A Unifying Infrastructure for Internet Services

Jaideep Chandrashekar
University of Minnesota
Minneapolis, MN 55414
jaideepc@cs.umn.edu

Y. Thomas Hou
Fujitsu Laboratories of America
Sunnyvale, CA 94085
thou@fla.fujitsu.com

Zhi-li Zhang
University of Minnesota
Minneapolis, MN 55414
zhzhang@cs.umn.edu

Abstract— Effective service delivery capabilities are critical to the transformation of the Internet into a viable commercial infrastructure. At the same time, there are several design limitations that prevent this. In this paper, we propose a novel service overlay architecture that serves as a *flexible, unifying platform* for delivering services over the Internet. We introduce a new addressing scheme and an associated *service layer*, which enables service-oriented routing and forwarding over the underlying IP network domain. We also describe the functionality of the network elements that are introduced by our architecture, namely *service gateway* (SG) and *service point-of-presence* (S-PoP). We also present examples to demonstrate the efficacy of our architecture.

Key Words: Internet architecture, service overlay network, naming and addressing, routing, Internet services.

I. Introduction

In our view, there are two fundamental limitations of the current Internet that prevent it from being a viable platform for delivering services.

To begin with, the Internet only provides a best effort delivery service. Further compounding this is the fact that a typical packet has to go through many congested network peering points. Such a scenario is inherently unsuitable for delivering value added services. Traditional approaches to solving this problem involve building application specific overlay networks that enable packets to avoid the congested peering points in order to reach the destination. This is the approach taken in [1], [8] and [12].

The other, more fundamental problem concerns the current naming and addressing paradigm. The current model uniquely associates a name to a physical interface (identified by IP address). This model is quite adequate for host to host communication, but unsuitable for almost anything else (e.g., anycast, multicast). Moreover, it is a poor model to support *service availability.* That is, if a service is identified by a certain name, then the availability of that service depends wholly on the availability of the host that it is mapped to. If for any reason, the host is unreachable, there is no easy way to redirect the service request to another capable host. This issue is more fundamental and the solution has far reaching consequences. Rather than using ad-hoc methods to work around the inadequacies of the current naming scheme, we propose to abandon it altogether in favor of a new naming and addressing scheme for the next generation Internet. The salient feature of the this new scheme is the logical separation of the name from the actual entity that provides the service (as described by the name). The rest of this paper is devoted to the development of a service overlay network (SON) architecture that addresses fundamental limitations for supporting services over the Internet, as discussed previously. Unlike other overlay networks being deployed or proposed in the literature, Our SON architecture builds upon the new naming and addressing scheme and is designed to be a *unified* platform for efficient and flexible deployment of *all* types of Internet services.

In our architecture, we decouple the system into two distinct layers, a data transport plane (comprising the autonomous systems) and a service plane (comprised of a set of service networks). Our service-based addressing scheme, which forms a layer above the current network layer, consists of a 4-byte Service ID (`SID`) and a variable length Object ID (`OID`). The `SID` is used to identify a particular *service network* while the `OID` is used to locate a specific *object* within the particular service network.

There are numerous advantages in introducing a new addressing scheme to support Internet services, the most important of which is that it enables a *functional decoupling* of the *service network* from the underlying *data transport network.* This makes it possible to define simple and meaningful bilateral business relationships among providers: a service network is concerned with providing specific services to users; at the same time the service network itself purchases access (points of presence) and resources (bandwidth) from the data network domains; the underlying IP architecture in the data network domains provides transport for service networks (i.e. bandwidth as a commodity). This allows each layer to evolve independently, without being constrained by the limitations of the other.

