Design of a Content-based Routing Architecture

by

Arindam Mitra

A dissertation submitted in partial satisfaction of the
requirements for the degree of

Master of Science

Department of Computer Science
Faculty of Graduate Studies
University of Manitoba

THE UNIVERSITY OF MANITOBA

FACULTY OF GRADUATE STUDIES

*****

COPYRIGHT PERMISSION PAGE

DESIGN OF A CONTENT-BASED ROUTING ARCHITECTURE

BY

Arindam Mitra

A Thesis/Practicum submitted to the Faculty of Graduate Studies of The University of Manitoba in partial fulfillment of the requirements of the degree of

MASTER OF SCIENCE

Abstract

Content networking is an emerging technology, where the requests for content accesses are steered by "content routers" that examine not only the destinations but also content descriptors such as URLs and cookies. In the current deployments of content networking, "content routing" is mostly confined to selecting the most appropriate back-end server in virtualized web server clusters. With the deployment of mechanisms for widespread replication of content among geographically distributed servers, a question arises regarding the optimal form of request routing. In this thesis, we examine the following questions via simulation-based performance modelling: (a) Is content-aware routing superior to address-based routing with wide-area replica placements? (b) Under what conditions does performance differences exist between the two approaches? and (c) What attributes of the content should be considered for efficient wide-area content routing? As part of this study, we present a novel content-based routing mechanism called the Virtual Content Network (VCN) for wide-area networks. Furthermore, this thesis examines the issues involved in creating, maintaining and using the VCN. It presents the overall architecture and outlines approaches for addressing some of the key issues in content routing that would eventually lead to the development of an IP-less routing architecture. Simulations are carried out based on the performance model that we developed to compare our content-based wide-area routing mechanism with a domain name based routing mechanism.

Acknowledgements

There are many people I would like to thank. First I would like to thank my supervisor Dr. Muthucumaru Maheswaran who is a fine resource for research ideas and a great source of help in a lot of matters. Mahes is the coolest supervisor I could ever wished for.

I thank my examining committee, comprising Dr. Jose A. Rueda from Electrical and Computer Engineering and Dr. Rasit Eskicioglu. from Computer Science, for evaluating this dissertation and Dr. Helen Cameron for chairing my thesis defence.

I thank TRLabs for their financial support and for the wonderful research environment they provide to students. I thank them for supporting the patent application based on my thesis. I also appreciate the support and efforts from Dr. Rueda for his insights into the material of this dissertation.

I appreciate the help from my colleagues, especially Vasee for his ideas and help in a number of matters and for sharing some of the codes from his simulator for developing the simulator in my thesis.

Finally, a special thanks to my parents, sister and brother-in-law for their constant support and perseverance.

Contents

1 Introduction 1

2 Literature Review 5

2.1 Class I: Name-based Routing Models 6

2.2 Class II: Content Routing Models 7

2.3 Simulation Techniques 12

2.4 DNS-based Routing Techniques 12

2.5 Solution Path 14

3 Issues in Content Aware Networking 16

3.1 Overview 16

3.2 Scalability 17

3.3 Efficiency 18

3.4 Content Characterization Problem 19

4 Protocol Independent Content Switching Architecture 21

4.1 Overall Architecture 21

4.2 Functional Components 23

4.2.1 Server Sites 23

4.2.2 Client Sites 24

4.2.3 Virtual Content Network 24
4.3 Content Characterization and Classification for Content-based Routing . 27
4.3.1 Content Characterization Solution 27
4.3.2 Content Classification 31
4.4 Content-based Tags 39
4.4.1 Temporary Tags 39
4.4.2 Tag Format 40
4.5 PICS: Modes of Operation 45
4.5.1 Route Push Method 47
4.5.2 Route Push and Pull Method 50
4.6 PICS Forwarding Algorithm 53

5 Modeling for Performance Analysis **55**
5.1 Overview 55
5.2 Assumptions 57
5.3 Model for PICS Routing Scheme 59
5.4 Model for DNS-based Routing Scheme 62
5.5 Link Utilization 64
5.6 Overhead Due to Tag Dissemination and Status Update 65
5.6.1 PICS Routing Scheme 66
5.6.2 DNS-based Routing Scheme 67

6 Simulations **71**
6.1 Simulation Setup 71
6.2 Results and Discussion 75
6.2.1 Resolution Time 75
6.2.2 Content Delivery Time 82

6.2.3 Throughput 87
6.2.4 Link Utilization 91
6.2.5 Impact of Streaming and Non-streaming Documents in PICS . . . 98

7 Applications of Content-based Routing **106**
7.1 Overview 106
7.2 Content Delivery Networks 107
7.3 Scalable Service Deployment on the Internet 107
7.4 Video-on-Demand Services 108
7.5 Intrusion Detection and Security Systems 108

8 Conclusions and Future Work **110**
8.1 Thesis Contribution 110
8.1.1 Architecture and Design Benefits 111
8.1.2 Performance Benefits: VCN vs DNS Based Routing 114
8.1.3 Next Generation Networking 116
8.2 Comparison with Existing Routing Methods 117
8.3 Future Research Directions 120

List of Tables

4.1 Structural Attributes 30
4.2 Supporting Attributes 31
4.3 A probable list for type and CF field combination. 43

List of Figures

4.1 An example deployment of a virtual content network. 22

4.2 The content tagging process in a CER. 25

4.3 The content-based routing process in a CSR. 26

4.4 (a) An example document-AS matrix with 8 documents and 3 autonomous systems. Each matrix element A[i,j] called *relative weight* is a relative measurement of popularity for the document i in AS j. (b) All the documents are sorted according to their relative weights for each AS (c) Grouping of documents with similar relative weights. 36

4.5 CEC creation algorithm. 39

4.6 (a) This phase is run at over periodic intervals to create the content-based tags. (b) This phase is executed when a new content is introduced to the VCN. 41

4.7 32-bit content-derived tag format. 42

4.8 Namespace tree showing the content-to-tag bindings. 44

4.9 (a) IP resolution scheme (b) Tag resolution scheme. 47

4.10 Tag Dissemination process for Route Push Method. 49

4.11 An example of request steering and content delivery for RPM. 50

4.12 An example virtual CER tree. 51

4.13 Content-based forwarding process. 54

5.1 The figure shows the total amount of time required to deliver a document from one node to another. In this example, s is the size of a content, b_0 is the bandwidth of the link between the two nodes and d_0 is the latency of the link. 59

5.2 The time values for events for discovering and accessing content in PICS routing scheme, this is a pipelined model where we assume that the packet processing time for a content router is negligible. 61

5.3 The time values for events to discover and access a content in DNS based routing scheme, this is a non-pipelined model where the packet processing time for each entity is negligible. 63

6.1 (a) The DNS name resolution process (b) Name resolution process in the VCN. 77

6.2 Resolution time with 0% replication and 10,000 requests. 79

6.3 Resolution time with 5% replication and 10,000 requests. 79

6.4 Resolution time with 25% replication and 10,000 requests. 80

6.5 Resolution time with 0% replication and 100,000 requests. 80

6.6 Resolution time with 5% replication and 100,000 requests. 81

6.7 Resolution time with 25% replication and 100,000 requests. 81

6.8 Content Delivery time with 0% replication and 10,000 requests. 83

6.9 Content Delivery time with 5% replication and 10,000 requests. 84

6.10 Content Delivery time with 25% replication and 10,000 requests. 84

6.11 Content Delivery time with 0% replication and 100,000 requests. 85

6.12 Content Delivery time with 5% replication and 100,000 requests. 85

6.13 Content Delivery time with 25% replication and 100,000 requests. 86

6.14 Throughput with 0% replication and 10,000 requests. 88

6.15 Throughput with 0% replication and 100,000 requests. 88
6.16 Throughput with 5% replication and 10,000 requests. 89
6.17 Throughput with 25% replication and 10,000 requests. 89
6.18 Throughput with 5% replication and 100,000 requests. 90
6.19 Throughput with 25% replication and 100,000 requests. 90
6.20 Link utilization for active links with 0% replication and 10,000 requests. . 92
6.21 Link utilization for active links with 0% replication and 100,000 requests. 92
6.22 Link utilization for active links with 5% replication and 10,000 requests. . 93
6.23 Link utilization for active links with 5% replication and 100,000 requests. 93
6.24 Link utilization for active links with 25% replication and 10,000 requests. 94
6.25 Link utilization for active links with 25% replication and 100,000 requests. 94
6.26 Link utilization for all links in the network with 0% replication and 10,000 requests. 95
6.27 Link utilization for all links in the network with 0% replication and 100,000 requests. 95
6.28 Link utilization for all links in the network with 5% replication and 10,000 requests. 96
6.29 Link utilization for all links in the network with 5% replication and 100,000 requests. 96
6.30 Link utilization for all links in the network with 25% replication and 10,000 requests. 97
6.31 Link utilization for all links in the network with 25% replication and 100,000 requests. 98
6.32 Impact of mixed traffic with set-based 0% replication and 10,000 requests. 99
6.33 Impact of mixed traffic with set-based 0% replication and 100,000 requests. 99
6.34 Impact of mixed traffic with set-based 5% replication and 10,000 requests. 100

6.35 Impact of mixed traffic with set-based 5% replication and 100,000 requests. 100
6.36 Impact of mixed traffic with set-based 25% replication and 10,000 requests. 101
6.37 Impact of mixed traffic with set-based 25% replication and 100,000 requests. 101
6.38 Impact of mixed traffic with site-based 0% replication and 10,000 requests. 102
6.39 Impact of mixed traffic with site-based 0% replication and 10,000 requests. 102
6.40 Impact of mixed traffic with site-based 5% replication and 10,000 requests. 103
6.41 Impact of mixed traffic with site-based 5% replication and 10,000 requests. 103
6.42 Impact of mixed traffic with site-based 25% replication and 10,000 requests. 104
6.43 Impact of mixed traffic with site-based 25% replication and 100,000 requests. 104

Chapter 1

Introduction

The hyper-growth of the Internet along with the emergence of several business-critical applications that heavily rely on the Internet have increased the load on its infrastructural services. Some of the crucial Internet services for applications include facilities to access content, support for efficiently transporting content and the ability to discover and access services. Caching and replication are two traditional strategies that may be employed to improve Internet services. These strategies can be implemented in two different ways: (a) on demand by the clients (content or service requesters) or (b) by the servers (content or service providers).

On-demand caching by clients is a well established technique that is commonly used in the Internet. Simplicity is one of the advantages of this technique. By reducing the net load imposed by an application on the Internet infrastructure, this technique attempts to improve the overall performance. In this scheme, the service providers do not have direct control over the caching decisions. As business critical applications are deployed over the Internet, it is becoming necessary for the service providers to ensure that the clients are receiving service at a satisfactory level to preserve brand equity and customer loyalty.

This issue has prompted a flurry of activities in the area of server-initiated caching and replication. One emerging technology in this area is *content-aware networking.* In content- aware networking, a new generation of routers specifically designed to address the unique requirements of Web traffic are used. These "content-based" routers have the ability to route TCP or UDP flows based on the URL and cookie in the payload. This ability allows the assignment of requests to one of various servers depending on the specific content requested. The previous generation of routers were able to route incoming packets based on the destination IP, protocol ID, and transport port number. Under this scenario the router cannot differentiate, for example, between a CGI script request or a streaming audio request, both of which use TCP port 80 but have very different *quality of service* (QoS) requirements. Content-based routers provide flexibility in defining policies for prioritizing traffic and for balancing load and content among servers so that Service Level Agreements can be met.

Content networking technologies can be deployed in several ways to improve a user's experience on the Internet. The first approach is the construction of virtual web servers, where a collection of servers or server clusters appear as one server. This approach can be generalized to arrive at the second approach where virtual web sites are constructed from one or more geographically dispersed data centers that appear as one domain name.

In the first approach [PaA98], a content router is interposed between the client and server, i.e., a request should pass through the content router. The content router examines the payload contained in the request packet and chooses an appropriate destination. In this mode of deployment, the content router is used as a load balancer for the back-end servers and the content router should be *all knowing* to make optimal decisions. In the second approach [TaD00], nameservers may be used to select the most appropriate "server site" depending on the geographical locations of the clients and servers and network and server loadings. The first approach has the disadvantage of lack of scalability and a single

point of failure and the second approach is inefficient in handling portions of a site (i.e., the hosting site cannot be partitioned on granularity of the documents, the whole site needs to be accessed or replicated).

