id) == 1$

o_k = $< oid_k = oid_i \cup oid_j$, $cap_k = cap_i \cup cap_j$, $pol_k^h = pol_i^h \cup pol_k^y$, $pol_k^y = pol_i^y \cup pol_j^y$, $oid_k.r.pol = (oid_j.r.pol) \cup (oid_i.r.pol)$, $e_k = e_i = e_j$ >	$m_{(e_i \in E,\, priority)} = m'_{(e_i \in E,\, priority+1)}$ or $m'_{(e_i - 1 \in E,\, priority)}$ depending on the entity level of the constituent objects in map such that $m'(e_i \in E, priority+1)$ or $m'(e_i - 1 \in E, priority) = \rho(\gamma(\chi(m_{(e_i \in E,\, priority)})))$ from definitions 10, 11, 12

Definition: *Composing object 'o_i' with null object 'φ'*

$$\Lambda_{m\,(e_i \in E,\, priority)}\,(o_i \in O, \varphi \in O) = o_k = \Lambda_{m\,(e_i \in E,\, priority)}\,(\varphi \in O, o_i \in O) \tag{14}$$

If $(m_{(e_i \in E,\, priority)}.local\_req\_specs) \cap (cap_i) \geq 1$

And, $m_{(e_i \in E,\, priority)} == m'_{(e_i \in E,\, priority+1)}$ or $m'_{(e_i - 1 \in E,\, priority)}$;

depending on entity level of the constituent objects in the map

such that, $m'_{(e_i \in E,\, priority+1)}$ or $m'_{(e_1 - 1 \in E,\, priority)} == \rho(\gamma(\chi(m_{(e_i \in E,\, priority)})))$ from definitions 10, 11, 12

Note: The remap function 'ρ' has been defined in (Section 2.1.5) such that it can handle e2e_req_specs for object pairs of the form (o_i Є O, o_j Є OP) and spawn a remap of e2e_req_specs in terms of local_req_specs if and when (o_i Є O, op_j Є OP) → (o_i Є O, o_j Є O). Hence,

Definition: *Extensions of definition 14*

$$\Lambda_{m(e_i \in E,\, priority)}\left(\Lambda_{m(e_i \in E,\, priority)}\,(o_i, \varphi), \Lambda_{m(e_i \in E,\, priority)}\,(o_j, \varphi)\right) = \Lambda_{m(e_i \in E,\, priority)}\,(o_i, o_j) \tag{15}$$

$$\Lambda_{m(e_i \in E,\, priority)}\,(o_i, o_i) = o_i, (m'_{(e_i \in E,\, priority+1)} \text{ or } m'_{(e_i - 1 \in E,\, priority)}) \text{ (From } \mathbf{(14)}\text{)} \tag{16}$$

***Definition*:** *Multi- composite*

$$\Lambda^{*}{}_{m_{(e_i \in E,\ \text{priority})}}(o_1, o_2, o_3, \ldots, o_n) = \Lambda_{m^{*}}(\Lambda_{m^{n-1}}(o_1, \Lambda_{m^{n-2}}(o_2, \Lambda(\ldots, \Lambda_{m^{1}}(o_{n-1}, o_n)$$

$$\text{where, } m^{p}_{(e_i \in E,\ \text{priority})} == \rho\left(\gamma\left(\chi\left(m^{p-1}_{(e_i \in E,\ \text{priority})}\right)\right)\right) \quad \textbf{(17)}$$

Principle #5: *Vertical Composite: A vertical composite is formed by stacking a simple/composite object of same or lower entity level below another simple/composite object if, 1) the difference in their entity level is at-most one , 2) their realm policies can be negotiated, 3) their object policies for vertical composite formation can be negotiated.*

The downward movement of the map initiates horizontal compositions. The vertical composite is an aggregation of these horizontal composites ensuring proper policy negotiations. The simple object is a special case when a single object satisfies the local requirement specifications of a map for that level. The vertical composite formation does not affect the map in any way.

Formal Representation:

Vertical composite aggregation operator (***π***): returns composite object formed by connecting the operand objects and stacking one object below the other. It is formally defined as:

Definition: $\pi(o_i, o_j) = o_k$; if **(18)**

1. $o_i.e_i - o_j.e_j == 1$ (from fundamental constraint);
2. $\tau(\{(oid_j.r.pol) \cup (oid_j.r.pol^{P})\}, \{(oid_j.r.pol) \cup (oid_j.r.pol^{P})\}) == 1$
3. $\tau(pol_i^{y}, pol_j^{y}) == 1$

such that, $o_k = \left\langle \begin{matrix} oid_k = oid_i \cup oid_j, cap_k = cap_i \cup cap_j, pol_k^{h} = pol_i^{h} \cup pol_k^{y}, \\ pol_k^{y} = pol_i^{y} \cup pol_j^{y}, oid_k.r.pol = (oid_j.r.pol) \cup (oid_i.r.pol), e_k = e_i \end{matrix} \right\rangle$

Definition: $\pi(o_i, \varphi) = o_i$; *Similarly,* $\pi(\varphi, o_i) = o_i$ **(19)**

Vertical composite of multiple objects is defined as:

Definition: $\pi^{*}(o_1, o_2, o_3, \ldots o_n) = o_k = \pi(\pi(\pi(\ldots(\pi(\pi(o_1, o_2,), o_3,) \ldots), o_{n-2}), o_{n-1}), o_n)$ **(20)**

Principle #6: *Termination condition: The termination of a composition is indicated by the prune function on a map returning an empty set for both local and end-to-end requirement specifications.*

The elements of object abstraction and the principles stated for object composition are the basic theoretical model that defines the architecture of Internet 3.0. However, the theoretical model stated in this section is abstract, greatly simplified and formed from "crude" formalisms. An actual prototype implementation of the architecture is likely to refine the existing basic principles and also provide more insights into defining additional principles.

V ARCHITECTURE

The discussion thus far presents the theoretical basis of the Internet 3.0 architecture and its underlying principles. These theoretical constructs shall guide the implementation of the protocols that support the framework. In this section we present a general prototype implementation plan that shall allow us to evaluate the feasibility, performance and deployability of the various design choices and guide research to improve, modify, abandon or create newer constructs that pave the way for adoption into production environments.

Internet 3.0 provides a generic framework for the co-existence of multiple application contexts. We define two deployment scenarios that represent the two extremes of application deployment over Internet 3.0, also representing the classical tradeoff between granularity of control versus acceptable complexity. These two scenarios are:

SCENARIO A. **Applications Composed Over Services:** Third-party service providers compose generic services over leased objects and expose capability through extremely simple and generic service interfaces. This is called a "**service context.**" Applications can compose their specific contexts over these

service interfaces and are thus shielded from the complexities of creating and maintaining the required service levels by themselves. The CABO [FEA07] project is based on similar ideas of third party service provisioning in the infrastructure tier. We extend it to the host tier as well. Also, in Internet 3.0, as we shall see in the next usage scenario, applications do have the options of creating specific contexts tailored to suit their specific requirements directly over objects. Applications that are expected to leverage this usage scenario include distributed scientific collaborative applications, content delivery networks, Web applications, cloud enterprise applications, etc. A simple example scenario would be a requirement of the Genome Center to transfer 100 TB of data to distributed cloud platforms for processing. Once the processing is complete, the data would again need to be delivered back to the Genome Center for analysis. The current Internet provides no means to provision resources across multiple providers to dynamically enable such huge data transfers to multiple random locations. At the same time, even with the object-oriented concepts of composing application contexts dynamically in Internet 3.0, it would be overtly complex and unreasonable to expect the Genome Center to be able to create this short lived context bottom up at the granularity of object compositions. Similarly, short lived web transactions requiring certain levels of quality of service can in no way justify the effort and time required to compose its context from objects. Thus, the Internet 3.0 protocol suite provides a set of extremely simple and standardized interfaces and secure protocols, allowing applications to design their contexts over pre-composed services.