It should be seen that the architecture we are proposing is quite ambitious, in the sense that it attempts to solve many problems in the current Internet. However, there have been many efforts focusing on isolated aspects of the problem. The most notable among these has been the recent widespread deployment of Content Distribution Networks [1], [12], which attempt to address the problem of delivering web content to end users — by building globally distributed delivery networks (coordinated

0-7803-7400-2/02/$17.00 © 2002 IEEE

collection of cache servers). Problems with such approaches have been detailed in [13]. Another recent work [4] describes an addressing schema that is somewhat similar to ours. However, the motivation here is limited to providing an alternative to *IPv6* deployment and does not deal with service delivery issues. The work that comes closest to our own architecture is the content routing scheme described in [6] in which the authors propose an integrated naming and routing scheme for web content delivery. Packets carry the full name of the resource they are requesting and intermediate nodes forward these packets based on the name of the resource. However, there are many scalability concerns with this scheme which our own architecture addresses.

The remainder of this paper is organized as follows. In Section II, we provide an overview of our proposed SON architecture by introducing key components and describing basic functionality. We also discuss the important advantages provided by our SON architecture. In Section III-A, we give an outline of the SGRP and in Section III-B, we discuss the packet forwarding behavior. To demonstrate the feasibility of our SON architecture, in Section IV, we show how various Internet services can be supported under our architecture. Section V concludes this paper and points out some of the directions for future work.

II. Architectural Overview

In this section we first present an overview of our proposed architecture, following which we present some details about the various components.

A. Overview

In our proposed architecture, we distinguish between the *data transport networks*, which correspond roughly to the existing autonomous systems, and the *service overlay networks* (SON). The role of the data transport networks is to transport bits from point to point. The service networks, on the other hand, are designed to provide specific value-added services to subscribers. These networks are operated by service providers and can be visualized as clouds which interface at multiple points with the data transport networks. Client requests are routed over the data transport network to the nearest (or most appropriate) point of entry into a particular service cloud. The client's request is then served from some host inside the cloud. The architecture is depicted in Fig. 1.

The *logical decoupling* between the data network domains and the service networks allows the independent evolution of each realm, thus providing the flexibility for the deployment of future Internet services, while still supporting existing services. This decoupling is enabled by three key components: a new *naming and addressing* scheme that is a significant departure from the existing IP addressing scheme, *service gateways (SG)*, and *service points-of-presence (S-PoP)*. In addition, there is also a new routing protocol, called *service gateway routing protocol* (SGRP), that binds these components together.

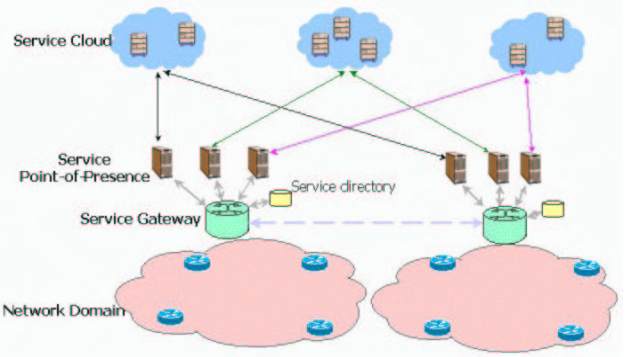

Fig. 1. Architecture of service overlay networks. The service networks are depicted as opaque entities since they can incorporate arbitrary routing mechanisms independent of any routing system external to the domain.

B. Components

We describe three key components in our SON architectural framework: *naming and addressing*, *service gateway* and *service point-of-presence.*

1) Naming and Addressing: The addressing scheme that we introduce is essentially a two level address hierarchy composed of two new identifiers, namely *Service ID* (`SID`) and *Object ID* (`OID`). The `SID` is used to identify a particular *service network* and is defined to be 4 bytes in length. The `OID` is used to map a specific *object* within the particular service network. The syntax and semantics of the `OID` is defined by the particular service network and its length can be variable. The `SID`/`OID` addresses are carried in a *shim* header (between the normal IP header and data field) in a packet. This header is read only by SGs that perform the forwarding based on the value of `SID` carried in the packet. The `SID`:`OID` tuple is used for end to end addressing. The IP address has reduced scope and is used only to carry packets between Service Gateways. Service networks are also associated with service names. Service name to `SID` resolution is performed by *Service Directories* which are located near SGs. Queries that are addressed to the service directory are expressed in some form that is easy to parse (e.g. XML).