With the deployment of mechanisms for wide-spread replication of content among geographically distributed servers, the issue of content aware request and data routing in wide-area networks is becoming very important. In this thesis, we examine the fundamental issues in content-aware networking: (a) Is content-aware routing superior to address-based routing with wide-area replica placements? (b) Under what conditions do performance differences exist between the two approaches? and (c) What attributes of the content should be considered for efficient wide-area content routing?

As part of this study, we propose a novel content-based routing mechanism called *Virtual Content Network* (VCN) for wide-area networks. The VCN is an overlay network built on top of the physical Internet framework. The outer edges of a VCN have *content-based edge routers* (CERs) and the interior has *content-based switching routers* (CSRs). The CERs may be either client site gateways or server site gateways. Either way the CERs are routers that can base their routing decisions on "higher" layer information instead of just the packet header information. The CERs are responsible for tagging packets at the edge of the VCN. The tags are used as proxies (or pointers) for the real content. The CSRs are switches that use the tags to steer packets through the VCN. The CERs and CSRs are connected by "pathways" or *content switched paths* (CSPs). The CSPs or pathways could be implemented using IP tunnels or dedicated/leased lines. Simulations are being performed based on the performance model that we have developed to compare the proposed content-based wide-area routing mechanism with a domain-name-based routing mechanism (also called DNS-based routing) which involves a routing scheme that intially requires the name resolvers to map a request to an specific IP and then route the request to that IP.

The following are some of the advantages of our architecture over existing approaches. Our architecture facilitates content routing in a global scale, which exploits the geographically distributed presence of resources and "time-of-day" properties of user behaviors. Scalability and lack of single point of failure are some of the other advantages. Further, our architecture enables the content to be handled at variable "granularity," i.e., portions of a site can be handled efficiently. Also, the content-based tags used in our routing scheme for content identification and access describe the resource characteristics of content and provide the foundation for developing an IP-less routing scheme.

In this thesis, we address several issues to make the VCN viable. In Chapter 2, we discuss related work and background information for content and DNS-based routing. Chapter 3 deals with the issues related to content routing and characterizing of Internet content. In Chapter 4, we present the overall architecture for protocol independent content switching, examine the individual components of the architecture, and describe our method of characterizing Internet content. In Chapter 5, we present a mathematical framework for our model and DNS-based routing scheme and in Chapter 6 we confirm the model's accuracy by experimenting with our simulator. Here, we also discuss the behaviour of our simulation model and compare the performance between our routing scheme and the DNS-based routing scheme. Finally, in Chapter 7, we discuss some applications that will benefit from content-based routing.

Chapter 2

Literature Review

In essence, content routing involves both locating and accessing content. Locating content may include content discovery on the network. Accessing content typically requires identifying a network path with the desired quality of service parameters and setting up sufficient resources along the path. Most existing approaches (e.g., domain name based routing) have devised separate schemes for locating and accessing content. Our content routing schemes provides a unified solution.

In this chapter, we examine a representative set of projects from the related literature on content-based routing approaches. These approaches can be divided into two classes. The first class represents highly distributed networked naming systems. These systems use some performance-based metrics to resolve a high-level content-based name to a machine name that will lead to the location of the content. The second class represents systems that "route" actual content requests considering network and server status. Currently, there are only few systems that fall into this category. These include scalable Web server clusters that are formed by interposing content switches between a cluster of high-performance servers and the clients.

2.1 Class I: Name-based Routing Models

The *Intentional Naming System* (INS) [AdS99] presents a resource discovery and routing system based on the description of the services. Applications create and advertise intentional names for each service they provide through an overlay network, of spanning tree topology, comprising *Intentional Name Resolvers* (INRs). The INRs locally cache the advertisements comprising the intentional names of the services and the IP addresses of the corresponding service providers. Any client requiring service will probe the INR using an intentional name of the service. The INR returns the IP addresses that correspond to the intentional name (called early binding) or tunnels the data received from the client to the location with the least routing metric (e.g., server status), called intentional anycast, depending on the service option required by the client. If the INR returns the IP address of a service provider, the client will tunnel the data directly to the service provider. However, this might lead to increased round trip time in case of cache misses. Another option (called intentional multicast) requires that the INR forwards the client data through the INR network to all the applications providing the service. This method may, however, lead to flooding of the network with the same service request. The intentional names are described using *name specifiers* whose main components consist of an attribute, to classify the service, and a value for the attribute. A hierarchical order of attribute-value pairs are used to identify different services and an unique identifier is used to identify the node advertising the service. A *Domain Space Resolver* (DSR) lists all the active INRs. For computation efficiency, the INRs are divided into subspaces such that each subspace resolves a subset of the entire set of intentional names.

The *Name Based Routing Protocol* (NBRP) [GrC01] presents a way of discovering and accessing content over the Internet based on the name of the content. Essentially, this scheme implements a highly scalable Internet-wide name resolver that takes into

consideration the current "performance" level of the target servers in performing the name resolution. The NBRP follows a two-phase approach similar to the current data access paradigm in the Internet. In the first phase, the name of the content is resolved to the "best" server and in the second phase the content is accessed by the client using the resolved address from the best server.

One of the key issues with name-based routing is that there are too many names even if only domain names are considered. The NBRP aggregates the names based on how the domains are connected to the Internet, i.e., an ISP content router may represent the content originating from the servers that are being served by the ISP. Depending on the replication patterns of a document, it could be part of several aggregates. A routing metric is calculated for each aggregate using the expected response latency of the content servers. The content routers running the NBRP also perform as conventional IP routers and may use IP routing information to choose among routes that appear equivalent at the content routing level.

An *Internet name resolution protocol* (INRP) [GrC01] is used to translate a client request to an INRP request, containing the server portion of the URL, which is then used to route the request to the best performing content server through a network of content routers. When a request reaches a content router that is nearest to the best performing content server, the router returns the address of the server back to the requesting client using the same path as the INRP request. If no such response is received within a predefined time, alternate paths are located.

2.2 Class II: Content Routing Models

The content routing method presented in [MiN00] uses a *Content Routing Protocol* (CRP) based on the URL decomposition of data objects. The CRP uses two different proto-

cols to create, propagate and update content routing information in cache server farms [ZhF98]. In this scheme, a URL is considered to identify the content in a tree-structured namespace. The CRP performs a systematic decomposition of the URL that traverses the namespace tree using the URL prefixes to identify each content uniquely. A *source information protocol* constructs and maintains the namespace tree and the *local content state protocol* is used to disseminate this information in a cache neighborhood. The namespace tree is formed by identifying the protocol, network location, and individual paths for the contents. The cache servers use the local content state protocol to exchange their list of URLs periodically by sending sequences of the tree depth and hash codes of the URL prefixes. Each cache maintains a URL forwarding table to represent the namespace tree and simply inserts the missing hash sequences, generated by the incremental hashing procedure used for the individual URL elements, in its forwarding table.

The *Content Addressable Network* (CAN) [RaF01] presents a highly distributed hash table architecture that can be implemented on the Internet. The main idea is to create a virtual coordinate space of d-dimensions and hash a key on to this space. The virtual coordinate space is divided into subspaces, and one or more CAN nodes are responsible for maintaining the information for their associated subspaces. Collectively, the CAN nodes form an overlay network that spans the coordinate space. Each CAN node holds routing information, IP address and coordinate space, for its neighbors.

A key is hashed to find the coordinate position in the virtual space. Using this position, the subspace and the corresponding CAN nodes that hold the relevant "value" are identified. Request packets are then sent towards the CAN nodes to retrieve the value. The request packets contain the coordinate positions of the destination CAN nodes and are routed using a greedy forwarding algorithm that attempts to minimize the "remaining distance" towards the destination at each step. The distance is measured in the coordinate space, i.e., the Euclidean distance between the current point and destination

point. Because logically adjacent CAN nodes need not be physically adjacent and the coordinate space measure of distance may not have any correlation with the "network" distance, this routing process is enhanced by adding the round trip time (RTT). In this scheme, each node determines the RTT to its neighbors and the routing process normalizes the *progress* (the reduction in coordinate distance achieved) by the RTT. This way the application level CAN routing can avoid long hops.

A Pastry system [RoD01] is an overlay network of nodes where each node is assigned a randomly generated 128-bit identifier to denote the node's position in a circular nodeid space ranging from 0 to $2^{128} - 1$. Given a message with a key, a Pastry node routes the message to a node whose identification number, called *nodeid*, is numerically closest to the key. The routing metric involved is the number of IP routing hops. The expected number of routing steps is O(log N), where N is the number of Pastry nodes in the network. At each routing step, a node message is forwarded to a node whose *nodeid* shares with the message key a given prefix that is at least 1 digit (or b bits, where b is a configuration parameter) longer than the prefix that the key shares with the forwarding node's *nodeid.*

Each Pastry node maintains a routing table, a neighborhood set and a leaf set. All entries at row n of the routing table contains the IP addresses of nodes whose *nodeid* shares a common prefix, usually the first d digits. Each entry of the routing table is chosen using the routing metric of number of hops and there will be a maximum of $2^b - 1$ entries at each row. The neighborhood set contains *nodeids* and the IP addresses of the nodes that are close to the local node in terms of the number of hops. The neighborhood set maintains locality properties of neighbor nodes. This information is used to update the routing table and the leaf set entries. The leaf set contains a set of nodes whose *nodeids* are numerically larger and smaller by some range than the *nodeid* of the node which maintains the leaf set.

Given a message with a key, a node first sees if the key falls within the range of *nodeid(s)* covered by its leaf set. If the key matches the leaf set's *nodeid* range, the message is forwarded to the node whose *nodeid* is closest to the key. If the key is not in the leaf set, then the routing table is checked to find a node whose *nodeid* shares a common prefix with the message key by at least one more digit.

The *Service Level Routing* [AnH99] framework provides a general architecture for accessing and discovering services and content on the Internet. The service level routing is performed by *Service Level Routers* (SLPs). The framework presented in [AnH99] is specific to two levels, although the architecture can be extended to more than two levels. In this framework, the SLRs are positioned at the borders of an autonomous system. The client-side SLRs are considered as the ingress routers and the server-side SLRs (i.e., the gateways to the server sites) are the egress routers. A client request first reaches the ingress router and it would use the service level routing information to tunnel the request to the appropriate egress router. In effect, the ingress SLR selects the server site that will service the client request. The gateway, or the egress SLR, uses possibly current information to select the actual server. The tunnels from the ingress SLR to the egress SLR form the first level and the tunnels from the egress SLR to the server host form the second level.

The servers handled by an egress SLR send their advertisements containing the load information to the egress SLR. The egress SLR, in turn, propagates these advertisements to the other SLRs. In this scheme, the higher level SLRs receive route information from multiple SLRs, (i.e., there will be more than one path for the virtual host).

In the SLR framework, each client request is mapped onto a flow. When the ingress router receives a flow request, it tunnels the request to the egress SLR that corresponds to the best performing server site. The routing metric is based on the load condition of each server as advertised by the next level of SLRs. The SLR with the lowest cost for

the server is chosen as the next hop for the request. The request may travel through several levels of tunnels until it reaches the egress SLR for the server site. The egress SLR that is also the gateway to the site holds the latest load information for the hosts of the site and tunnels the request to the appropriate host. The server host will terminate the tunnel and retrieve the original client request. The server host receiving the packet learns, from the datagram packet, the client address and also the virtual host address. Using these addresses, it tunnels the data back directly to the client.

The main idea of *Caching Neighborhood Protocol* (CNP) [ChL99] is to have an origin server and a number of proxy servers acting as caching servers, known as *cache representatives (C-Rep)*. The origin server is responsible for disseminating the information to the C-Reps in its neighborhood. A C-Rep can be a neighbor of one or more origin servers and may be either on the client or server-side. The origin server cannot be a proxy server and there is no specific rule about the size of the caching neighborhood. The size totally depends on the geographical locations of the origin server and the proxy servers.

The origin server is responsible for inviting or dismissing the neighboring proxy servers to act as C-Reps, performing load distribution among the C-Reps, managing information disseminating among the C-Reps and maintaining continuous communication with the C-Reps. The C-Reps attempt to serve a client's request originating from their neighborhood. When a C-Rep is unable to serve the request, it passes the request on to the next immediate C-Rep or the origin server.