SCENARIO B. **Applications Composed Over Objects:** The second usage scenario pertains to applications such as third party service provisioning and distributed applications with specific needs. Third-party services are examples of Scenario B that can dynamically provision their service offerings based on requirements, thus making them adaptive and hence profitable. These applications compose their specific contexts at the granularity of objects and have to bear the complexities associated therein. Thus, complexity is the price to be paid for higher granularity of control. Such high granularity of control is essential for many application contexts that need to dynamically adapt to topology and requirement changes, fault situations, optimizations to ensure profitability, etc. As already discussed third party services are examples of such contexts. Other contexts that may require such high levels of control include distributed gaming applications, virtual space applications, etc. The Internet 3.0 framework provides protocols and object interfaces for object compositions, object advertisements, object brokering and leasing, accounting, monitoring etc, to facilitate creations of such services and applications. Albeit, the map primitive and the map rendering process pertains to this more general usage scenario of which the first usage scenario is just a special case. Also, it can be observed that the present Internet's world-wide web application context can be modeled as the first usage scenario where the ASs contribute connectivity objects to setup an end-to-end connectivity service that applications can access through the routing sub-system.

5.1 The Base Architecture: The base architecture consists of a two-tier object management plane over infrastructure and host object realms. The management plane implements protocols for object realms to advertize objects, lease objects to application contexts, monitor objects for compliance with advertized capability, secure accounting of object leases to applications and other such object management functions. Each realm implements an object–level control and management plane. An object-level management plane involves interaction between the object's realm manager and the actual resources that map the logical capabilities advertised by the object. The management at this level derives many similarities with distributed resource management mechanisms involving resource naming, allocation, sharing, accreditation, and reclamation. Also, security to ensure the integrity of the underlying resources and admission control for honoring QoS are important management functions at this level.

Unlike the current Internet design, wherein management functions operate over paths setup through distributed routing protocols in the control plane, the management plane of Internet 3.0 is the ***"bootstrapping plane."*** The design of the Internet 3.0 architecture dictates this requirement on the management plane which aids the self-configurability of individual objects and the publish/lease framework for service compositions. Thus, the initial requirement is to setup a self-configured

management path between the realm managers and their specific resources that instantiate their objects. This management path is used for distributed control and management of the resources within the realm.

A. Infrastructure Tier

Routing domains or autonomous systems of the current Internet represent infrastructure realms in the Internet 3.0 architecture. Central to the architecture of Internet 3.0 is the requirement that these infrastructure realms provision objects over their distributed resources. Examples of "infrastructure objects" in the current Internet are BGP "connectivity" objects advertized by ASes. These connectivity objects also implicitly represent the "export" policies of the AS that determines the ASs decision to provide connectivity between the advertized points based on some underlying commercial arrangement.

The Internet 3.0 architecture allows these infrastructure realms to be able to provision more diverse set of capabilities and express a richer set of policies though the object-level management plane. Distributed resource allocation and sharing within routing domains has been extensively researched. Different infrastructure owners may employ different methods such as virtualization of its resources[AND05][FEA07][TUR107][ON408][ON431][GENI101][GENI102][ON802], MPLS based fixed circuits [ROS01], flow identification and classification[KN98][BR94], etc. combined with a variety of queuing, buffering, scheduling disciplines that best suit their resource capability and organization. Internet 3.0 overlays these mechanisms with a management plane implementation that encapsulates capabilities and policies within a common standardized interface while leaving the specific mechanisms for implementation of these capabilities and policies to the AS.

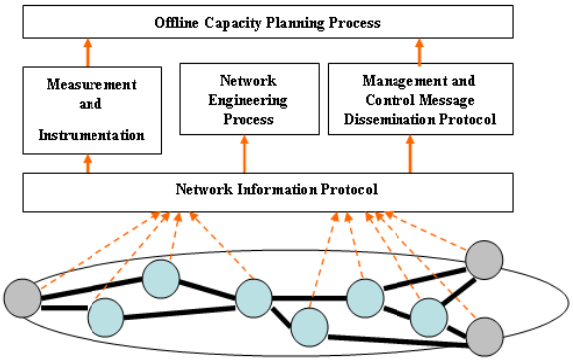

Figure 3a. Infrastructure Object Management Plane: Decision Process Based on Network State Information

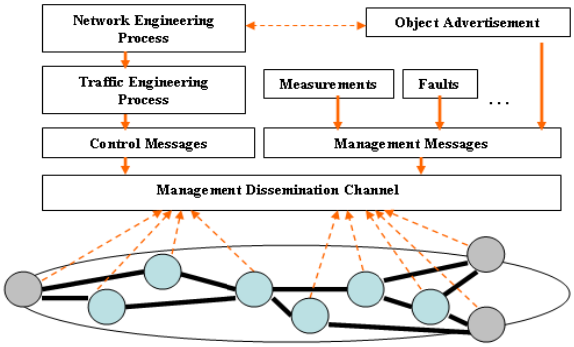

Figure 3b. Infrastructure Object Management Plane: Management Decision Dissemination Process

Figure 3a shows a generic set of elements of the management plane decision process. A Network Information Protocol (NIP) lies at the base of the decision process and collects the necessary resource state information. This information is fed into different modules such as the measurement module, the network engineering module and the management and control message dissemination protocol module. The measurement module ensures that the measured performance of the realm complies with the planned performance. The network engineering module is responsible for managing the capacity use of the installed resource base for optimal performance, utilization, and profitability. The management and control message dissemination protocol module is responsible for maintaining a full map of the connectivity state of the network elements and maintaining a dissemination channel that provides management plane connectivity of all the resources in the system. The measurement and traffic engineering modules log their information into the offline capacity planning module that allows the realm owner to make informed decisions regarding long term capacity planning of the system.

Figure 3b shows the management decision dissemination process that allows the different distributed management and control modules to communicate between themselves and also with the resource elements. The dissemination channel is maintained by the management and control dissemination

protocol module (Figure 3a) and provides the required management plane connectivity. The network engineering process plans the capacity allocation of the realm resources according the present demands. It creates objects over its available resources and advertizes it with the object advertisement module. When an object is leased, the network engineering process is responsible for allocating resources to the object to ensure compliance with performance as well as maximize profitability. The network engineering module implements the results of its planning process through engineering the existing usage of the resources. The traffic engineering module is the tool used by the network engineering module to implement its decisions. The traffic engineering module in turn communicates with the control messaging module to setup the necessary distributed state over the resource elements to implement the traffic engineering decisions. Similarly, the dissemination channel is used by other management and control plane modules such as fault management modules, measurement modules, etc to effectively manage the distributed resource allocation and other state variables of the realm.

It must be noted that Figure 3a and Figure 3b is just a generic representation of the organization of the different management modules that constitute the infrastructure object management realm. The exact mechanisms for use within the different modules shall vary across different realms and shall be transparent outside the realm. The interface of the capabilities and policies of the realm is abstracted through objects. As already discussed, the present best-effort routing in the internet is already based on the concept of simple connectivity objects. However, the scenario in Internet 3.0 is much more complex. Internet 3.0 allows objects to be leased by anybody who cares to use it for a certain price. Usage policy is not based on offline commercial arrangements between ASs. Thus, a dynamic framework for object advertisement and lease is required. The "Context Router" is the central component for the object advertisement and leasing framework and allows application specific contexts to be setup in the infrastructure tier.