2) Service Gateway (SG): SGs are deployed at the edge of a network domain and perform a role similar to that of domain border routers in the current Internet. An SG serves as an interface between the data transport network and the service plane. In the data plane, an SG forwards packets based on the `SID` carried in the packets. In the control plane, a SG builds and maintains a routing table based on the `SID` address space. The details on routing and forwarding will be presented in Section III-A.

3) Service POP (S-PoP): A service point-of-presence (S-PoP) serves as a local proxy for a particular service network. An S-PoP is located in the proximity of an SG. Under our architecture, we have many S-PoPs that are located near a SG,

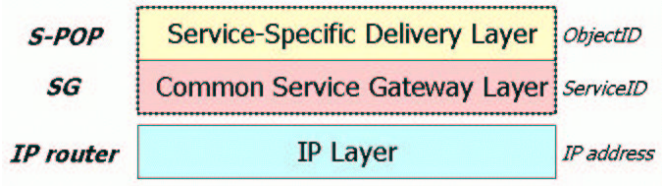

Fig. 2. Protocol layering and address naming under our SON architecture.

each providing a point of entry into a particular service cloud[1].

A S-PoP is associated with, and receives traffic meant for, a particular service network (and this is based on the SIDcarried in the packet). S-PoPs also have attributes that describe their capability (which may differ from S-PoP to S-PoP). When first deployed, an S-POP registers itself with the local SG which proceeds to create a corresponding entry in the forwarding table). In addition, the SG propagates this service adjacency to other SGs in neighboring domains.

When a SG forwards a packet to a particular S-PoP based on the packet's SID, the packet is considered to have entered, *logically*, into the service cloud. Within the service cloud, the packet forwarding is based on the OID. The definition of object is service-specific (e.g., a web page, a calling number for VoIP) and is opaque to the SGs involved in forwarding.

A characteristic of the OID is that binding from OID to a physical object is *dynamic*, by which we mean that the OID is not statically mapped to any particular host, but acts more as a hash pointing to a particular host that might be able to serve a request at a given time. Thus, the service network can ensure availability by directing the OID dynamically to *any* host that meets specific criteria specified in the service request. Note that unlike the SGs, which all run the same routing protocol and forward packets based on the SID obtained from a common, global SID space, each service cloud can implement its own *service-specific routing* (using its particular OID space) independent from other service clouds.

Figure 2 shows the protocol layering introduced by our architecture.

C. Basic Operation

When the user needs a particular service, it creates a packet, with the destination IP address set to the the SG[2]. The client fills the packet with a description of the service that it is requesting and sends it to the SG. Upon receipt at the SG, the packet is referred to the *Service Gateway* to perform the service name resolution. Once the SG obtains the SID, it fills in the destination SID field in the packet and then forwards the packet based on the destination SID to the appropriate S-PoP (which might not be local). Upon receiving the packet, the S-PoP examines it and performs the requisite object binding (if the service is a simple request-response transaction, this binding is unnecessary, and the S-PoP can just forward the packet to a host that can service the request). Once the binding is performed, the S-PoP installs some state corresponding to the binding and fills in the OID of the packet. This OID is simply a pointer to the session state that is maintained in the S-PoP. Then the packet is forwarded to a host within the service cloud based on the OID. For the reverse path, no SID or OID name resolution is necessary as the packet already contains addressing information for the client.

D. Advantages

There are a number of important advantages of our SON architecture. Firstly, the decoupling of application services from network services potentially makes the domains easier to manage. The IP network is only used to transport bits while the service networks take care of application-level service routing and forwarding. This separation allows the network domains and the service networks to evolve independently. Secondly, our SON architecture is designed as a *flexible, unifying* platform. As we shall show later, existing Internet services which are currently delivered through some proprietary architecture, can be seamlessly supported under our SON architecture with very little deployment complexity. Therefore, our SON architecture helps to accelerate the pace of deploying new Internet services.