The performance analysis shows that this architecture is extremely effective and most of the time the request is served at the C-Rep level and does not have to reach the origin server. The CNP provides an effective solution for cache coherency problem, increases the number of cache hits at the proxy servers, thus improving performance and resource utilization, maintains persist HTTP connections and performs load distribution to a large extent.

2.3 Simulation Techniques

Traditionally, there are two techniques that are most used to measure the performance of computer networks: (a) packet-level simulation and (b) fluid simulation. The packet-level simulation [AhD96] is a discrete event simulation technique that is used to speedup simulation of high speed, wide area packet networks. Such low level traffic representations can gain a tight control over the simulation events and provide an accurate measurement for the event lists. However, modeling each packet and each network element requires significant amount of computational resources. As the size and complexity of networks increase, this approach proves to be computationally expensive. An alternative to this is the fluid simulation technique [YaG97]. It simulates the network traffic as a continuous fluid flow rather than discrete packets. This technique is a tradeoff for simulation performance against accuracy [LiF01]. However, in our thesis the prime goal is to compare the performance efficiencies (i.e., amount of computation efforts) between the two routing schemes and not measure their computational accuracies. We have designed a simulator based on the fluid simulation technique and the models developed by [BuG98] and [LaZ84].

2.4 DNS-based Routing Techniques

The Domain Name System (DNS) is a distributed database maintained and stored across a hierarchy of nameservers. The nameservers in DNS are used to map mnemonic names to network locations. To the user, this is a service provided by the local or a remote host over the Internet. A DNS-based routing technique is a routing scheme that uses DNS name resolution service to map a request on to a network location and then routes the request to that location. This method is widely deployed over the Internet. The DNS uses

a manually delegated name space that is partitioned in domains and subdomains. Each domain is administered by an authoritative nameserver. The clients and the hosting servers will query their own local nameserver for name resolution. Name servers will either reply to a query or refer the query to an higher level nameserver. In this thesis, we define *query* as a request to resolve a domain name, *referral* as the query being forwarded or referred to a higher level nameserver that may be able to reply the query and *reply* is the response for query. The concepts and structure of the DNS can be found in [Pos94, Kes97]

A recent study [ShT01] shows that a server chosen through DNS-based name resolution is not necessarily close to a client. This stems from the fact that often the clients are not close to their local nameservers in terms of network hops. As a result, the hosting server chosen to be close to the client's local nameserver is not necessarily the closest to the client. This study also suggests that to improve the performance of the DNS protocol, it is necessary to carry additional information to identify the requesting client. Another study [JuS01], which explores the effect of varying degrees of cache sharing of DNS cache hits among clients, reveal that the performance of the DNS is not dependent on aggressive caching.

Currently, a number of content distribution overlay networks use the DNS for redirecting user requests and perform load balancing among the content hosting servers . A recent study [KrW01] identifies two types of Content Distribution Networks (CDNs) using the DNS redirection technique: full-site and partial-site content delivery. In the full-site content delivery, the content provider modifies its DNS nameserver to reflect the authoritative DNS server of its content delivery network (CDN) provider. In the partial-site content delivery, the content provider will reconstruct some of the embedded links in its web pages to redirect client requests to the servers of their CDN provider. In the first case, the entire content is served by a CDN, whereas in the second case,

some embedded objects are delivered by the CDN servers and the rest of the content is served by the content provider. The final results of the study, however, reveals that in average case situations, response times do not improve by using the DNS lookups for a new server. Yet another study [JoC01], reveals that the CDNs using DNS do not select the best server in a consistent manner.

The DNS-based routing technique, however, is the most popular technique used on the Internet. In this thesis, we choose this technique as a benchmark.

2.5 Solution Path

Content characterization, classification, and aggregation are key issues that impact content routing. Existing content routing schemes have several limitations in handling these issues. For example, the existing schemes use URLs, cookies and/or some attribute-based descriptions in conjunction with some load metrics to determine the "best" way to route the content request. This can be considered as a simple but limited approach for characterizing requests. The drawback of these approaches include the locational dependency of the characterization of the content. Instead, content characterization should be able to deduce the structural and semantical characteristics of the content which will help to identify advanced ways of describing the content. The description can then be used to locate and access the content. In this thesis, we present a method for characterizing content by identifying some key attributes of the content that are significant from content-based routing perspectives. These attributes are used to create groups of content such that each content group has some distinct set of properties. For example, documents having similar behaviour and being served by the same set of hosts may be grouped together. Each group comprises a distinct set of documents with similar characteristics. Each group is identified by a unique tag which is derived based on the attribute values

of the content objects in the group. The tags are then used to discover and access content. In Section 4.3.1, we identify some key attributes of the content that are significant from content-based routing perspective and in Section 4.3.2, we present a method for characterizing and tagging content.

Chapter 3

Issues in Content Aware Networking

3.1 Overview

Content-aware networking is defined as a technique that helps to make sophisticated routing and transport decisions based on the content that is being requested or transported. Network address derived routing information are also included while routing across geographically distributed wide area networks. By becoming content-aware, routing infrastructure will be able to accurately predict the resource requirements for handling the content transactions. This in turn leads to the development of advanced load balancing techniques and also improves the performance of the network through proper scheduling of network-based resources. However, to achieve this, a content-aware network should consider a number of issues: (a) content characteristics: identify the various properties or attributes of content that can used to efficiently describe a content, (b) content analysis: analysis of content attributes is required to determine the resources needed for delivery, (c) query analysis: exhaustive parsing of the user queries to identify the content requested and mapping the request to available resources, and (d) transparency: the entire process for content identification and request routing should be transparent to the end users.

These issues in turn lead to a number of practical problems like scalability and network efficiency. In this section we discuss some of the issues in content-aware networking and in the later sections we propose a solution for a new content-aware routing architecture.

3.2 Scalability

Most current deployments of content-aware networking techniques involve a content router or a group of content routers acting as a front-end load balancer for the back-end server clusters. Using such techniques, content delivery systems try to cope with the challenge of delivering content to client with consistent and high service level. However, scaling remains a challenge as the number of Internet content and the demands of the users grow. In order to perform web traffic management suitably, the content routers or content switches need to be *all knowing* about their back-end servers to ensure the best performance and total service availability. Any increase in the number of Internet components (e.g., number of distinct content elements) will lead to performance problems for the content routers.

Moreover, content providers often use the services of a CDN provider to create multiple data centers to allow serving content locally to the clients. This requires the content providers to reconstruct the links in their web pages to redirect client requests to the servers of their CDN providers. In practice, this has lead to the development of a highly distributed name resolution service. However, this also restricts the content providers to use the services of a single CDN provider. This in turn restricts the network reachability for the content providers unless different CDN providers inter-operate with one another.

We address the problem of increasing content elements by creating content aggregations or groups of contents and considering each group as an individual content entity. Content-derived tags are used to identify each content group uniquely within the routing

domain. The content-derived tag is created based on specific attribute values of the contents in a group. The routing mechanism uses the tags to discover and access content. The content-to-tags bindings are maintained by the routers using an active and passive mode. In the active mode, bindings for more popular content are placed closest to the clients and bindings for less popular tags are placed further away from the client. As the popularity of a content increases, the tag binding for the content is pulled closer to the client. In the passive mode, as a content's popularity decreases, its tag bindings are pushed away from the clients. This way, the routers does not have to know about all the content-to-tag bindings at all times. The increase in the number of contents is handled by creating new content groups or appending new content to existing groups. This way, the number of tags is much lower than the total number of contents available. Moreover, a generalized tagging scheme based on content attribute values allow content providers to use the services of multiple CDN providers thus allowing faster delivery of content.

3.3 Efficiency

Content routing is not a simple pattern matching task. In the current scenario, a content router intercepts a client request destined for server clusters and distributes them accordingly depending on their content. In order to do that, the router must first parse the packet contents to locate a possible next hop and also complete a TCP handshake for the request with the client and the possible next hop. This is a time consuming job and if such a policy is to be followed at all the routers through which the request passes, then the overall time taken to route a request is very large. As the number of routing elements in the network increases this overhead also increases. Keeping this in mind, the ultimate goal of this thesis is to develop an encapsulation protocol that operates upon the network traffic to generate a digital signature at a router nearest to the client,

depending on which packets can be forwarded by other routers without having to look into the the packet contents repeatedly. Such a technique will certainly reduce the total overhead of the network. Moreover, by making the digital signatures content-aware (i.e., the signatures are derived based on the attribute values of the content) and considering global routing information, it is possible to facilitate load balancing across the server access path using the signatures.

3.4 Content Characterization Problem

The massive popularity of the Internet is mainly due to its ability for content distribution and exchange between geographically dispersed hosts. Usually, most of the popular traffic or the increased demand for some popular content is temporary: For example, sudden flash news being served by web sites or very large files being exchanged by peer-to-peer applications. Such instances often result in overwhelming consumption of resources and can prompt congestion in the network. To handle the sudden increase in demand for content, most content providers create multiple replicas of their content at geographically distributed locations (i.e., move the content closer to the clients). Then the problem that arises is to discover and access the content replica that is the best from the point of view of the client (i.e. the content routing problem).

In its current form, content routing examines URLs and cookies to determine the destination for a request for data. URLs and cookies are a very simplistic way of describing the content. In particular, they use a combination of network and application level addresses to describe the request for the content. The drawbacks of these approaches include the locational dependency of the characterizations of the content. In contrast to this, we argue that a robust content routing network should also have prior exposure to the resource characteristics of the content which will enable a router to make optimal

routing decisions. Our reason for such an argument is as follows.

The process of mapping document requests to server names can be considered as a load balancing problem. However, in practice, load balancing can be performed not only across the servers but also across the networks paths leading to the servers. But, in order to perform load balancing on the network paths, we need to know the "load" that is imposed by different content requests. Therefore, we need to know about the resource requirement for each content request. In other words, we need to have adequate knowledge about the resource requirements for the documents that are being requested. Once the resource requirements of documents are available to the routers, they can make optimal decisions about the path leading to a server. For example, by knowing the bandwidth required for delivery of a content, the router will be able to steer a request around the congestion points (e.g., bandwidth starved links) and direct it along a path that has, or will have in near future, enough bandwidth to deliver the content requested. Such a path will not always be the shortest path but it will guarantee continuous flow of content and thereby deliver the content in the shortest possible time. The primary motivation to make the routing network content-aware is that by knowing the resource requirements for contents, a priori, it is possible to disperse the network load across the entire routing domain, instead of clustering them in one section of the network which will act as bottleneck for rest of the network. However, in order to make the network content-aware it is necessary to learn more than the locational characterisitics of content. In Section 4.3.1, we examine a list of possible characterisitics of content and describe how content characterization can be used to create a content-profile that can adequately describe the resource requirements of content.

Chapter 4

Protocol Independent Content Switching Architecture

4.1 Overall Architecture

In this chapter, we introduce a *Protocol Independent Content Switching* (PICS) architecture that is suitable for content aware networking in very-large scale geographically distributed networks. In the PICS architecture, several client and server sites are interconnected through a *Virtual Content Network* (VCN). The client sites contain the eventual consumers of the content that is managed by the VCN. The VCNs themselves can be networked to build larger networks. The client and server sites connect to the VCN using gateways that are content routers. Because the content is first examined by these routers, and they are at the edge of the VCN, we refer to these routers as the *content edge routers* (CERs). In our architecture, a client or server site can connect via multiple CERs to the VCN. Allowing multiple CERs per site precludes a single point of failure and provides an opportunity for enhanced load balancing among the content hosting servers. The CERs encapsulate the packets using a content header. The content

header contains a content-based tag that is used to uniquely identify a content within the VCN. The VCN core has content-based switching routers (CSRs) to steer the content requests from the ingress (i.e., client side) CER to the egress (i.e., server side) CER based on the tags in the content header. The CSRs are simple switches that support a single forwarding component (i.e., algorithm for tag-based forwarding). The CER, on the other hand, are also responsible for characterization and classification of content.