A.1 Context Router: The context router is the central component of the object advertisement and leasing framework of Internet 3. 0. The context router is positioned at the Internet Point-Of-Presence (POPs) and allows al the AS's that participate in the POP to advertise their infrastructure objects. Figure 4 presents a high level view of the present Internet POPs enhanced with a context router that allows the current Internet infrastructure to transition to Internet 3.0 infrastructure tier.

Figure 4 presents a highly simplified POP design where each AS has a border router that connects to the POP, enhanced with the context router. The context router has two key functionalities: 1) It maintains an object repository on behalf of the participating ASs and makes them available for lease to application contexts, 2) It allows applications to lease resources on programmable router platforms within the POP (shown as the hatched and dotted contexts) and set appropriate filters to ensure the application specific flows to be forwarded through them. This allows application contexts to set-up their own packet processing contexts at POPs. There are two points that need to be noted here:

1) Application contexts **may lease** objects at the context router **without having to lease** packet forwarding resources at the POP and setup their own packet forwarding paradigm.

2) It is **necessary** for an application context to lease objects at the context router to lease packet forwarding resources at the POP. The restriction of necessarily requiring application contexts to lease objects at the context router to be able to setup their specific packet processing context at the POP

protects ASs from application contexts misusing the knowledge of AS level connectivity maps. For example, an application requiring best effort connectivity over a multicast tree topology may want to

setup its specific packet processing context at certain POPs to be efficiently able to implement the required topology. It may use the AS level connectivity map to ascertain the best POPs where it should duplicate packets. However, such a design is oblivious to the policies of the underlying ASs. The connectivity map divulges AS level connectivity to aid applications to plan their contexts over leased objects from multiple ASs. Thus, use of this map to do best-effort forwarding interferes with the policies of the ASs. ASs thus advertize "best-effort connectivity" objects at the context router that explicitly allows certain best-effort transits through the AS. Any application context that requires to setup its own

forwarding context over this best effort connectivity needs to lease these best-effort connectivity objects at the context router.

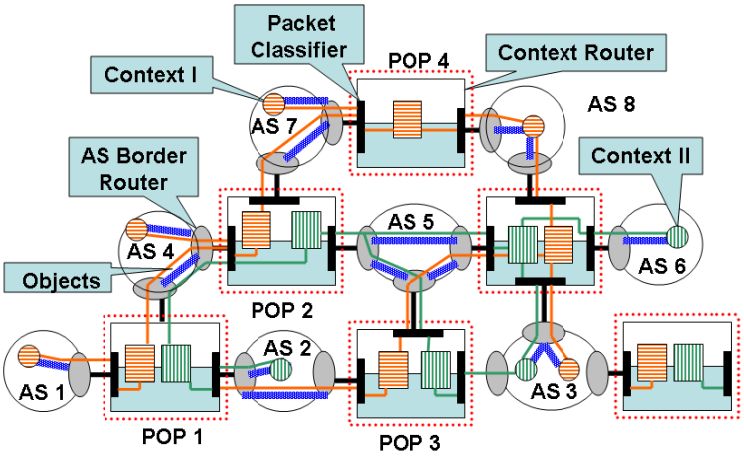

Figure 4. Internet 3.0 Infrastructure Management Plane: Context Router Enhanced POPs

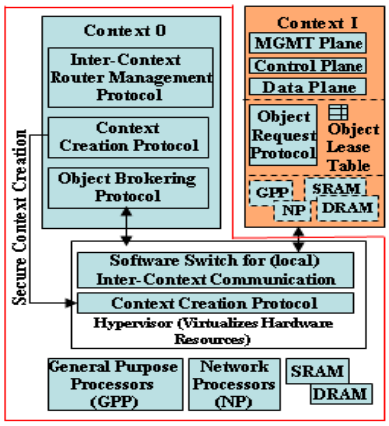

Figure 5. Context Router Design Overview

A.2 Context Router Design: Figure 5 presents an overview of the context router design. A context router needs to have multiple virtualized contexts. A hypervisor is responsible for creating and controlling the multiple contexts. Also, the hypervisor platform runs protocols that allow applications to request for context creation on virtualized hardware resources. There is a base context called the context 0 that hosts the object store. Participating AS's advertize their objects at the POP and they are stored at the context 0 of the context router. Also, the context 0 participates in the inter-infrastructure realm management plane and stores AS level connectivity maps. It runs a brokering protocol that allows application contexts to query, block, lease and release objects from the object store. A secure software switch allows inter-context communications, mostly to allow application contexts to be able to communicate with the context 0. Different application contexts may setup their own packet processing contexts over the virtualized resource allocated to them. Also, they run object request protocols and application specific data plane, management plane and control plane protocols.

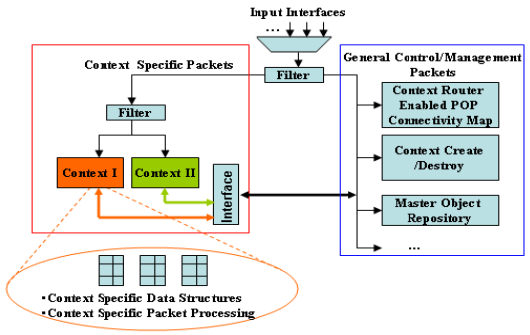

Figure 6. Context Router: Application Context Perspective

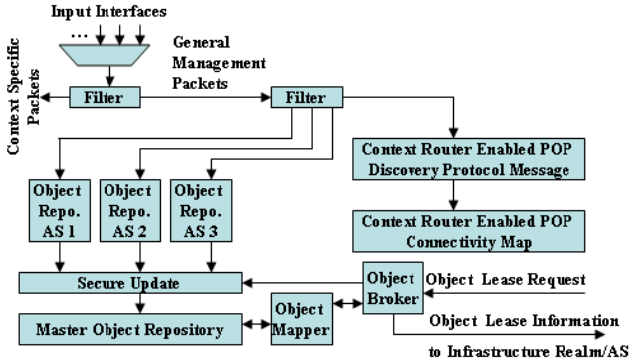

Figure 7. Context Router: Context 0 - Object Store

Figure 6 presents the view inside the content router from the perspective of the application contexts that install themselves on virtualized resources. Filters at the line card interfaces of the context router de-multiplex packets destined for the application contexts and general management and control packets responsible for maintaining the global tables such as master object repositories, management plane

connectivity for context-routers at various POPs, application contexts co-located in the POP, etc. Context specific packets are further de-multiplexed and forwarded to the correct application context. Application contexts maintain local data structures for context specific packet processing and forwarding functions. Also, application contexts are allowed secure and controlled access to the global tables through a standard interface. This interface allows the application contexts to avail of the object brokering

services, next-hop context router information and other such services. Also, the application contexts may decide to perform measurements to determine the compliance of the leased objects with the advertized capabilities. However, in cases of discrepancies, the application context needs to attest its measurement with an independent measurement function implemented in the management plane of the context router. Such attestation allows for arbitrating disputes pertaining to accountability. Most application context may delegate the responsibility of performance measurement to the generic measurement function of the context router management plane.

Figure 7 presents a closer look at the context 0 object store. Each participating AS advertizes its objects with the context router through a secure object advertisement management protocol. These advertisements are stored as part of local object repositories specific to each AS. These repositories are parsed at fixed time intervals to securely update a master repository. The object brokering function interfaces with an object mapper function that maps requirements to object capabilities, policies, price, etc. and returns a vector of object pointers into the master repository. The object broker, ether passes this information back to the application context that made the request or selects the most suitable object lease if delegated the authority to make this decision by the application context. Finally, upon receiving/making the leasing decision, the object broker updates the master repository through the secure interface and also informs the infrastructure realm manager about the object lease by the application context over the management plane.