Thirdly, by performing the binding from OID to physical endpoint dynamically, provides a way to counter the serious and pressing concern posed by *denial-of-service* attacks. Furthermore, because of the distributed nature of the service clouds, choking a single point of entry (by targeting the S-PoP) does not disable the service itself, as data can be transported to and from other ingress/egress points of a service cloud. Lastly, our SON architecture provides a framework for providers to establish multi-lateral business relationships (involving multiple network domains) to deploy end-to-end services that cannot be supported by the traditional best-effort delivery model. In particular, value-added Internet services such as VoIP, video conferencing, Video-on-Demand (VoD) e.g. can be supported over our SON with end-to-end QoS guarantees. An example is presented in section IV-A.

III. Routing and Forwarding

A. Service Gateway Routing Protocol

SGRP, which forms the control plane component of our architecture, has two main functions — construction and management of a virtual network of service gateways and the distribution of service reachability information between these service gateways. Although functionally similar to the existing Border Gateway Protocol [11], it is very different in design. Although it would be possible to extend BGP to our architecture, we choose to design it from scratch so as to avoid some of the

[1] As a service provider would typically deploy more than one S-PoP , the collection of these S-PoPs can be viewed as a logical *service cloud*.

[2] The address of the local SG is configured at the client in a manner similar to configuring the default gateway.

design problems inherent in BGP. The design issues and operational shortcomings in BGP have been studied extensively in [5],[9] and [10]. In fact, there is already discussion in the research community to reach a consensus about how the next generation Inter Domain Routing system will look like [3]. In this context, we wish to present SGRP as a viable alternative. In this section, we present a brief overview of the protocol, postponing a more detailed presentation for the future.

To begin with our naming and addressing scheme allows the separation of the virtual topology of SGs and the actual service reachability. In SGRP, the internals of the service networks are hidden away and only information about ingress points (of the service networks) is advertised to the outside. This serves to isolate any internal instabilities from the external routing domain. To avoid the pitfalls exhibited by path vector routing protocols, we base SGRP upon a link state approach. This has the added benefit of potentially reducing the resource requirements in maintaining routing state. The current Internet topology has evolved into a mesh like structure (rather than the well defined hierarchy that it started out as). In BGP, a destination is mapped to a path (ordered list of autonomous systems). With the observed topology, one can expect the number of paths to grow very rapidly with the network size, so a reduction in routing state could be achieved by maintaining link information rather than path information. For the rest of this section, we discuss the different aspects of SGRP.

The construction of the virtual network of SGs is achieved by the controlled distribution of *link state advertisements* (LSA), while the service reachability information is disseminated through the network by propagating *service state advertisements* (SSA). The SSAs are generated in response to an S-PoP registering itself with a local SG. This registration qualifies the S-PoP to receive, from the SG, traffic meant for the `SID` of the service network that it represents. The registration message describes the capabilities of the S-PoP to the SG and includes the following information.

- The service ID (`SID`) of the service network that the S-PoP is proxying for.
- An optional numeric value that represents the S-PoP's position within the service providers hierarchy[3]. If no value is specified, a default value is assumed.
- A bit-mask that describes special capabilities of the S-PoP. This description is service specific. For example, a certain bit could indicate that the S-PoP handles objects that are "cacheable" (so the packet from the client requesting an object that is described as cacheable, can be serviced by the S-PoP). Another possible scenario is the S-PoP providing a transcoding function, adapting content to suit low-resolution clients. A particular bit could represent the transcoding capability. Requests from low resolution clients (with the particular bit enabled) could be directed to it