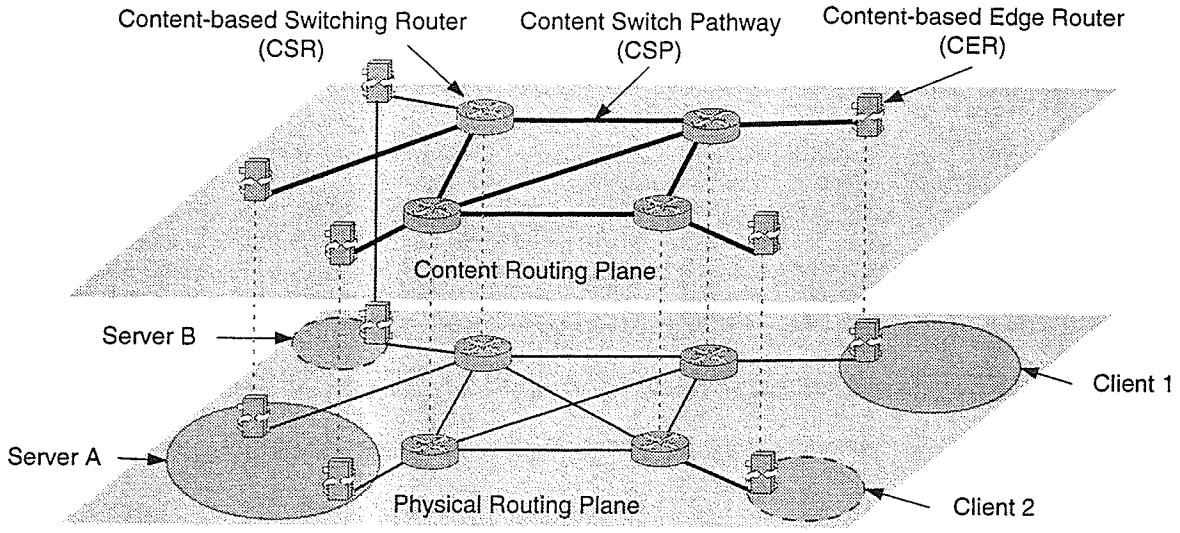

Figure 4.1: An example deployment of a virtual content network.

An example deployment of PICS is illustrated in Figure (4.1). It shows two client sites `Client-1` and `Client-2` connecting to the server sites `Server-A` and `Server-B` through a VCN. `Server-A` is connected to the VCN through two CERs whereas `Server-B` is connected through a single CER. Each of the client sites, `Client-1` and `Client-2` are

also connected to the VCN through a single CER. The VCN consists of several CSRs that are connected by sequences of *content switched paths* (CSPs) or "pathways." The CSPs interconnect the CSRs and also connect the CERs to the CSRs.

The CSPs are virtual paths that are created when the VCN is initialized. As client and server sites enter and depart the VCN, the corresponding CSPs may be created and removed, respectively. The CSPs are used to "forward route" the content requests to the appropriate server site and to "reverse route" the content back to the "requesting" client site. There are various possible ways of handling the forward and reverse routing processes as described later in this section.

4.2 Functional Components

4.2.1 Server Sites

The server sites hold the content that is requested by the clients. The simplest server site can be a CER and an attached server that serves the corresponding content. In the general case, multiple CERs can act as front-ends to the servers that are in the site. The servers can be either origin servers or surrogate servers. An *origin server* holds the original copy of the content, i.e., if there are no replication of content this server will have the only copy. A *surrogate server* is a server that "rents" its resources so that content from origin servers can be replicated on it. The owner of the origin server that is referred to as the *content provider* is responsible for "paying" the rent for the surrogate server. By using a computational economy [AnK98] based scheme for replicating content among the origin and surrogate servers it is possible to increase the server side resources for content that are popular and reduce the dedicated resources as the popularity decreases.

Therefore, in the PICS architecture, the requested content can be delivered either by

an origin server or a surrogate server. The VCN does not differentiate between surrogate and origin servers. The server selection algorithm is built into the forward routing process.

4.2.2 Client Sites

The client sites have the clients that are requesting content. Typically a client site may have a single CER connecting it with the VCN. However, in some situations it may be useful to have multiple CERs. Some advantages of such configurations include lack of single point of failure and opportunity for traffic shaping within the VCN. Depending on the local routing policies, the CERs can be allocated traffic according to the *quality of service* (QoS) considerations.

4.2.3 Virtual Content Network

The VCN is a graph with the CERs and CSRs as the nodes and the CSPs as the edges. The VCN provides a *Content Delivery Network* [DaC01] for the server sites to efficiently distribute their content. The VCNs may be deployed using a "peer model" similar to that used in Virtual Private Networks [Bro99]. This model assumes the existence of VCN service providers who host the service. The client and server sites that need "speedy" content delivery would subscribe to the VCN by having content routing enabled gateways (i.e., CERs as gateways) and by connecting such gateways to the VCN core. The different VCN service providers could be connected among themselves to provide a larger virtual network.

Content Edge Router (CER)

As mentioned above, the CERs are at the "edges" of the VCN and the client and server sites. When a request for content is generated from a client site, it reaches the client side CER. The CER parses the packet carrying the request and if the content is

handled by the VCN, then it identifies the content and tag it. The tag is generated by a combination of the content derived and the policy-based information. Figure 4.2 illustrates the overall tagging process in the CERs. The process is discussed in detail in later part of this chapter.

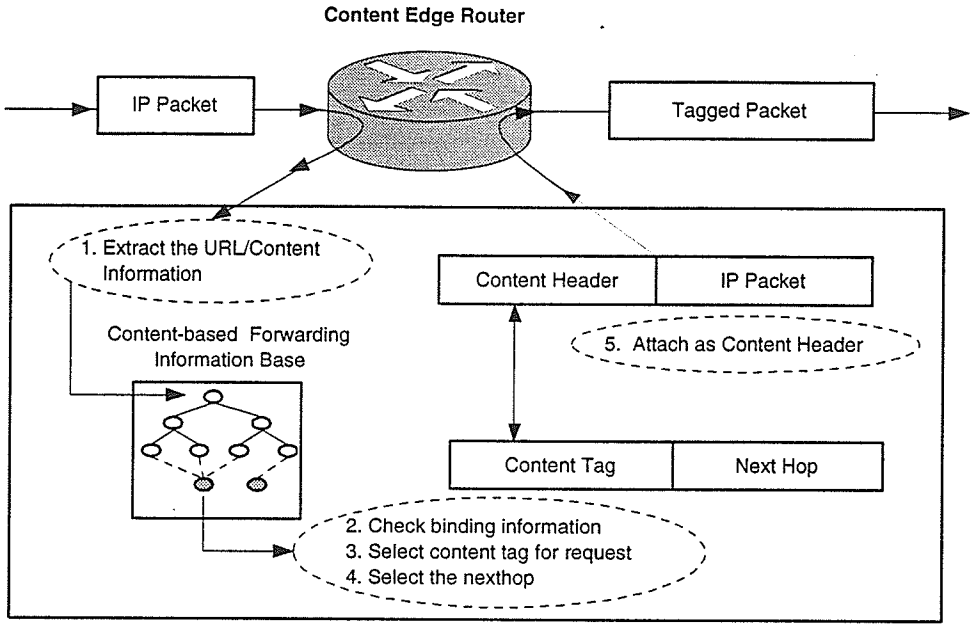

Figure 4.2: The content tagging process in a CER.

A CER can be connected by multiple CSPs to the core of the VCN. Therefore, for the content that is handled by the VCN, the CERs should make initial routing decisions as to which core CSR should handle the request next. Each CER has a content-based routing table that is used by the CER for determining the routes. The information can be disseminated across the VCN by either flooding the routing information towards all CERs and maintaining a consistent route image throughout the entire VCN or operate on an on-demand basis. The on-demand approach provides the high activity CERs with more up-to-date route information.

Content-based Switching Router (CSR)

The CSRs form the core of the VCN. Similar to the CERs, the CSRs also maintain content-based routing tables. However, the information contained in the CSR routing tables are different than the information found in the CER routing tables. The CER routing tables also contain the content value (e.g., the URL or cookie value) in addition to the tags generated from the content. The content value is necessary for the CER to perform the content-to-tag mapping. The CSRs use the content-based routing tables to steer the requests towards the appropriate server side CER. Figure 4.3 shows the routing process in the CSRs.

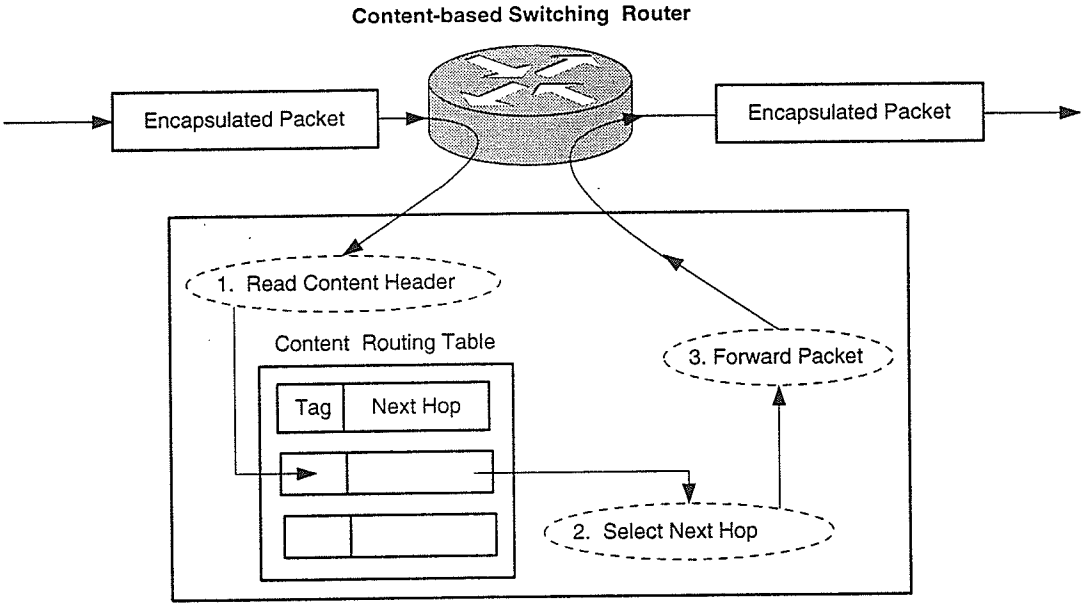

Figure 4.3: The content-based routing process in a CSR.

Content-based Switched Path (CSP)

The CSPs interconnect the CERs and CSRs to form the VCN. A CSP is similar to a *label switched path* (LSP) in the *Multi-Protocol Label Switching* (MPLS) [DaR00]. CSPs can be implemented using frame relay circuits, ATM circuits, IP-in-IP tunnels, *generic routing encapsulation* (GRE) tunnels [FaL00], or IPSec [KeA98] tunnels.

IP-in-IP and GRE tunnels can be subject to data spoofing. Some malicious router that is not an end-point of the tunnel could inject packets into the tunnel. Additional packet filters may be used to solve this problem resulting in increased overhead and complexity. Using IPSec tunnels is another way to solve this problem. However, the "key" management used by IPSec to authenticate packets adds cost to the data transmission.

To provide QoS, all tunneling mechanisms rely on services such as IP *differentiated services* (Diff-Serv) [BlB98, Bla00]. When leased lines, frame relay circuits, or ATM circuits are used, stricter guarantees can be given for QoS than those possible with tunnel-based CSPs. One of the advantages of tunnel-based CSPs is that they can be used more widely. If stricter QoS guarantees are essential for the data traffic, then the routing algorithms may constrain such traffic to the appropriate portions of the network.

4.3 Content Characterization and Classification for Content-based Routing

4.3.1 Content Characterization Solution

Content characterization is a process that identifies the key attributes of content which can be used to generate an accurate description of the content and its resource requirements from the perspective of content-based routing and delivery mechanism. The primary motivation is to use this description to discover and access content. Our preliminary studies for characterizing and classifying content have shown that a deeper understanding of this issue will help to develop advanced description methods that will significantly impact the content tagging schemes. This facilitates the development of a flexible content-aware networking scheme. In this section, we identify a possible list of content attributes that can be used to profile a content from the perspective of content-based routing. In

Section 4.3.2, we use some of the attributes identified here to describe a content classification and a tagging strategy that we use in our content-based routing scheme.