A.3 Inter-Realm Management Plane

The infrastructure realms co-operate to a distributed, secure inter-realm management plane. The management plane maintains an inter-realm dissemination channel for control and management messages. Context routers also participate in this management plane. Application contexts make object requests to different context routers to setup an end-to-end context specific path over the management plane. Infrastructure realms and context routers participating in this management plane are given a set of routable identifiers within a management plane identifier space. Infrastructure realms and context routers can be directly addressed through these identifiers. The management plane also aids the setup of an end-to-end provisioned path between two end-points. It supports various forwarding paradigms such as unicast, multicast, broadcast, etc. A path computation message with application specific requirements is sent to the local context router. The message has a couple of bits that represent the forwarding paradigm for this management message. For instance, if the broadcast bit is set, the context router shall forward this message to all next hop context routers reachable through each of its outgoing interfaces. At every context router hop, the object broker maps the most suitable objects from its master repository that maps the requirements of the application context. These broadcast messages finally reach the final destination which consolidates all the different messages and sends back a reply to the source. The source then computes the most suitable set of objects and sends an object lease and composes messages addressed to the specific context router identifiers that advertized the selected objects. It must be noted that the context routers are operated by ownerships different from that of the ASs and thus protected against selfish interests of ASs to block the management message across certain paths.

B Host Tier

The host tier of Internet 3.0 consists of compute resources consolidated over end-user personal compute resources, private and public cloud computing and storage resources, Content Delivery Network (CDN) storage resources, server farms, grid resources, etc. The mechanisms for sharing common compute resources across multiple application contexts may vary from virtualization techniques [XEN][VMWare][VServer] achieving near perfect isolation and providing strong deterministic

performance guarantees, to traditional operating system based resource allocations based on global optimization and fairness considerations. Similar to the infrastructure realm, Internet 3.0 allows complete autonomy to host realms to choose the specific mechanisms for allocation of compute resources to application contexts. Also, it provides a common object abstraction interface that allows host resources to be shared across multiple ownerships over a policy negotiation plane. However, unlike the infrastructure realm which was marked by a physical realm boundary, host realms could have physical as well as logical boundaries. Enterprise networks, cloud platforms, etc represent host realms that have a physical boundary whereas end-user personal realms, virtual enterprises, etc. represent host realms that do not have a physical boundary. The significance of this fact is that host tier object composition is not dictated by a physical connectivity pre-condition, as in the case of the infrastructure tier. In the underlying commercial fabric, this allows host realms to be completely independent of each other and add value to their objects to distinguish themselves from other host realms. Infrastructure realms on the contrary need to dwell on a co-operative competitive behavior where object offering by one infrastructure realm needs to be matched by its neighbors and so on to be leased to provide an end-to-end service. However, Internet 3.0 exposes enough underlying connectivity diversity to greatly mitigate the connectivity dependence among different infrastructure realms. Also, the scale-free nature of the Internet connectivity graph, together with the incentive in Internet 3.0 architecture for transit infrastructure realms to participate in more neighbor associations is expected to reduce the significance of the restriction imposed by the connectivity pre-condition.

Internet 3.0 host objects are identified through a <Host Realm ID(HRID), Host Object ID(HOID)>. These objects are advertized over a common host tier management plane through a mechanism similar to the infrastructure realm. However, as already discussed, there is no connectivity dependency among the host realms allowing a much larger set of selections for each application context requirement. Also, the equivalent of the context router in the host realm does not need to be an SPP-like platform (Supercharged PlanetLab Platform [TUR207]) that allows application contexts to co-locate themselves. The host object store is implemented over a standard management protocol through which the host realms advertize their objects. The host-object store may be distributed for scalability and the same objects may be advertized across different stores. This necessitates the requirement of synchronization protocols that synchronizes object availability across distributed object repositories. This is implemented through a hierarchical organization of the object repository with the authoritative repository (one that is maintained by the host realm itself) synchronizing the object lease with other repositories that advertize its objects. Also, object lease in the host tier, is implemented over an authentication protocol that allows the object lessee to prove its identity and also its authorization to use the object.

C Data Tier

The data tier represents data and user objects. Users and data are managed by user and data realms. The present Internet too has the concept of data and user realms. Currently Internet data may be classified to belong to three realm types. The first type is the “Null Realm.” “:Null Realm” data does not have any record of its creator or its ownership. It can be freely distributed and used. Also, there is no guarantee of data integrity. Malicious behavior of “Null Realm” data cannot be attributed to any specific accountable entity. This is another caveat in the present Internets security model wherein malicious null realm data is associated with the host that served it. Clearly this is a faulty attribution, and one that needs to be modified. The second type of realms are “Loose Ownership Realms.” “Loose Ownership Realm” data have loose ownership attributes. Generally the ownership realm ensures integrity of the data through encapsulated meta-data. However, the guarantees of data integrity do not automatically imply that it can be attributed to a specific ownership, or that the data is not malicious. The reason being that “Loose Ownership Realms” do not participate in any trust associations wherein their identity and authenticity may be attested through a trust group or a third party trust anchor. The third type of data realms are “Strong Ownership Realms.” “Strong Ownership Realms” encapsulate data within layers of authentication, authorization and accounting mechanisms. “Strong Ownership Realms” generally participate in trust associations that validate their identity and authenticity. Data from these realms may be

attributed to particular ownerships that take responsibility for it. The reason for the existence of these different realm types is owing to the lack of a basic underlying security framework for the Internet. Security mechanisms have been overlaid over the current Internet architecture rather than being developed base-up. Internet 3.0 seeks to integrate security into the basic architectural framework. We recognize the fact that security comes for a price and may not always be required. In many contexts it may be an undesirable tradeoff for performance. Thus, Internet 3.0 realizes security as a policy that is specific (and different) for each separate application or communication context. Internet 3.0 just allows the framework for explicitly expressing and negotiating security policies. It allows application contexts to decide and enforce the level of security that they deem necessary for their specific use.

Similarly, users in the current Internet may be classified into three types of realms. The first type is the "Null Realm," where the user is not accountable and its identity not authenticated. The second type is the "Self Realm." This type of realm allows the user to authenticate itself to another user/data realm with which it has setup a prior association. The third type of realm is the "Enterprise Realm." The "Enterprise Realm" associates the user to belong to a particular enterprise and allows the privileges and rights that it is authorized to enjoy as part of the enterprise. In this type of realm, the user does not need to have any prior association with the user/data realm that it is communicating with. The "Enterprise Realm" sets up these prior associations that every user may avail of. Also, the "Enterprise Realm" generally participates in trust associations with other "Enterprise" or "Strong Ownership" data realms. "Enterprise Realm" belongingness allows the user to associate certain guarantees about its identity. However, here lies another caveat in the present Internets security model. User membership to an enterprise realm is attested through his location within the enterprise network. This leads to unnatural and inefficient, externally patched authorization and authentication mechanisms such as virtual private networks, network proxies, etc.

VI RELATED WORK

Internet 3.0 is an overarching architecture. Unlike, most other architectures which solve specific problems, Internet 3.0 aims at re-defining the basic underlying primitives. A comprehensive survey of world-wide recent efforts in next generation Internet design is presented in [PAU09].Also, Internet 3.0 is a *communication paradigm* based architecture rather than a *communication system* based architecture (*see discussion in Section 2*). It is a mammoth undertaking and the natural motivation is towards being able to re-use as much of prior research experiences in more specific areas as possible. Some prior research in specific areas of networking that are relevant to the Internet 3.0 architectural framework are discussed here.