1) SG Topology Construction: The SGs distribute LSAs in order to build the SG topology map[4]. LSA distribution is accomplished with a mechanism similar to that employed in existing link state protocols. In SGRP, an advertised LSA contains more than just a set of neighbor adjacencies. It might be possible to associate certain QoS parameters with the virtual links and include these attributes in the path selection process. Maintaining topology information for the entire Internet is both computationally intensive and requires a lot of space. In most stub networks, it is not necessary to maintain this topology, as all of the traffic will be directed to one or more providers. In situations like this, to reduce the need for a domain to maintain full network topology, we use the concept of a core network (which essentially represents the whole Internet, except the domain in consideration). Domains can simply point all routes to this core network via an upstream domain. So all packets for which there is no match in the local domain will be forwarded to this upstream domain. This is very similar to the concept of a default route network prefix that is employed in the IP address realm (which matches any valid IP address). There is certain tradeoff involved here: if a domain chooses not to maintain full global connectivity state, it is at the cost of routing optimality. Sub-optimality in routing could occur when a domain multi-homes to more than one domain (all within the data transport plane). The choice of a next hop to choose for a particular `SID` is made with abstracted topology information, potentially leading the SG to choose a longer path, when there is a shorter one available.

2) Service Reachability Propagation: In the next stage, SGs exchange SSAs, and populate the `SID` forwarding tables. An SSA carries the following information.

- The *identifier* of the SG that is originating the advertisement.
- *Service information*, which is a triple of the form (`SID`, S-PoP level, service attributes). The `SID` represents the unique tag for the service that is registered at the SG by a nearby S-PoP. *S-PoP level* has the same semantics as previously explained and is included in messages between SGs since it is used in the packet forwarding. The service attributes are associated bits that provide some service-specific information. These bits include what was announced by the S-PoP to the SG.
- *Distribution tags*, which are essentially directives that control how the SSAs are to be propagated by the SGs that receive the SSA. The exact nature of these tags are being developed and will be presented in a future paper.

[3]This is particularly useful for certain services that naturally support the concept of hierarchy, e.g. web cache services. If not used, it can be safely ignored.

[4]LSA's used in SGRP have somewhat similar semantics as the identically named messages exchanged in OSPF/IS-IS, with the difference that what is advertised are actually *virtual* links, rather than *physical* adjacencies

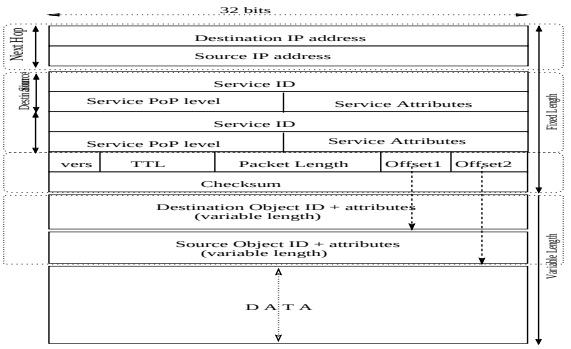

Fig. 3. Packet header format

The SSA distribution is initiated by an S-PoP's registration with a local SG. SSA distribution respects any specific routing policies that are specified by the domain. Our architecture allows routing policies to be specified on a *per service* basis, which allows fine-grained policy control. In SGRP, service reachability advertisements are independent of the LSA distribution. Thus, a SG can see paths to all other SGs, but it does not know the services that are supported at these SGs. So, a SG can forward packets (for a service) to a destination SG only if the latter has indicated that it will accept packets for the particular service. The determination of which SG can forward packets to a destination SG for a particular `SID` is defined by the distribution tags that enforce the routing policies. In the next section, we describe the actual packet forwarding mechanism which uses the `SID` table that is constructed by SGRP.

B. Packet Forwarding

The packet forwarding function at each SG is the mapping of the (`SID`, S-PoP level, service attributes) triple to a next hop SG. As part of our architecture, we define a very flexible packet structure, as shown in Fig. 3. The semantics of the different sections in the packet are as follows.