To distinguish a content uniquely, it is necessary to understand the structure and semantics of data. The structural attributes and their values are directly dependent on the content while the semantic attributes relate to the behaviour of the application that accesses the content, i.e., the intended usage of the data by the requesting application. For example, an application may use the FTP protocol to download a large video clip for off-line viewing. Another application may access the same video clip for online viewing through a "media player" that requires real-time data transfer capabilities with QoS attributes. The routing schemes should be able to deduce this difference in service.

In our proposed content-based routing architecture, we intend to create a content profile a priori to the routing process. Such a profile is used to locate the content and also allows the routing protocol to infer those characteristics that directly affects the content delivery mechanism (e.g., bandwidth required for delivery). In a delivery network, such a profile should also surmise a content provider's relative status with respect to the other providers using the CDN services. Again, a content routing process, in effect, is triggered by an application agent requesting the content. At the time of request initiation, the agent specificies some rules for content access, like the QoS parameters like Diff-Serv. The routing system interfaced with the agent should ensure the requested QoS while delivering the content. To ensure that all these requirements of the content are met, in our characterization scheme we group the content attributes in two distinct classes. The first class of attribute values are known prior to the routing process and can be used to create a content description, a priori, which is then used to discover the content on a network. The second class of attributes are initialized only at the time a request for the content is submitted and is used for accessing the content. A combination of these attibutes will decide the network resources that should be allocated for a request. More

specifically, these classes can be defined as follows:

Structural attributes: These are the properties of the content that impact the amount of computation and/or communication resources needed for handling the content and they are *invariant* of the requesting application or the user.

Semantical attributes: These are the properties of the content that impact the amount of computation and/or communication resources needed for handling the content and they are *dependent* on the requesting application or the user. Their values depend on various conditions that are imposed at the time the content is being requested.

Table 4.1 shows a representative set of the structural attributes that are relevant for describing content. We have further classified structural attributes into four categories, where each category is used to define a definite set of structural properties for the content. In our content-based routing, some values from each category are used to describe the content. The *physical* attributes describe the properties of the content that are shared between the file system and the network. The *name-based* attributes are used to label the content for identification and location purposes. The names may be chosen in various ways. Some names may have components of the location embedded in them, e.g., URLs. While other names such as filenames may be arbitrarily chosen. The *end data type* attributes refer to the content formats. Although these formats are relevant to the end applications, these formats are independent of the applications and may have impact on the way the content could be handled by the network. The *popularity* attributes gives the spatial and temporal "interest" for the given content among the user population. The demand imposed by the content on the network may be indirectly depend on the popularity attributes. The spatial popularity refers to the *current demand* for the content in a specific portion of the VCN. The temporal popularity refers to the *global demand*

across the VCN for the content over a specific time period.

Table 4.1: Structural Attributes

Physical	size, version, modification date, ownership permissions, copyrights, author
Name-based	filename, location name, URL, origin server name
End data type	HTML, cookies, scripts, audio/video clips, text
Popularity	temporal, spatial

Table 4.2 shows a representative set of semantic attributes. The semantic attributes can be further classified into four different categories with each category describing a definite class of resource requirements for the content and the application requesting the content. The *access* attributes define the requirements of the applications that access the content. These attributes may determine some of the service measures associated with or required by the content. For example, whether a content is of streaming type and its bit-rate characteristics may determine its bandwidth requirements. The *quality of service* (QoS) attributes defines the amount of network resources needed to support the application's access of the content. While the access attributes define an application's "intentions" from the application perspective, the QoS attributes define the same intentions from the network perspective. The *end data type* attributes are used to differentiate static and dynamic contents. Dynamic content is assembled on-demand based on inputs from the user or the accessing application.

The main idea to group the contents using their variable and non-variable properties is due to our intention to create aggregates (or groups) of contents with identifiable properties. A combination of some of the variable and non-variable attributes is used to create the aggregations and enable our content-based routing system to learn about

Table 4.2: Supporting Attributes

Access	duration, streaming, non-streaming, variable-bit rate, constant-bit rate, adaptive-bit rate
Quality of service	bandwidth, delay, loss tolerance
Document Type	static, dynamic

the contents and their resource requirements. We call this a learning process for our routing model. Some of the variable content attributes are used to relate the contents to the forwarding algorithm built into our content-based routing scheme to implement suitable traffic shaping policies. Using such a scheme, the routing process in the content aware network is partially accomplished by setting up virtual path segments during the learning process. The remaining portion of the routing process may be done by switching the content requests and replies across the virtual path segments using some "content-based tag" that is generated from the content-aggregate attribute values.

4.3.2 Content Classification

Content classification is a process of grouping documents into classes such that the expected resource requirements for handling the documents within a class are similar. We refer to these classes as *Content Equivalence Classes* (CECs). The resource requirements for handling documents include the resources used for storage and processing purposes and resources used for transporting purposes. The storage and processing resources are determined by the structural attributes of the content while the resources for transporting the content are determined by the semantic attributes of the content. The structural and semantic attributes are discussed in Section 4.3.1. A CEC will inherit the attribute

values from its constituent documents. A CEC is identified by a tag that is derived from the attribute values of the CEC. This tag is used for discovering and accessing each individual document that constitutes the CEC.

Our motivation of using content groups and content-derived tags is to compress the namespace that is used by the routing system for locating and accessing content. In traditional routing schemes, the routing is performed based on addresses with a pre-defined format and length, i.e., the size of the address space is fixed. These routing schemes are, however, completely location dependent. The main idea behind content-based routing is to liberate content access and discovery from the "original" location that holds or owns the content. In general, to achieve this the content-routing schemes base their routing on a set of attributes that describe the content. Mostly, some form of content-name that is not dependent on the actual location is used for content-routing (e.g., URL of the content). One of the major challenges with a routing scheme based on content-naming is that the namespace can be *very* large. However, in practice, the number of distinct content names that are handled by a content-based routing system is a very small fraction of the total namespace. In our content-based routing scheme, we use the attribute values of the CECs to generate fixed-length tags. By grouping multiple contents in a single CEC, we actually bind the specifications of multiple contents in one description (i.e., one name can be used to identify multiple contents). Since each CEC has distinct attribute values, the tags will always be unique within the VCN, and also the total number of tags will be much less than the actual number of individual documents. Further, the tag can be used by the content routers to infer about a content i.e., the routers do not have to be all knowing about the contents for making an optimal routing decision.

Our content classification process includes two phases. In the first phase, the resource requirements for storage and processing are identified and the CECs are created. This

process is usually performed when a content is first introduced in the VCN. It can also be repeated when the configuration of the VCN changes. During this phase, a portion of the tag is derived that is used to discover content. We call this portion of the tag as the primary tag. The remaining portion of the tag, called the secondary tag, is derived during the second phase of the classification process. The second phase identifies the values for the semantic attributes of content at the time a client submits a request to the VCN. The semantic attribute values are used to select an appropriate path from the client to the server. The secondary tags may also be used by the CERs/CSRs to partition the set of all possible requests for the same content into disjoint subsets. From a forwarding point of view, all requests within a subset will be treated in the same way by the content routers in the VCN.

The first phase of content classification process can be performed in two ways similar to the differentiated services: (a) *behavior aggregate* (BA) and (b) *multifield* (MF). In the BA, the content owner is cognizant of the content behavior (e.g., demand) of the content to specify how the content should be handled. Typically, the content owners would request the VCN to handle their content as a class by itself. The tag assignment, however, is performed by the VCN. For example, a suitable pricing scheme and service level agreements for the content management service that is beyond the scope of this document may be used to prevent a customer from denying services to other customers. In the MF, the content owner is not cognizant of the content behavior but wishes to use the VCN for faster content delivery. In this case, the VCN performs content classification using the scheme described below.

In our content-based routing scheme, during the first phase, the CERs use the structural attributes to describe a content's storage and processing resource requirements. These are (a) content size that indicates the storage space required at the surrogate servers, (b) the type of content e.g., streaming or non-streaming, audio/video or text,

(c) popularity of the document containing the content (i.e., the number of times the document was requested with respect to other documents available in the VCN), and (d) the minimum network-based resources (e.g., bandwidth) and the time required to deliver a content (this is considered only in case of streaming contents). These attributes provide the most generic description about what is being transported across the network. Statistical analysis of the Internet traffic provides all the attribute information required for classification of content.

We may think about the first phase as a procedure of partitioning the set of all possible documents that the VCN can handle into a finite number of disjoint subsets, called CECs, where each subset holds a finite number of documents of similar type and having similar storage space requirements. Storage space requirement gives the measure of how much storage space should be allocated to contents with respect to their popularity within the routing domain. We group together all contents that require similar storage space and are of similar type. Each CEC is identified by a tag that is unique within the VCN. Client requests are marked by the tag which is used to identify the CEC (and in turn the hosting server) holding the content requested by the client. From the routing point of view, all client requests that have the same tag are handled in the same way by a CER or CSR as compared to other requests with different tags (e.g., requests with similar tags require similar resources).

Next, we describe the process for creating the CECs. We assume that the VCN being an overlay network, will cover portions of several *autonomous systems* (AS) which make up the Internet. Our classification algorithm uses the popularity of the contents within each AS as a rank estimation factor to measure the amount of storage resources required by each document. This is done by constructing a document-AS matrix A where each row of the matrix corresponds to a document hosted by the VCN and each column corresponds to an AS which is part of the VCN. The matrix elements represent

the relative storage space required for each document as per its popularity level within each AS, i.e.,

$$\text{matrix } A^{mxn} = [a_{ij}]$$

where m denotes the number of documents hosted by the VCN and n denotes the number of AS(s) which are part of the VCN and a_{ij} is the ratio and popularity of the document i in AS j such that $0 < i \leq m$ and $0 < j \leq n$. We call a_{ij} as the *relative weight* for document i in AS j. In effect, the relative weight scale will determine the storage space requirements of each content relative to other contents in the VCN. In practice, local and global weights are applied to increase/decrease the popularity of documents within each AS. Specifically, we write

$$a_{ij} = \frac{s_{i,t}/\overline{s}_t}{L(i,j) \times G(i)}$$

where $s_{i,t}$ is the size of the document i and which is of type t, $\overline{s}_t$ is the approximate size of all documents which are of same type t as the document i, $L(i,j)$ is the local weight representing the popularity of the document i in AS j, and $G(i)$ is the global weight representing the subscription level of the document i (more specifically, the subscription level of the content-provider hosting the document i) within the VCN. The value $\overline{s}_t$ is tuned by the VCN administrator while configuring the VCN at the setup time. This parameter is used mainly to differentiate between the different types of contents, like an HTML document or a movie document, such that sufficient storage space is allocated to a document in relation to other documents of the same type.

Next, all documents of similar content type and having similar a_{ij} values are grouped together to create a single CEC. At any given time a particular document can be present in only one CEC. Figure 4.4 shows an example for creating the CECs from a document-AS matrix. In the example, we assume that for the documents C_i $(0 < i \leq 8)$ we know their size (s_i), content type (t), popularity in AS j $(L(i,j)$ where, $0 < j \leq 3)$, and the

subscription level ($G(i)$) of the providers hosting the C_i. As shown in Figure 4.4(a), using these parameters we calculate the matrix A. Next, using the matrix elements a_{ij} we sort the contents for each AS, as shown in Figure 4.4(b). Then we create the CECs from the sorted content lists, where we put all the contents which are most popular for all AS in one group (i.e., CEC), as in Figure 4.4(c). The groups CEC_1 and CEC_2 can further be broken down based on the content types of their constituent documents. For example, in class CEC_2 if C_1 and C_3 are of type streaming movie while C_2 is a image, we can group C_1 and C_3 to form the class CEC_2 and create a new class CEC_5 to hold C_2. Using this technique, we can create content classes at variable granularity. Figure 4.5 illustrates the grouping of the content.