A. Programmable Networks, Active Networks, OpenSIG: Past research on "*programmable networks*," represented by *Active Networks* and *OpenSig* failed to make the desired impact despite extremely radical and revolutionary (to be interpreted in the positive sense) ideas. They explored mechanisms through which the underlying network infrastructure could be programmed to serve the specific packet processing requirements of multiple application contexts. However, they failed to address the *misalignment with the basic policy framework of the multi-ownership infrastructure of the commercial Internet.* The infrastructure owners are extremely reluctant to surrender control over their networks to arbitrary behavior of a specific application context. Also, they could not satisfactorily resolve the security issues involved with such architectures. Internet 3.0 handles both these issues as a requirement within its basic design. It acknowledges the autonomy of resource owners over the management of their resources and designs around it to provide comparable degrees of diversity.

B. Distributed Object Technologies: Common Object Request Broker Architecture (CORBA) [CORBA93][NIC93][OZC94][SOL95][BOO91][UNO95][SOL95][OMG94a] was designed to be a distributed object technology that allowed interoperability across services implemented across different operating platforms and programming languages. Similarly, our "Object Abstraction" concept is designed to achieve interoperability across different policies and resource management implementations across different ownership (root realm) or administrative (realm hierarchy) boundaries. We borrow heavily from CORBA elements of an Interface Definition Language (IDL) and the object brokering architecture.

However, despite being developed by one of the world's largest software consortium (Object Management Group), CORBA did not reach its desired levels of success. The reason for CORBA's apparent failure is attributed to its extremely complex and (sometimes) ill-defined interface design and complicated standards in an attempt to provide an extremely flexible design platform. A large portion of the industry adopted component technologies like Microsoft's DCOM [COM] instead, nonetheless at the price of loss of flexibility. The Internet 3.0 is faced with a similar challenge. The higher granularity of control and explicit representation and negotiation of policies come at the price of increased complexity. This issue of dealing with complexity is of utmost importance for the success of any successful architecture. The biggest proof to this fact is the Internet itself. The current Internets success can be largely attributed to its simple design. Thus, keeping in mind the obvious tradeoff between complexity and flexibility, Internet 3.0 introduces two application composition scenarios, Scenario A and Scenario B (*see Section V*). Scenario A and Scenario B represent the two extremes of the complexity versus flexibility tradeoff. Also, Internet 3.0 expects evolutionary market forces to undertake "natural selection" in defining a finite number of policy profiles that suitably represent most policy contexts. Newer and more complex policies may be defined as special cases. The nature of Internet 3.0 design allows such specific cases to co-exist within the generic framework as an "isolated" specific context.

C. Capability Publish Language (CPL) and Policy Publish Language(PPL): Object capabilities need to be published in terms of standardized parameters through a Capability Publish Language. Similarly, policies need to be published through a Policy Publish Language (PPL). Object capabilities represent functionalities. These functionalities are published in terms of "performance parameters". These "performance parameters" are aggregated abstraction of more specific "functional parameters" of the resources. As an example, "delay" in an infrastructure object represents a performance parameter that abstracts a set of functional parameters of queuing discipline, buffering mechanism, scheduling algorithm, etc. The same performance parameter may be translated to different sets of functional parameters, depending on the resource context. We intend to borrow and extend from prior research in multi-level QoS mapping [KOL02][LI99][JIN04][KLA00][DAS00][HUA97].

There has been a lot of previous research on developing policy languages to represent security policies in distributed systems that could be mapped into access control parameters of firewalls, databases, operating systems, etc.[KAG02][DAM01][SLO94a][LUP98][SLO99][AO92][SLO94b]. Also, distributed management policy frameworks facilitating dynamically changing policies have also been designed [MAR96]. We shall extend these policy languages and their related framework that are mostly concerned with enterprise security and access policies to include policies that reflect the trade and usage of resources in a commercial paradigm within a framework supporting active negotiation of policies. Also, the policy framework shall be interfaced with the monitoring framework to ascertain compliance.

D. *Resource allocation and sharing:* The allocation and sharing of physical resources amongst multiple logical contexts under specified performance metrics such as availability, latency, etc. is a highly explored area in operating system scheduling, real-time systems and other such disciplines. These methods may be contrasted with mechanisms of virtualization [XEN][VMWare][VServer] that partition resources and make dedicated allocations to each context.

Distributed resource allocation and sharing (for quality of service provisioning) across routing domains has been extensively researched. Different infrastructure owners may employ different methods such as virtualization of its resources [AND05][FEA07][TUR107][ON408][ON431][GENI101] [GENI102][ON802], MPLS based fixed circuits [ROS01], flow identification and classification[KN98][BR94], etc. combined with a variety of queuing, buffering, scheduling disciplines that best suit their resource capability and organization.

E. Monitoring and Management Framework: An object lease implicitly entails a Service Level Agreement (SLA) between the object owner and the object leaser. Any architectural framework that provides differentiated services and is designed for SLAs, necessarily need to be supported by a sound, efficient, robust and secure monitoring and measurement framework [CHA00]. Moreover, SLA compliance monitoring and measurement has to devise fair and unbiased evaluation methods that can

mitigate potential conflicts and can establish accountability at the required level of granularity. The two approaches generally used for SLA compliance monitoring are; 1) Passive network measurements [ZSE01], and 2) Active measurements by injecting measurement probes into the network. A third type of hybrid mechanism is also used [AID03][CIA03]. Active measurements methods to establish end-to-end performance characteristics is a well researched area[ALM99a][ALM99b][BOL93][CHO05] [COL01][DEM02][PAS01][PAX97][PAX98][YAJ99][ZHA01]. Unlike the current Internet, Internet 3.0 shall provide native support for monitoring and measuring. The idea is to standardize interfaces through which a monitoring tool can query different aspects of the network performance. Also, in the context of a tiered diversification architecture, end-to-end measurements have different meaning in the different entity levels. Objects represent SLA's themselves and hence granularity of measurements has to establish object level compliance. Additionally, since monitoring and measurement is a management plane activity, there are scalability issues for monitoring at high levels of granularity and business implications for not monitoring.

F. Policy Negotiation (Scalability and Feasibility): The policy negotiation framework for horizontal service composition requires each object realm to negotiate policies with each object realm that is already part of the composite. This leads to N(N-1)/2 policy negotiations for a N object horizontal composite. For vertical composite of two horizontal composites containing N, M objects respectively, requires N X M policy negotiations. This problem is similar to the N X N blowup in PlanetLab's initial design [PET06]. PlanetLab solved it by designating the PLC node as a central trust anchor acting as a central point for trust negotiations. The problem in our case is a bit trickier. Objects composed to form services belong to separate realm ownerships with their own set of policies. It is thus difficult to establish a centralized trust anchor. However, the semantics of object composition could include some trust aggregation policies that could represent group policy maintained at a centralized location for the composition. The semantics of group policy would vary for each group. Also, it would depend on the actual architectural instantiation to determine the central location that shall maintain and update these group policies for each composite.

G. Security: Basic security research has resulted in technologies that are important building blocks for any comprehensive security solution. Relevant technologies include digital certificates and certification authorities; PKI [BRA00]; authenticated session key exchange [BEL93]; zero-knowledge based identification [FIE87]; anonymity and pseudo-anonymity techniques [REI99][GOL99][GAB99][DUR07]; access control models and techniques, such as RBAC [BER01] [SAN99] and credential-based access control [ADA02][BER02][CAM06]; trust-negotiation systems [YU03][YU03a]; and privacy-preserving techniques, such as private information retrieval [CAN01] [DIN03][FEI01]. However, these technologies have never been investigated from the angle that is relevant to Internet 3.0, that is, the development of security tools and policies for a new Internet. We would like to emphasize that the challenge here is how to coordinate the execution of several security services in order to achieve strong security, while at the same time having flexible security solutions. Other relevant work is related to systems and tools for policy management (see the many papers in the IEEE Policy Symposium Series - www.ieee-policy.org). A main novelty of our work with respect to this past work is the introduction of new policy types and the investigation of the notion of high-assurance policy enforcement. Finally a lot of past work has addressed network security [DEN87][DUR09][MIR05][SON01]; however most of these approaches just address specific security threats, like denial of service attacks, and they do not provide systematic solutions to a secure Internet.