- *Source and Destination IP addresses*, which are used to transport packets between SGs
- *Source and Destination service information* (`SID`, S-PoP level, service attributes), which is the information used to direct the packet to the appropriate ingress point of the service network.
- *Packet State information*
- *Object ID fields*, which is a variable length field used to carry the addresses of the entities inside the service clouds.
- *Data field*, which is of variable length and carries the actual data being transported

Although our proposed architecture is independent of any structure within the individual service networks, an internal hierarchy can be supported easily by using the S-PoP level field in the packet. S-PoP's can be assigned some level in the hierarchy (in the service network) and this can be communicated to the SG as an attribute. This attribute allows the service network to control which particular ingress point will be used for a particular session (determined when the `OID` mapping is done). This attribute also provides an implicit preference function (at an SG) to rank the S-PoPs for a given `SID`.

The `SID` switching table contains entries indexed by the triple (`SID`, S-PoP identifier, service attributes). Although, the size of the table, in theory, can be up to 2^{64} entries, which is a significant number, in practice, the number of entries is comparable to the number of network domains is far fewer than the number of routing prefixes seen in todays BGP tables.

The actual forwarding of packets at the SG is based on a perfect match of the `SID` field and a wild card match of the remaining two fields, i.e., S-PoP level and service attributes[5]. If the S-PoP level field of the packet carries a non-zero value, the SG will only forward the packet to an S-PoP that matches the value. However, if this value is zero (i.e., matching *any* S-PoP), the packet is forwarded to the local S-PoP. Thus, an S-PoP that registers itself at an SG is implicitly assigned the pop level *zero* (with local scope) along with any other number that it might be using (dictated by the service provider).

The service attributes bit-mask in a packet is used by the client to indicate certain constraints that must be satisfied along the forwarding path, for example the service attributes bit-mask can be used to forward packets based on QoS parameters. Note that all the matches are performed on fixed length fields and are thus implementable with efficient hardware.

IV. Examples

In this section, we illustrate how various Internet services can be supported under our SON.

A. Real Time Communication

One of the significant barriers to supporting real time communication over the Internet is that the underlying best effort delivery model does not provide any delivery guarantees or QoS support across network domains[6]. Our proposed architecture makes it possible to provide better QoS support by defining a clear bilateral relationship (enforced by means of SLA's) between a service provider and the the underlying data transport network. The service provider can purchase bandwidth from each data network domain, and the data network domains will guarantee a certain modicum of service for the bandwidth. Thus the service provider can provision bandwidth by cutting through domains and achieve some end-to-end service guarantees.

We describe the deployment of a VoIP service as an example of supporting a real-time communication service. A VoIP provider deploys S-PoPs in various domains and sets up a virtual network between these S-PoPs (with some performance guarantees). Now if the service provider were to provide the

[5]Here wild card is taken to mean a match that allows don't care bits.

[6]It is possible to approximate this within a domain by means of traffic engineering mechanisms.

VoIP service in some region, S-PoPs would be deployed in that region to move calls to and from the VoIP cloud to users in the region.

Let us consider a scenario where a user, say *A*, located in network domain *X* wishes to communicate with *B*, located in domain *Y*. To accomplish this, *A* sends a connection packet to the nearest S-PoP (via the SG in domain *X*), bearing the description of *B*. The VoIP cloud maps this description to some `OID` which corresponds to a path leading to the S-PoP in domain *Y*, and some locator for *B* within that domain. If there are enough resources in the service cloud to support the call, the call is admitted. This corresponds to the S-PoPs setting up some state to forward the packets for this session. Once the connection is setup, packets travel from user *A* to SG to the local S-PoP. Inside the service cloud, the packets are routed along a pre-allocated path (with bandwidth reservation) to the nearest S-PoP for *B*, which is close to the SG for domain *Y*. From the SG, the packet is then forwarded to the actual destination *B*.

The key point to note in this example is that our architecture enables a service provider to provide some level of assurance about the end-to-end performance by contracting with various data transport domains.