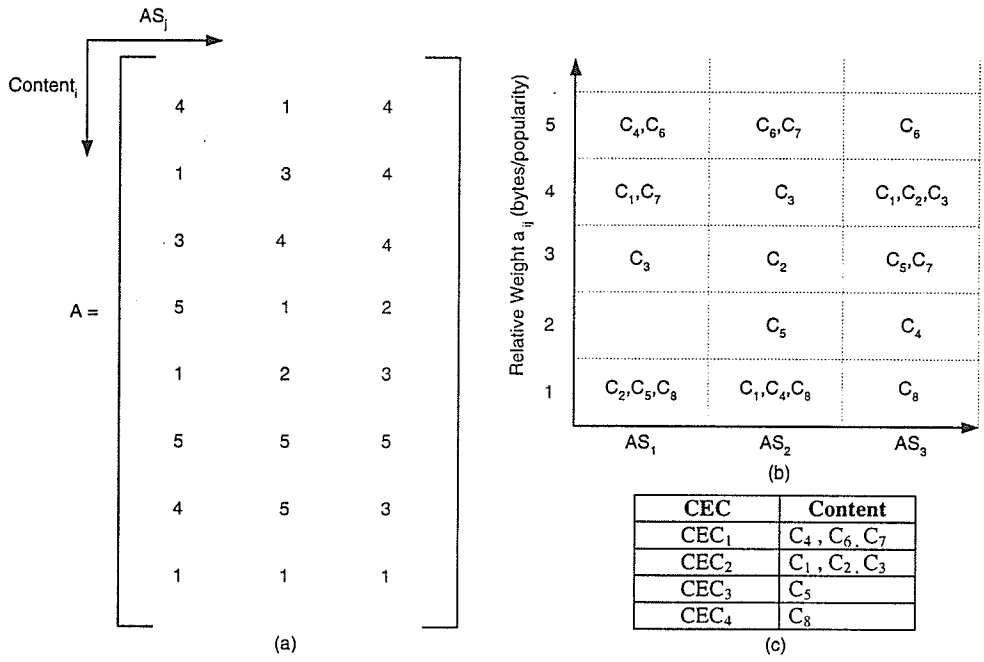

Figure 4.4: (a) An example document-AS matrix with 8 documents and 3 autonomous systems. Each matrix element A[i,j] called *relative weight* is a relative measurement of popularity for the document i in AS j. (b) All the documents are sorted according to their relative weights for each AS (c) Grouping of documents with similar relative weights.

As mentioned earlier, all the CERs in the VCN create groups among themselves. The

matrix A is computed and maintained by each group of CERs. Each server-side CER informs each of the CER groups about the new contents that are introduced to the VCN. The CER groups generate the matrix A, create the CECs and their corresponding tags. This method reduces the overall processing load for each of the CERs and distribute the load across the network. The namespace tree comprising the content-to-tag bindings are maintained by each of the CER groups. Moreover, at times of failure each CER group acts as a backup for other CER groups.

It should be noted that until now we were calculating the relative weight (a_{ij}) values on a per document basis. This becomes a computationally intensive process for a very large number of documents. Instead, we first create groups of documents and calculate the relative weights for each group. The CECs are composed of these groups of documents. This reduces the size of the matrix A and also improves the processing efficiency. For example, we can create a group of documents using the URL prefix http://www.abc.com/sports. This URL prefix identifies a directory containing some number of documents. We consider all the documents in this directory as a single content entity during content classification. The sum of the sizes of each of the documents in the directory is the total size of the content entity and an average of the total popularity of all the documents in the directory for a particular AS represents the local weight for the content entity.

We introduce a pair of tunable parameters while creating the CECs: low (l_b) and high (h_b) bounds to limit the total amount storage space required for each CEC. The total storage space for a CEC is the sum of the storage spaces for each data objects in the CEC. Higher bound h_b depicts the maximum amount of storage capacity that can be allocated to a CEC and lower bound l_b depicts the minimum amount of storage resources that should be allocated for a CEC. The l_b and h_b values are chosen suitably by the VCN administrator at the time of VCN setup for proper functioning of the routing

protocol. Surpassing h_b may create a single CEC that contains all the documents hosted by the VCN. Instead, it is more efficient to create smaller content groups and distribute the groups across the VCN. This helps in (a) managing the available storage space at the surrogates more efficiently, (b) store the contents as per their popularity in different sections of the network, and (c) allow an even distribution of the client requests across the VCN. The replication algorithm, which is beyond the scope of this thesis, will use the CECs to create copies of the contents (i.e., all contents in a CEC are copied at the same surrogates). The number of copies for each CEC and their storage location will, however, be determined by the relative weight (i.e., a_{ij}) of the CEC.

In the second phase, the semantic attributes are used to identify the delivery requirements that are imposed at the time a client requests a content. The values of the semantic attributes are used to create the secondary tags. The primary and secondary tags are together used to tag a request packet. The semantic attibutes that are used in our scheme are (a) the QoS attribute and the (b) access attributes. The QoS attributes can be explicitly mentioned by a service level agreement like Diff-Serv. The intended usage of the requesting application are also included while creating the secondary tags. This is identified by the protocol number mentioned in the IP header of the request packets. When a client-side CER (ingress CER) receives a request packet, it will parse the packet to identify the content being requested and also identify any specific resource requirements that are explicitly mentioned in the packet. An examination of the TOS (type of service) field, also called DS (Diff-Serv)field, [NiB98] reveals any per-hop behaviour (PHB) specified for the packet. In case no PHB is mentioned, the protocol field identifies the protocol being used by the requesting application. The ingress CER uses a pre-defined code to describe the protocol in the secondary tag. We use a 8-bit long secondary tag to define a PHB or any other specific services being requested.

```
GroupObj(numDoc m, numAS n):
/* Executed by a CER group for m documents introduced to VCN */
  create matrix A[m x (n+1)]
  for i= 0,m
      for j= 0,n
          A[i,j] = s_Content_i,t / s̄_t  /  L(Content_i,AS_j) × G(Ccontent_i)
      endfor
      A[i,j] = false
  endfor
  cntCEC = 0
  for i= 0,m
      if(A[i,n] == false)
          create new CEC set CEC_cntCEC → { null } and initialize N=0
          add Content_i to the set CEC_cntCEC
          A[i,n] = true
          for j= (i+1),m
              for k= 0,n
                  if(A[j,n] = false and A[j,k] = A[i,0] and (l_b ≤ Σ_{i=1}^{N} s_Content_i ≤ h_b))
                      if(Type(Content_j) = Type(Content_i))
                          add Content_j to the set CEC_cntCEC and increament N by 1
                          A[j,n] = true
                      endif
                  endif
              endfor
          endfor
          cntCEC = cntCEC+1
      endif
  endfor
```

Figure 4.5: CEC creation algorithm.

4.4 Content-based Tags

4.4.1 Temporary Tags

Most of the information that are used for content classification and tag generation are obtained from the statistical analysis of the VCN traffic. The traffic analysis process is handled distributively by the CER groups in the VCN. This analysis will generate a

detailed report on the usage and characteristics of content attributes. This is, however, a time consuming process and is repeated infrequently. As a result, when a new content is introduced to the VCN, it may not be assigned a tag based on its attributes immediately. This is mainly because no information will be available for the new content from the last traffic analysis. To cope with this problem, we introduce *temporary* tags.

Each group of CER will maintain a pool of free tags (i.e., tags with no content bindings). When a new content is introduced to a CER, it chooses a free tag maintained by its group of CERs and creates a new content-name to tag binding using the free tag. The tag chosen from the pool of free tags to create a new content to tag binding is called the temporary tag. More specifically, the primary portion of the tag will contain the temporary tag and the secondary tag will be created when a request for the content is submitted to the VCN. This content name to temporary tag binding is pushed to all the CER groups in the VCN. All client requests for the new content will be routed based on the temporary tag assigned to the content. Once a temporary tag is assigned to a new content, the VCN will start collecting traffic information for the new content. During the next cycle of content classification process a content-based tag will be assigned to the content with a temporary tag. The content-based tag is derived from the content's new attribute values. The temporary tag is returned to the free pool of tags. Figure 4.6 shows the sequence in which the temporary and the content-based tags are generated and managed. The forwarding algorithm will always give priority to all contents with content-based tags over the contents with temporary tags.

4.4.2 Tag Format

As mentioned earlier, the content tags are created based on the values of the CEC attributes and have a primary and a secondary portion. The primary portion is created

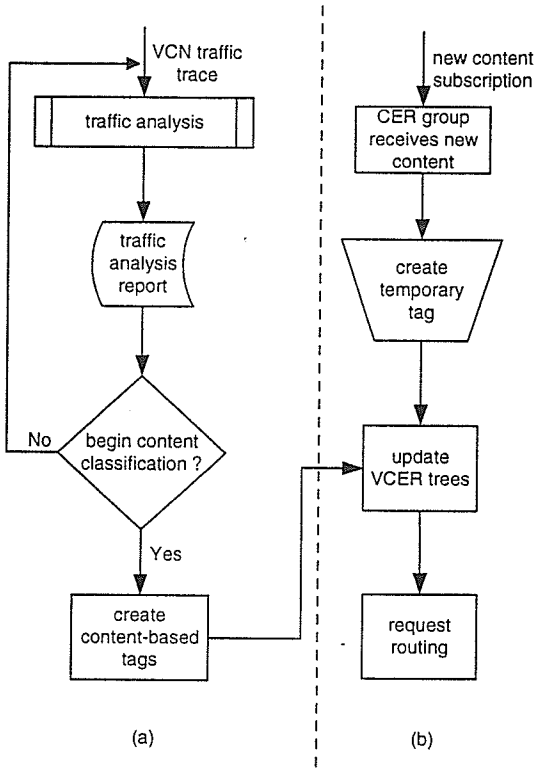

Figure 4.6: (a) This phase is run at over periodic intervals to create the content-based tags. (b) This phase is executed when a new content is introduced to the VCN.

during phase-I of the CEC creation process by the server-side CERs. The secondary portion of the tag is created during phase-II by the client-side CERs. The primary portion is used to locate the content while the secondary portion is used for accessing the content. In effect, a combination of the primary and the secondary tag portions will be used to select a suitable path from the client to an appropriate hosting site. A 32-bit tag is attached as a *content header* to encapsulate the IP packets before pushing the packets into the VCN. The encapsulation is somewhat similar to that used by MPLS. Those links which cannot include the content header in the link layer header (e.g. Ethernet), the content header is carried in a shim header between the link and network layers (layer 2 and layer 3 of the OSI model). For links like ATM and Frame Relay, the content header

can be carried inside the layer 2 (i.e., the link layer). The format of the content header is shown Figure 4.7.

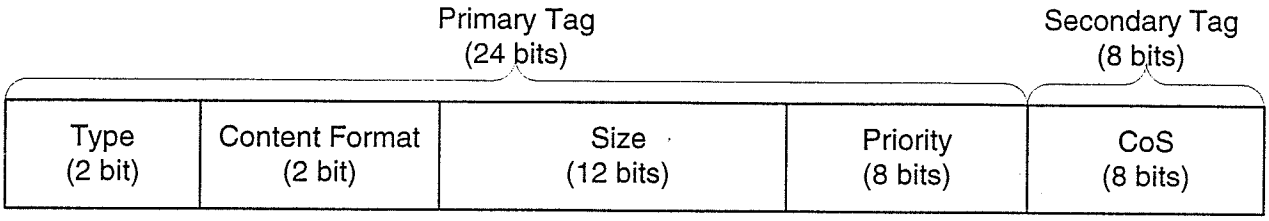

Figure 4.7: 32-bit content-derived tag format.

The 24-bit primary tag is the content-based name that is used to identify a content. The primary tag comprises of a 2-bit type field, 2-bit content format (CF) field, 12-bit size field and a 8-bit priority field.

The type field is used to distinguish between the different types of content that is being delivered. The following table shows the different values for the type field:

0 0 Streaming content

0 1 Non-streaming content

1 0 Temporary

1 1 Reserved

Streaming content identifies all audio/video data objects and non-streaming content refers to all text or application data and multimedia objects like images. Temporary field identifies that the tag assigned to the content is temporary. In case of a temporary tag, the CF, size and priority fields are replaced by the temporary tag. This means that the temporary tag can be of maximum 22 bits. The reserved field indicates future usage.

The 2-bit CF field will identify the end data type of the content. Table 4.3 gives a probable list of different end data types as defined by [FrB96]. The list shows an example of how end data types can be specified for different types of content. This list

can be altered, by adding or removing any end data types, by the discretion of the VCN administrator.

Table 4.3: A probable list for type and CF field combination.