The discussion in this section highlighted prior research that is immediately relevant to the core design framework of Internet 3.0. However, for lack of space, we are forced to leave out from this discussion a lot of other relevant research efforts that has in some way motivated some design decisions of Internet.

VII Summary

In this paper we presented Internet 3.0, an overarching architecture for the next generation Internet. Internet 3.0 allows multiple diverse applications to dynamically create and optimize their specific contexts over resources leased from multiple ownerships. The key design primitive of the "Object Abstraction" enables functional and policy interoperability among multiple administrative domains or

resource ownerships. “Objects” are the fundamental building blocks in the architecture. A generic framework for object lease allows, resource owners to advertize their capabilities, and application contexts or third party service providers to compose their specific application contexts/ services over them. Internet 3.0 represents a hybrid design paradigm, between the clean-slate and dirty-slate design paradigms of most next generation Internet architecture proposals. The hybrid design paradigm of Internet 3.0 is powered by the design of the “Context Router,” that allows the implementation of the clean-slate ideas of Internet 3.0 architecture over the existing Internet with minimal changes. Also, the design of the Internet 3.0 architecture is based on the “generalized three- tier object model” that realizes the current Internet as a special case within its more generic framework. This ensures the ease of deployability of Internet 3.0 without adversely affecting the current Internet.

References

[ABO04] Bernard Aboba, W. Dixon “IPsec-Network Address Translation (NAT) Compatibility Requirements, RFC 3715, March 2004

[ADA02] N. Adam, V. Atluri, E. Bertino, and E. Ferrari. "A Content-based Authorization Model for Digital Libraries," IEEE Trans. on Knowledge and Data Engineering 14(2):296-315, 2002.

[AGG08] V. Aggarwal, O. Akonjang, A. Feldmann, “Improving user and ISP experience through ISP-aided P2P locality,” INFOCOM Workshops 2008, 13-18 April 2008, pp 1 - 6

[AGG07] V. Aggarwal, A. Feldmann, and C. Scheideler, “Can ISPS and P2P users cooperate for improved performance?” *SIGCOMM Comput. Commun. Rev.* 37, 3 (Jul. 2007), 29-40.

[AID03] M. Aida, N. Miyoshi, and K. Ishibashi. “A scalable and lightweight QoS monitoring technique combining passive and active approaches,” In Proceedings of IEEE INFOCOM ’03, March 2003.

[ALM99a] G. Almes, S. Kalidindi, and M. Zekauskas. “A one-way delay metric for IPPM,” IETF RFC 2679, September 1999.

[ALM99b] G. Almes, S. Kalidindi, and M. Zekauskas. “A one way packet loss metric for IPPM,” IETF RFC 2680, September 1999.

[AND05] T. Anderson, L. Peterson, S. Shenker, J. Turner, “Overcoming the Internet Impasse through Virtualization,” *Computer, Volume* 38, Issue 4, April 2005, pp 34-41

[AND01] David G. Andersen, Hari Balakrishnan, M. Frans Kaashoek, Robert Morris, “Resilient Overlay Networks,” Proc. 18th ACM SOSP, Banff, Canada, October 2001.

[BAN07] Eric Bangeman, “P2P responsible for as much as 90 percent of all Net traffic,” ars Technical, 3 September, 2007 http://arstechnica.com/old/content/2007/09/p2p-responsible-for-as-much-as-90-percent-of-all-net-traffic.ars

[BEL93] M. Bellare and P. Rogaway. "Entity Authentication and Key Distribution," In Douglas R. Stinson (Ed.): Advances in Cryptology - CRYPTO '93, 13th Annual International Cryptology Conference, Santa Barbara, California, USA, August 22-26, 1993, Proceedings. Lecture Notes in Computer Science 773 Springer 1994.

[BER01] E. Bertino, P. Bonatti, and E. Ferrari. "TRBAC: A Temporal Role-based Access Control," ACM Transactions on Information and System Security 4(3): 191-233, 2001.

[BER02] E. Bertino, B. Carminati, and E. Ferrari. "A Temporal Key Management Scheme for Secure Broadcasting of XML Documents," 9th ACM Conference on Computer and Communication Security, Washington D.C. (USA), November 17-21, ACM Press, 2002.

[BOL93] J. Bolot. “End-to-end packet delay and loss behavior in the Internet,” *Proceedings of ACM SIGCOMM ’93*, September 1993.

[BOO91] Grady Booch. “Object-Oriented Design with Applications,” The Benjamin Cummings Publishing Company,Inc., Redwood City, CA., 1991.

[BR94] R. Braden, D. Clark, S. Shenker, **“**Integrated Services in the Internet Architecture: an Overview,” RFC 1633, June 1994

[BRA00] S. Brands. Rethinking Public Key Infrastructures and Digital Certificates, MIT Press, 2000.

[CAM06] J. Camenisch, T. Groß, D. Sommer. "Enhancing privacy of federated identity management protocols: anonymous credentials in WS-security," In Ari Juels, Marianne Winslett (Eds.): Proceedings of the 2006 ACM Workshop on Privacy in the Electronic Society, WPES 2006, Alexandria, VA, USA, October 30, 2006. ACM 2006.

[CAN01] R. Canetti, Y. Ishai, R. Kumar, M. Reiter, R. Rubinfeld, and R. Wright, "Selective Private Function Evaluation with Applications to Private Statistics," 20th ACM Symposium on Principles of Distributed Computing (PODC), 2001.

[CHA00] M.C. Chan, Y.J. Lin, and X. Wang, “A scalable monitoring approach for service level agreements validation,” In *IEEE International Conference on Network Protocols (ICNP)*, pages 37–48, 2000.

[CHO05] B.Y. Choi, S. Moon, R. Cruz, Z.-L. Zhang, and C. Diot, “Practical delay monitoring for ISPs,” In *Proceedings of ACM CoNEXT ’05*, 2005.

[CIA03] L. Ciavattone, A. Morton, and G. Ramachandran, “Standardized active measurements on a tier 1 IP backbone,” *IEEE Communications*, 41(6):90–97,June 2003.

[COM] Microsoft Corporation, “The Component Object Model (Technical Overview),” Microsoft Corporation, One Microsoft Way, Redmond, WA.

[CORBA93] Object Management Group, “The Common Object Request Broker: Architecture and Specification(CORBA),” Object Management Group (OMG), Framingham, MA., Revision 1.2. Draft 29, December 1993.

[COL01] R. Cole and J. Rosenbluth, “Voice over IP Performance Monitoring,” *ACM SIGCOMM Computer Communcation Review*, April 2001.

[DAM01]Nicodemos Damianou, Naranker Dulay, “The Ponder Policy Specification Language,” Lecture Notes in Computer Science, Springer-Verlag,2001,pp 18—38

[DAN07] Marco Danelutto; Paraskevi Fragopoulou, Vladimir Getov (Eds.), “Making Grids Work,” Proceedings of the CoreGRID Workshop on Programming Models Grid and P2P System Architecture Grid Systems, Tools and Environments, Heraklion, Crete, GreeceSpringer US, 12-13 June 2007, pp 309-321

[DAS00] L. A. DaSilva, “QoS mapping along the protocol stack: discussion and preliminary results,” IEEE International Conference on Communications, 2000, Volume 2, pp 713-717

[DEM02] C. Demichelis and P. Chimento, “IP packet delay variation metric for IP performance metrics (IPPM),” IETF RFC 3393, November 2002.