B. Web Content Delivery

In this setting, the deployment scenario would be very similar to what is currently seen in the Content Distribution Networks. The S-PoPs correspond to the edge servers deployed by Content Distribution Networks [1], [12]. Web content delivery transactions usually involve a short request message sent from the client which elicits a longer reply message (containing the content) from the server. A client in our architecture could simply put the URL of the object into the variable length destination `OID` field of the packet. The packet would also contain a description of the service provider that the content is requested from. Upon receipt at the SG, the packet is referred to a service directory, which determines the mapping from service name to `SID`. Once this is obtained, the SG forwards the packet to the appropriate S-PoP. At the S-PoP, the packet could be redirected to other S-PoPs by modifying the S-PoP-level field of the packet. This feature is especially useful when the cache servers are organized in a hierarchy - and cache misses are required to be sent to servers higher up in the hierarchy.

The key difference from the current Content Distribution Networks, as seen in this example is that there is no round trip time wait required for the service name to be resolved, prior to the request being made, as compared to the existing paradigm, where a separate request is first made to resolve the URL to a particular IP address, following which a connection is established with the actual host.

C. Multicast Streaming Media

Using our generalized platform, supporting multicast is relatively easy. The scenario is very similar to what is described in [2], [7]. This serves to highlight one of the key features of our architecture — current overlay networks can be easily integrated into the unifying framework which we are proposing.

V. Conclusion

In this paper, we proposed a novel architecture as a unifying platform for flexible support of service delivery over the Internet. It should be clear by now that our new architecture provides an easy deployment path that reuses the existing IP network infrastructure. This is evident from the fact that packets are routed from client to SG and between SGs using the IP address of the destination SG.

As part of our new architecture, we introduced a new service-oriented naming and addressing scheme which essentially decouples service layer and underlying IP network layer. We also described the functionalities of the network elements that are introduced as part of our architecture, namely, service gateway (SG) and service point-of-presence (S-PoP). Using examples, we demonstrated how our architecture could be used to deploy Internet services with improved performance.

Our ongoing work focuses on developing the specifics of the SGRP and producing a software prototype of our SON architecture.

References

[1] Akamai Technologies, `http://www.akamai.com`.

[2] Y. Chawathe, "Scattercast: An architecture for Internet broadcast distribution as an infrastructure service," *Ph.D. Thesis*, University of California, Berkeley. December 2000.

[3] E. Davies et al "Future Domain Routing Requirements" Internet Draft draftdraft-davies-fdr-reqs-01.txt

[4] P. Francis and R. Gummadi, "IPNL: A NAT-extended Internet architecture," in *Proc. ACM SIGCOMM*, San Diego, CA, August 2001.

[5] T. Griffin and G. Wilfong, "An analysis of BGP convergence properties," in *Proc. ACM SIGCOMM*, Cambridge, MA, August 1999.

[6] M. Gritter and D.R. Cheriton, "An architecture for content routing support in the Internet," in *Proc. 3rd USENIX Symposium on Internet Technologies and Systems*, March 2001, San Francisco, CA.

[7] Inktomi Corporation, "The Inktomi overlay solution for streaming media broadcasts," *White Paper*, `http://www.inktomi.com`.

[8] Internap Network Services, `http://www.internap.com`.

[9] C. Labovitz, A. Ahuja and A. Bose, "Delayed Internet routing convergence," in *Proc. ACM SIGCOMM*, Stockholm, Sweden, August 2000.

[10] C. Labovitz, G. Malan and F. Jahanian, "Internet routing instability," in *Proc. ACM SIGCOMM*," Cannes, France, September 1997.

[11] Y . Rekhter and T . Li "RFC 1771: A border gateway protocol 4" March 1995

[12] Speedera Networks, `http://www.speedera.com`.

[13] A. Shaikh, R. Tewari and M. Agarwal, "On the effectiveness of DNS-based server selection," in *Proc. IEEE INFOCOMM*, Anchorage, AK, April 2001.