Type Field	CF field	Document Format
0 0	0 0	Audio e.g., MPEG, Quicktime, Realplay, etc.
	0 1	Audio reserved
	1 0	Video e,g., MPEG, Quicktime, Realplay, etc.
	1 1	Video reserved
0 1	0 0	Image e.g., JPEG, BMP, TIFF, GIF, etc.
	1 1	Text e.g., plain, richtext, hypertext, directory, etc.
	0 1	Application data e.g., postscript, octet-stream, word, word perfect, etc.
	1 1	Multipart e.g., mixed, digest, form-data, encrypted, etc.
1 0	0 0	Streaming video
	0 1	Streaming audio
	1 1	Non-streaming video
	1 1	Non-streaming video
1 1	Reserved	Reserved

The size field represents the size of the CEC in the VCN. It is obtained by mapping the CECs onto divisions representing different sizes. Each division represents a range of document sizes in bytes and is identified by a 12-bit number i.e., we define 2^{12} or 4096 divisions and each division defines a range of bytes. The CECs are mapped to one of the divisions based on the CEC size which is the average size of all the documents in the CEC.

The priority field represents the popularity of a CEC in the VCN. The CEC popularity is given by the average popularity of all the documents in the CEC. Similar to the size field, the priority field is also represented by a set of divisions (2^8 or 64 divisions) with each division identifying a range of values.

The secondary tag consists of the 8-bit CoS (class of service) field. This will indicate the intended usage of the content or the service level agreement policies e.g., Diff-Serv. The intended usage is identified by the protocol number of the requesting application. A pre-defined list managed by the VCN administrator will provide a CoS field value for the different protocol numbers. The Diff-Serv class of services mentions the differentiated services that is to be provided to the client request. The CoS value identifying intended usage is always greater than the 6-bit codepoint value (codepoint identifies the standardized per hop behaviors in Diff-Serv).

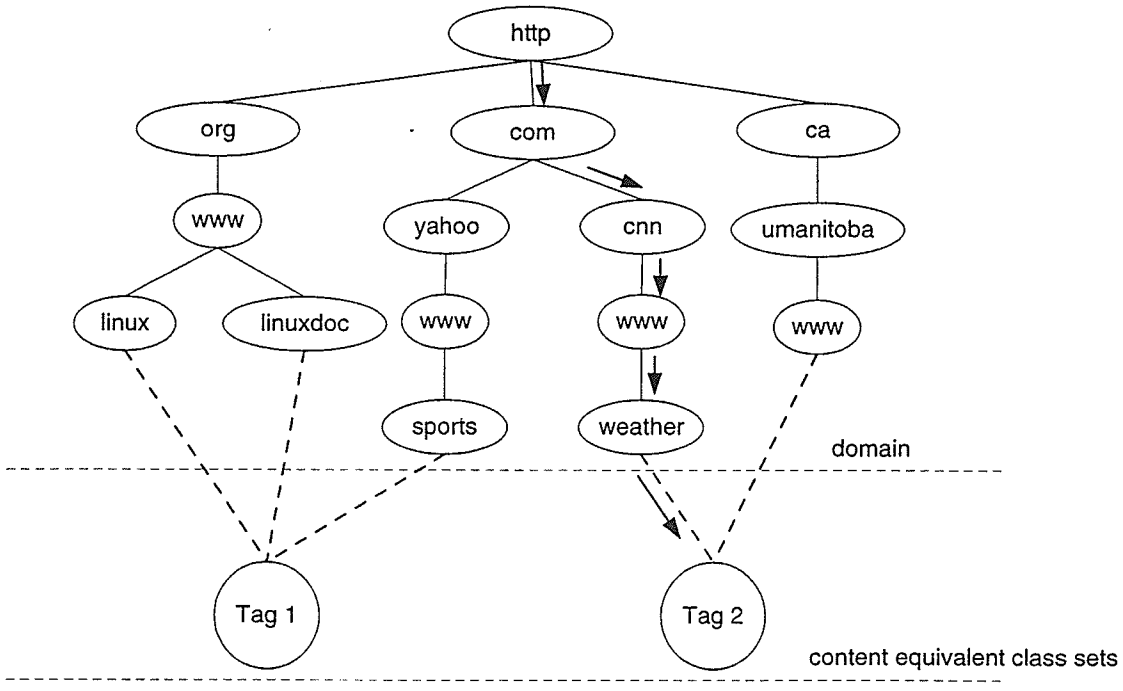

Figure 4.8: Namespace tree showing the content-to-tag bindings.

As mentioned earlier, all the CERs will create groups among themselves to hold the

entire namespace for all the documents hosted by the VCN. We represent the namespace in form of a tree. Figure 4.8 shows such a tree. The leaves of the tree hold the primary portions of the content-derived tags and the higher levels of the tree holds some part of the URLs to identify the documents. A client-side CER, upon receiving a request will parse the request to identify the document requested. It will then walk down the namespace tree to identify the tag assigned to the document. Once, the primary tag is identified the CER will generate the secondary tag for the request. Once the whole tag is generated the routing process is initiated.

4.5 PICS: Modes of Operation

This section examines the various operating modes for PICS. Before presenting the different modes, we examine the fundamental differences between the existing routing approaches and PICS approach.

A *content accessing scheme* should optimally implement the following two major functions to ensure efficient usage of the resources and to enhance the content delivery performance to the client.

Server selection: selects the site and the server within the site that serves the requested content.

Path selection: selects the path along which the selected server delivers the content to the clients.

Most of the existing content accessing approaches proceed in two phases: (a) resolution of a location-based name to obtain an IP address that specifies the destination host and (b) access the content from the destination host using the IP address. This process is illustrated in figure Figure 4.9(a). In the current Internet, the server selection

is exclusively handled by the name resolution phase. The path selection is performed in the second phase and is determined by the traditional Internet routing protocols such as *Open Shortest Path First* (OSPF) [Moy98] and *Routing Information Protocol* (RIP) [Mal94]. In case QoS constraints should be met by the selected path, we can use *Resource Reservation Setup Protocol* (RSVP) [Her00] or *Differentiated Services* (Diff-Serv) [BlB98].

Recently, several content-based name resolution schemes have been proposed. These schemes differ from traditional name resolvers such as *Domain Name Server* (DNS) [Moc87] in that they use a highly distributed "flat" resolver network. The request for resolution is routed through the flat resolver network using the content that is being requested. One of the advantages of this approach is that the name resolution may take into consideration fast varying parameters such as server load. The dissemination of such parameters may be localized to the nearby name resolvers.

Another alternative is to unify the server and path selection processes. The PICS scheme proposed here splits the name resolution into two phases. In the first phase, a content-based identifier (typically a location-based name but can be any other set of content attributes) is resolved to a fixed format tag. In the second phase, the tag is further resolved using a highly distributed "flat" routing network to reach the eventual server for the content. Depending on which mode of PICS operation is selected, the path for delivering the content back to the client is also selected by this process. This process is illustrated in Figure 4.9(b).

Below we describe two different modes of PICS operations. The first mode is referred to as the *Route Push Method* (RPM). In this method, the content-based routes are pushed from the server-side CERs, which are the eventual destinations of any request, towards the client-side CERs. The second mode is referred to as the *Route Push and Pull Method* (RPPM). In this method, the content-based routes are disseminated by a combination of

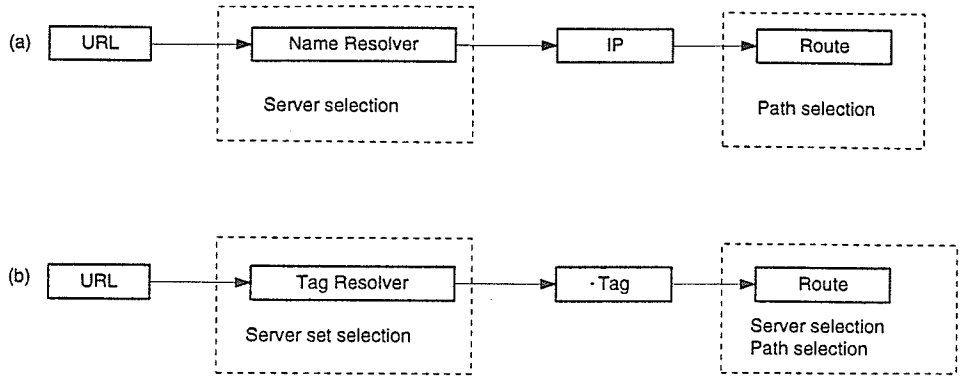

Figure 4.9: (a) IP resolution scheme (b) Tag resolution scheme.

push and pull-based schemes. In both the methods, we assume that the reverse flow that carries the data from the server to the client follows the same path that was traversed by the request.

4.5.1 Route Push Method

As mentioned previously, the VCN consists of CERs at its edge and CSRs at its core. The VCN can be considered as a highly distributed content router where the request for content enters the VCN via a client-side CER and is routed by the CSR network towards a server-side CER that can lead to the server with "best" performance. The server-side CER then delivers the request to the best server.

In this method, the server-side CERs receive advertisements from the hosting servers that are connected to them. The advertisements may contain some characterization of the content hosted at the servers (typically a content name) and utilization of the servers. The server-side CERs use a "tagging function" (content classifier) that takes the advertisements as arguments to derive content-based tags. The tagging function may be implemented at the server-side CER as a tree-walking process that uses a *namespace*

tree to bind a content-name to a tag. The binding process may use information from the content characterization schemes as explained in the next section. Once a server-side CER binds a content-name to a tag, two processes should take place: (a) dissemination of the tag and the corresponding content "serving" performance at the CER into the VCN, and (b) dissemination of the tag and content-name bindings to other CERs so that they could reuse the tags.

The tags formed by the server-side CERs are used to form a content-based route entry containing the (a) content tag and (b) effective server utilization index at the CSRs within the VCN. The tag dissemination is done in a "feed-forward" manner along a tree like structure with a server side CER initiating the distribution and acting as the root of the tree, the CSRs form the intermediate layers of the tree and the client side CERs form the leaves of the tree. Multiple *Tag Distribution Trees* (TDTs) may be formed in this way and may have common CSRs at different levels for the different trees. The advantage of such a scheme is that almost all the information available at the input layer, comprising of different server side CERs, will be available at the output layer, comprising client side CERs, and also provide multiple paths for each tag disseminated through the VCN. Routing table at each content router holds the content tags, the next hops for the tags and a route factor. Since there can be multiple paths for each tag, the route factor reflects the condition of each path.

Figure 4.10 shows two TDTs with CER-A as the root of TDT_1 and CER-B as the root of TDT_2. CSR-1 is at the first level with respect to TDT_1 and at the second level with respect to TDT_2. CER-1 and CER-2 form the leaf nodes of the two TDTs. Loop prevention algorithms are used to detect and prevent any loops among the content routers due to the tag-based routing. When a client request arrives at a client side CER (CER-A), a content tag that is derived from the content name is bound to the request. The CER-A examines its content-based routing table to obtain a possible next hop. A "route factor"

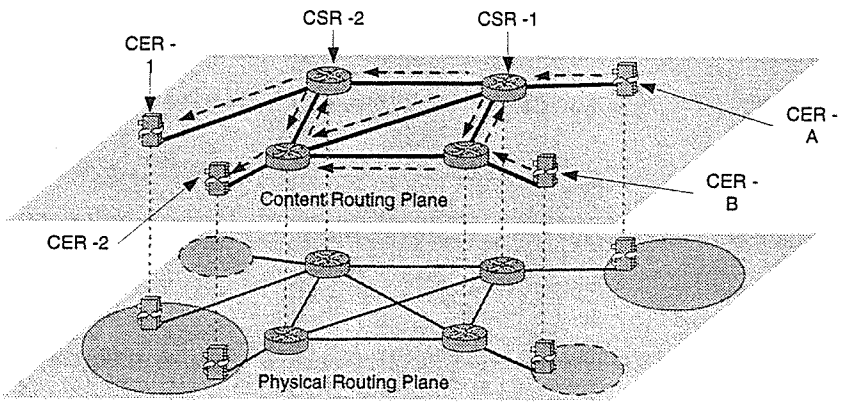

Figure 4.10: Tag Dissemination process for Route Push Method.

associated with the next hops is used to select the best performing next hop. Once a request is pushed into the VCN, it is routed towards the destination (server-side CER) by the CSRs using the content tag, i.e., the content name or any other information is not examined by the CSRs.

The server side CER forwards the request towards the best performing hosting server. Due to route propagation delays, the request routing will be based on progressively accurate information, i.e., as the request nears the destination it will be routed using more current information.