[DEN87] DE. Denning, "An Intrusion-Detection Model," IEEE Transactions on Software Engineering, 13(2):222-232, 1987.

[DIN03] I. Dinur and K. Nissim. "Revealing Information while Preserving Privacy." ACM Sysmposium on Principles of Database Systems (PODS), 2003.

[DUR07] A. Durresi and V. Paruchuri, "Anonymity in the Internet," Cluster Computing: The Journal of Networks, Software Tools and Applications, Springer, 10(1):57-66, 2007.

[DUR09] A. Durresi, V. Paruchuri and L. Barolli, "FAST: Fast Autonomous System Traceback," Journal of Network and Computer Applications, 32(2):448-454, 2009.

[FEA07] Nick Feamster, Lixin Gao and Jennifer Rexford, "CABO: Concurrent Architectures are Better Than One, NSF NeTS FIND Initiative," http://www.nets-find.net/Funded/Cabo.php

[FEA05] Nick Feamster, Ramesh Johari, Hari Balakrishnan, "Implications of autonomy for the expressiveness of policy routing," ACM SIGCOMM Computer Communication Review, Volume 35, Issue 4 (October 2005), pp 25 - 36

[FEI01] J. Feigenbaum, et al. "Secure Multiparty Computation of Approximations." 28th International Colloquium on Automata, Languages, and Programming (ICALP), Springer Verlag LNCS 2076, 2001.

[FEI06] Joan Feigenbaum, Rahul Sami, Scott Shenker, "Mechanism design for policy routing," Journal of Distributed Computing, Springer, Volume 18, Number 4 / March, 2006, pp 293-305

[FIE87] U. Fiege, A. Fiat, and A. Shamir. "Zero Knowledge Proofs of Identity." Proceedings of the 19th Annual ACM Conference on Theory of Computing, New York, New York, pp. 210-217, 1987.

[FIND] NeTS FIND, http://www.nets-find.net/

[FIRE] FIRE,http://www.ict-fireworks.eu/

[GAB99] E. Gabber, P. B. Gibbons, D. M. Kristol, Y. Matias, and A. J. Mayer. "Consistent, Yet Anonymous, Web Access with LPWA," Communications of the ACM, 42(1), pp. 42-47, 1999.

[GAO00] L. Gao, "On Inferring Automonous System Relationships in the Internet," IEEE Globe Internet, Nov 2000

[GENI] GENI: Global Environment for Network Innovations, http://www.geni.net/

[GENI01] GENI-SE-SY-RQ-01.9, "GENI Systems Requirements," Prepared by GENI Project Office, BBN Technologies, January 16, 2009

[GENI02] GENI-SE-CF-RQ-01.3, "GENI Control Framework Requirements," Prepared by GENI Project Office,, BBN Technologies, January 9, 2009

[GOL99] D. M. Goldschlag, M. G. Reed, and P. F. Syverson. "Onion Routing," Communications of the ACM, 42(1), pp. 39-41, 1999.

[GRE05] Albert Greenberg, Gisli Hjalmtysson, David A. Maltz, Andy Myers, Jennifer Rexford, Geoffrey Xie, Hong Yan, Jibin Zhan, Hui Zhang, "A Clean Slate 4D Approach to Network Control and Management," ACM SIGCOMM Computer Communication Review. 35(5). October, 2005.

[HUA97] Jean-Francois Huard, Aurel A. Lazar, "On QOS Mapping in Multimedia Networks," Computer Software and Applications Conference, Annual International, pp. 312, COMPSAC '97 - 21st International Computer Software and Applications Conference, 1997.

[HAI00] T. Hain, "Architectural Implications of NAT," RFC 2993, November 2000.

[HOH06] A. Hoheisel, "Grid Workflow Execution Service–Dynamic and interactive execution and visualization of distributed workflows," In Proceedings of the Cracow Grid Workshop 2006, Cracow, 2006.

[HOL01] Matt Holdrege, Pyda Srisuresh, "IP Network Address Translator (NAT) Protocol Issues," November 1998, draft-ietf-nat-protocol-issues-01.txt

[HOL00] M. Holdrege, P. Srisuresh, "Protocol Complications with the IP Network Address Translator," RFC 3027, January 2000.

[JIN04] Jingwen Jin, Klara Nahrstedt, "QoS Specification Languages for Distributed Multimedia Applications: A Survey and Taxonomy," IEEE MultiMedia, vol. 11, no. 3, pp. 74-87, July 2004

[JOR07] Scott Jordan, “Implications of Internet architecture upon net neutrality,” ACM Transactions on Computational Logic, Vol. V, No. N, July 2007, pp 1 - 27.

[KAG02] Lalana Kagal, “"Rei : A Policy Language for the Me-Centric Project",” Tech Report, HP Labs, September 2002

[KLA00] Klara Nahrstedt, Duangdao Wichadakul, Dongyan Xu, “Distributed QoS Compilation and Run time Instantiation,” Proceedings of IEEE/IFIP International Workshop on QoS, 2000

[KN98] K. Nichols, S. Blake, F. Baker, D. Black, **“**Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers,” RFC 2474, December 1998

[KOL02] C. Koliver, K. Nahrstedt, J. Farines, J. D. Fraga, and S.A. Sandri, „Specification, Mapping and Control for QoS Adaptation,” *Real-Time Syst.* 23, 1/2 (Jul. 2002), 143-174.

[LEE08] Gene Moo Lee, Taehwan Choi, “Improving the Interaction between Overlay Routing and Traffic Engineering,” NETWORKING 2008 Ad Hoc and Sensor Networks, Wireless Networks, Next Generation Internet, Lecture Notes in Computer Science, Springer, Volume 4982/2008, 2008

[LEI08] Tom Leighton, “Improving performance in the Internet,” ACM Queue, Volume 6, Issue 6, October, 2008, pp 20-29.

[LEH07] William H. Lehr, Sharon E. Gillett, A. Marvin, Jon Sirbu, M. Peha, “Scenarios for the Network Neutrality Arms Race,” International Journal of Communication 1, 2007, pp 607-643

[LI99] B. Li and K. Nahrstedt, “A control-based middleware framework for quality of service adaptation,” *IEEE Journal on Selected Areas in Communications (JSAC)* 17(9): 1632-1650

[LIU05] Y. Liu, H. Zhang, W. Gong, D. Towsley, “On the interaction between overlay routing and underlay routing,” INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE, Vol. 4 (2005), pp. 2543-2553 vol. 4.

[LUP98] Emil Constantin Lupu, “A Role-Based Framework for Distributed Systems Management,” Journal of Network and Systems Management, 1998

[MAR96] Damian Marriott, Morris Sloman, “Management Policy Service for Distributed Systems,” IEEE Third Int. Workshop on Services in Distributed and Networked Environments (SDNE'96},1996,pp 2--9

[MIR05] J. Mirkovic, S. Dietrich, D. Dittrich, P. Reiher. Internet Denial of Service: Attack and Defense Mechanisms, Prentice Hall, 2005.

[MON00] G. Montenegro and M. Borella, “RSIP Support for End-to-end IPsec", RSIP Support for End-to-end IPsec, July 2000

[NAK03] A. Nakao, L. Peterson and A. Bavier, “A routing underlay for overlay networks,” In *Proceedings of the 2003 Conference on Applications, Technologies, Architectures, and Protocols For Computer Communications* (Karlsruhe, Germany, August 25 - 29, 2003). SIGCOMM '03. ACM, New York, NY, 11-18.

[NIC93] John R. Nicol , C. Thomas Wilkes , Frank A. Manola, Object orientation in heterogeneous distributed computing systems, Computer, v.26 n.6, p.57-67, June 1993

[OMG94a] Object Management Group, “Common Facilities Architecture,” Object Management Group (OMG), Framingham, MA., revision 2.0 edition, 26 September 1994. OMG TC Document 94-9-40.

[ON606] (Online) P4P working group, http://www.openp4p.net/

[ON607] (Online)P2PNext Project, http://www.p2p-next.org/

[ON408] (Online) AKARI Project, http://akari-project.nict.go.jp/eng/index2.htm

[ON431] (Online) FEDERICA, http://www.fp7-federica.eu/

[OZC94] M. Tamer Ozsu, Patrick Valduriez , Umeshwar Dayal, "Distributed Object Management," Morgan Kaufmann Publishers Inc., San Francisco, CA, 1994

[PAU09] Subharthi Paul, Jianli Pan, Raj Jain, "Architectures for the Future Networks and the Next Generation Internet: A Survey," WUSTL Technical Report, WUCSE-2009-69, October 2, 2009, 59 pp.

[PAU10] Subharthi Paul, Jianli Pan, Raj Jain, "Multi-Tier Diversified Architecture for the Next Generation Internet, " Proceedings of Cloud Computing and virtualization Conference (CCV 2010), Singapore, May 17-18, 2010.

[PAS01] A. Pasztor and D. Veitch, "A precision infrastructure for active probing," In *Passive and Active Measurement Workshop*, 2001.

[PAX97] V. Paxson, "*Measurements and Analysis of End-to-End Internet Dynamics*," PhD thesis, University of California Berkeley, 1997.

[PAX98] V. Paxson, G. Almes, J. Mahdavi, and M. Mathis, "Framework for IP performance metrics," IETF RFC 2330, 1998.

[PET06] L. Peterson, A. Bavier, M. E. Fiuczynski, and S. Muir, "Experiences building planetlab," in *OSDI '06: Proceedings of the 7th symposium on Operating systems design and implementation*, Berkeley, CA, USA: USENIX Association, 2006, pp. 351-366

[REI99] M. Reiter and A. Rubin. "Anonymous Web Transactions with Crowds," Communications of the ACM, 42(1), pp. 32-38, 1999.

[ROB08] Mike Robuck, "Survey: P2P sucking up 44% of bandwidth," CED Magazine, 25 June 2008, http://www.cedmagazine.com/P2P-44-percent-bandwidth.aspx

[ROS01] E. Rosen, A. Viswanathan, R. Callon, "Multiprotocol Label Switching Architecture," Request for Comments: 3031, January 2001

[ROS05] J. Wang, D. Rosca, W. Tepfenhart, A. Milewski, and M. Stoute, "An intuitive formal approach to dynamic workflow modeling and analysis," In *Proceedings of the Third International Conference on Business Process Management* (Nancy, France, 2005).

[SAN99] R. S. Sandhu, V. Bhamidipati, Q. Munawer. "The ARBAC97 Model for Role-Based Administration of Roles," Transactions on Information and System Security, 2(1), pp. 105-135 (1999).

[SHA05] B. Shafiq, A. Samuel, H. Ghafoor, "A GTRBAC based system for dynamic workflow composition and management Object-Oriented Real-Time Distributed Computing," 2005, ISORC, 18-20 May 2005 Page(s): 284 – 290

[SIE00] S. Sieh, F. Ho, Y. Huand and J. Luo, "Network Address Translators: Effects on Security Protocols and Applications in the TCP/IP Stack," IEEE Internet Computing, November-December 2000.

[SLO94a] Morris Sloman, K Twidle, "Domains: A Framework for Structuring Management Policy," Network and Distributed Systems Management, chapter 16, 1994

[SLO94b] Morris Sloman, "Policy Driven Management For Distributed Systems," Journal of Network and Systems Management,1994, Vol 2, pp 333—360

[SLO99] Morris Sloman, Emil Lupu, "Policy Specification for Programmable Networks," 1999

[SRI99] P. Srisuresh, "Security Model with Tunnel-mode IPsec for NAT Domains", RFC 2709, October 1999.

[SOL95] Richard Mark Soley , Christopher M. Stone, Object management architecture guide (3rd ed.), John Wiley & Sons, Inc., New York, NY, 1995

[SON01] DX Song, A Perrig, "Advanced and Authenticated Marking Schemes for IP Traceback," Proceedings of IEEE Infocom, , Anchorage, AK , pp. 878-886, 2001

[STE00] M. Stenberg, S. Paavolainen, T. Ylonen, and T. Kivinen, "IPsec NAT-Traversal," draft-stenberg-ipsec-nat-traversal-00.txt, July 14, 2000

[SYV00] P. Syverson, M. Reed, D. Goldschlag, "Onion Routing Access Configurations," *DARPA Information Survivability Conference and Exposition,* vol. 1, pp. 0034, DARPA Information Survivability Conference & Exposition - Volume 1, 2000.

[TUR107] Jonathan Turner, Patrick Crowley, Sergey Gorinsky, John Lockwood, An Architecture for a Diversified Internet, NSF NeTS FIND Initiative, http://www.nets-find.net/Funded/DiversifiedInternet.php

[TUR207]Jonathan Turner, Patrick Crowley, John DeHart, Amy Freestone, Brandon Heller, Fred Kuhns, Sailesh Kumar, John Lockwood, Jing Lu, Michael Wilson, Charles Wiseman and David Zar, Supercharging PlanetLab - a High Performance, Multi-Application, Overlay Network Platform Multi-Application, Overlay Network Platform, In *Proceedings of ACM SIGCOMM*, 8/2007.

[UNO95] Object Management Group, "Universal Networked Objects," Object Management Group (OMG), Framingham,MA, March 1995.

[VMWare] VMWare,http://www.vmware.com/

[VServer] VServer, http://linux-vserver.org/Welcome_to_Linux-VServer.org

[WAT05] J. Han, D. Watson, F. Jahanian, "Topology aware overlay networks," Proceeding of IEEE INFOCOM, Vol 4, 13-17 March, 2005, pp 2554 - 2565

[XEN] Xen, http://www.xen.org/

[YAJ99] M. Yajnik, S. Moon, J. Kurose, and D. Towsley, "Measurement and modeling of temporal dependence in packet loss," In *Proceedings of IEEE INFOCOM '99*, March 1999.

[YAN07] Hong Yan, David A. Maltz, T. S. Eugene Ng, Hemant Gogineni, Hui Zhang, Zheng Cai, "Tesseract: A 4D Network Control Plane," Proceedings of USENIX Symposium on Networked Systems Design and Implementation (NSDI '07), April 2007.

[YU03] T. Yu and M. Winslett. "A Unified Scheme for Resource Protection in Automated Trust Negotiation," Proceedings of the IEEE Symposium on Security and Privacy, Oakland, May 2003.

[YU03a] T. Yu, M. Winslett, and K. E. Seamons. "Supporting Structured Credentials and Sensitive Policies through Interoperable Strategies for Automated Trust Negotiation," ACM Transactions on Information and System Security 6(1), pp. 1-42, 2003.

[YU05] J. Yu, R. Buyya, "A Taxonomy of Scientific Workflow Systems for Grid Computing," SIGMOD RECORD, VOL 34; NUMB 3, 2005, pages 44-49

[ZHA01] Y. Zhang, N. Duffield, V. Paxson, and S. Shenker, "On the constancy of Internet path properties," In *Proceedings of ACM SIGCOMM Internet Measurement Workshop '01*, November 2001.

[ZSE01] T. Zseby, "Deployment of sampling methods for SLA validation with non-intrusive measurements," In *Proceedings of Passive and Active Measurement Workshop*, 2001.