The data from the server is delivered to the client along the path that was traversed by the request. The route factors that were used to choose the next hops when the request was steered through the VCN are dependent on the network status and capacity as well. By computing the route factors appropriately, it may be possible to give probabilistic network QoS to the requests. Figure 4.11 illustrates the request steering and data forwarding in an example VCN. The client request initiates from `CER-1` and flows

through the VCN to the CER-A. The return path is the same as the forward path and it is used to deliver the content to the client.

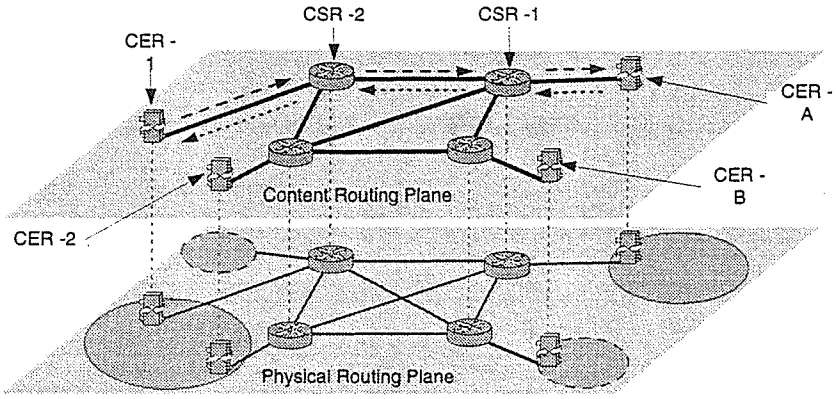

Figure 4.11: An example of request steering and content delivery for RPM.

The CERs may be grouped together such that a group maintains a single namespace tree. In each group, a CER holds a portion of the tree and maintains a pointer to the CER in the group that maintains the remaining portion of the tree. It should be noted that the group of CERs maintain a single namespace tree. Therefore, when a request reaches a CER it uses the appropriate CER within the group to resolve the content name to the tag and then injects the tagged request into the VCN.

4.5.2 Route Push and Pull Method

In the RPM, once a content-name to tag binding is created, it is pushed to all CER groups. Therefore, a group of CERs is supposed to know about a content-name to tag binding if the content with the given content-name is managed by the VCN. This approach may not be scalable unless the VCN is restricted to manage a reasonably small

number of different contents. In this case it may be best to choose the most popular set of content to be managed by the VCN.

The RPPM, on the other hand, provides a flexible scheme that enables the VCN to handle much larger numbers of content-name to tag bindings. In this method, the CER groups are organized into a virtual hierarchy as shown in Figure 4.12. The leaves of the hierarchy have physical CERs and the interior nodes of the hierarchy have *virtual CERs* (VCERs). A VCER may be implemented using one physical CER or a group of physical CERs.

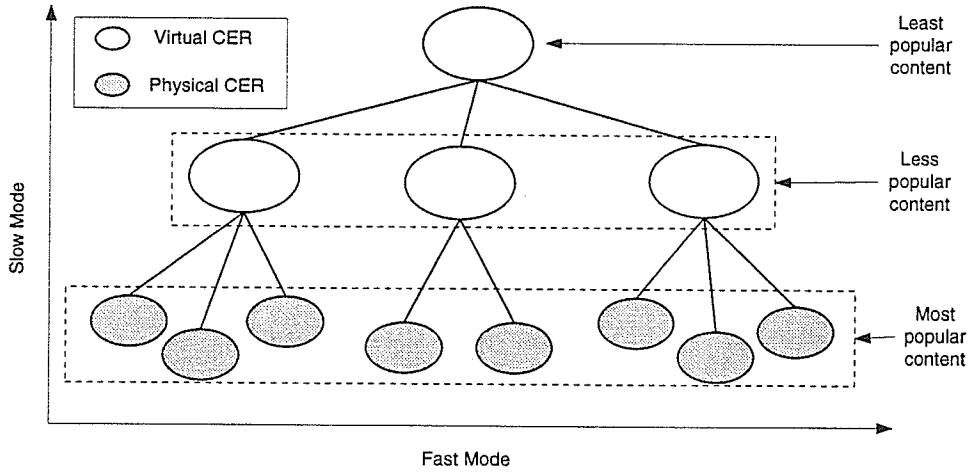

Figure 4.12: An example virtual CER tree.

The CERs examine the content requests that pass through them. With a virtual hierarchy such as the one presented above, it is possible to perform different types of traffic analysis. This analysis could be used to determine several traffic characteristics including (a) surges in demand, (b) relative demand of the different content, and (c) security violations and intrusions.

With the statistical information obtained through traffic analysis, we can determine the relative popularity of the various content handled by the VCN. This information is

used to map the content-name to tag binding information onto the virtual hierarchy. The idea is to map the content-name to tag bindings for the highly popular content onto the physical CERs. More precisely, each group of interoperating CERs should know about all the content in the "popular" set. Therefore, for a content in the popular set, there will be a maximum of one "miss" while resolving the content-name to a tag. The miss could happen at the ingress CER. In case of a miss at the ingress CER, a "hit" is certain, for popular content, at the next CER pointed by the namespace tree at the ingress CER. We refer to this timely processing of popular content as "fast mode" processing.

The less popular content are mapped onto the VCERs such that the least popular ones are mapped only at the root of the hierarchy. When a request for a less popular content (managed by a VCER) arrives at a CER, it will miss at the ingress CER and at the group level. The virtual hierarchy will be traversed to find the content-name to tag binding for this content. The resolution time increases as the content-name to tag binding information is held further up in the VCER tree. This is referred to as the "slow mode" processing. As the popularity of a content increases, the RPPM "pulls" the content-name to tag binding information down the hierarchy, thus decreasing the resolution time for subsequent accesses for the particular content.

The pulling of the content-name to tag binding information down the VCER tree creates only temporary copies of the binding information. When the surge in popularity subsides, the copies are deleted. Therefore, pulling the binding information down the hierarchy effectively moves the content processing from slow to fast mode. As the popularity for a document decreases, the content processing returns to the slow mode.

Initially, a content to temporary tag binding will be maintained at the lowest level of the VCER tree. Once the VCER tree starts receiving traffic analysis reports about the demand for the contents with temporary tags, the temporary tags will be pushed up or pulled down according to the demand across the different hierarchies on the VCER tree.

4.6 PICS Forwarding Algorithm

The forwarding algorithm used by the routing components of the VCN is based on the content-based tag and the routing fractions generated by the VCN. A routing fraction (RF) refers to the load status of a path between any two routing components (e.g., CSR and CER). A content hosting server site (S) will frequently update it's neighbor CER(s) with a RF that indicates the server's most current load status. The receiving CER will include the load status of the CSP, along which it received the status update, to the RF value received from the server site. The CER will then distribute the updated RF value to its neighbor CSR(s) and CER(s). This way, the RF value received at a client-side CER will indicate the condition of an entire path (i.e., sequence of CSPs) from the server S to itself. Each of the content hosting sites in the VCN will generate such a status update packet. Each of the CSR/CER(s) will maintain an RF table to store the RF values received from all its neighbors. The RF table will be updated very frequently. Figure 4.13 illustrates the PICS forwarding process.

Each CSR and CER (specifically CER groups) will also maintain a routing table, where each entry contains a tag component and a probable next hop for the tag. The next hop will be a neighbor CER or a CSR, or a content hosting server. The forwarding algorithm which uses the RF values for the neighbors to make routing decision works as follows. When an ingress CER receives a request packet, it will encapsulate the request packet with an appropriate tag obtained from a VCER tree. The ingress CER will use its routing table to find the possible next hops for the tag used to encapsulate the request packet. It will choose the next hop with the best RF value and forward the encapsulated packet to that next hop. A CSR on receiving an encapsulated packet will extract the tag from the encapsulated packet and use it as an index to select the possible next hops from the routing table. Again, next hop with the best RF value is chosen and the CSR

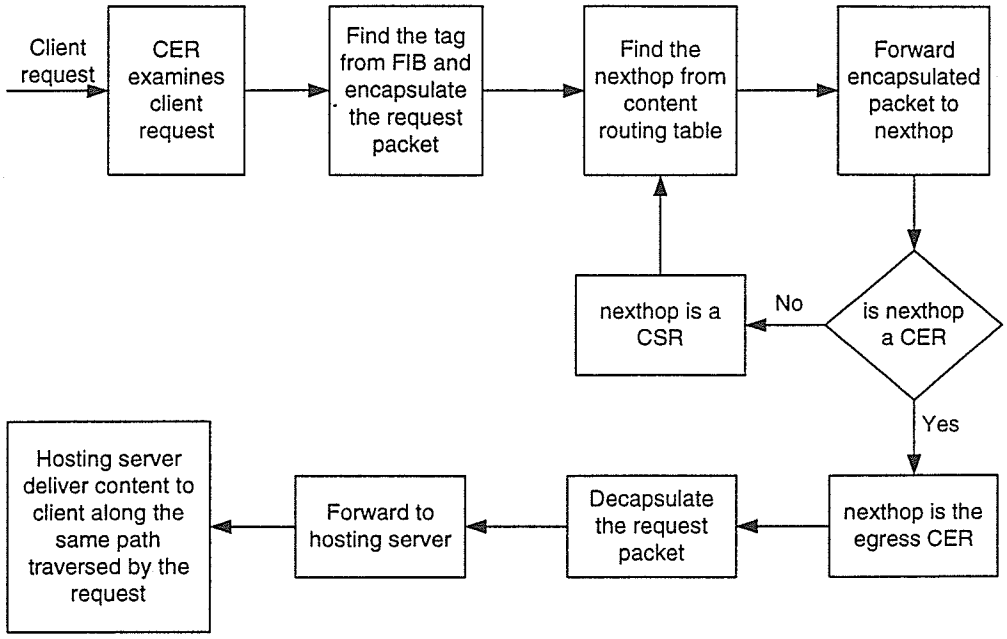

Figure 4.13: Content-based forwarding process.

will forward the encapsulated packet to the next hop. An egress CER will receive an encapsulated packet, extract the tag and find the best next hop from the routing tables. If the next hop selected is a content hosting site, the encapsulation (i.e., the tag) is stripped off and the original packet is forwarded to the server. If the next hop is a CER or CSR, the encapsulated packet is forwarded to the next hop. A combination of the primary tag and secondary tag will identify the resource requirements of the requested content. The RF value from the RF table at a router will identify whether enough resources are available for delivery of the content along a CSP.

Chapter 5

Modeling for Performance Analysis

5.1 Overview

In this section, we develop a linear model to compare the performance of our VCN-based routing scheme with the existing DNS-based routing scheme. First, we develop mathematical equations to express the total time required for request routing and content delivery phases for both routing schemes. Next, we compute the costs incurred due to the overhead caused by server status updates and tag updates for the two routing schemes. In Chapter 6, we corroborate the model we developed here via simulation studies.

As explained earlier, any request routing scheme is composed of two functions: discovery and access of content. The goal of this analysis is to observe the events that occur during the discovery and access and develop linear equations to express the timing relationships among the events. The model considers the major entities that lie on the path traversed by the requests for content and the replies with content. These entities differ based on the the scheme being modeled. For the VCN-based routing scheme, the major entities are: (i) client (ii) VCN comprising the ingress CER, CSR(s), and the egress CER

and (iii) content hosting servers. For DNS-based routing scheme, major entities are: (i) client (ii) WAN comprising of the name resolvers and routers (mainly IP routers) and (iii) content hosting servers. We measure the time required for each event to complete and use these measurements to compute following parameters:

Access latency: refers to the total time required by an entity to send a request to its next hop and receive a response from it.

VCN/WAN delay: is the total delay incurred by the VCN and the WAN to route a request to an appropriate content hosting site and deliver the contents back to the client.

Request bandwidth: is the bandwidth required by the VCN/WAN to deliver a content to the client.

Link Utilization per request: computes the utilization of the links in the network due to the traffic generated by the requests. This measurement is the largest recurring cost factor in any network.

Model Setup

To measure the above mentioned parameters, we consider OBJ to be the set of K objects (i.e., documents) available on the Internet. For the PICS routing scheme, we assume that the VCN is aware of the locations of all the K objects and their replicas. For the DNS-based routing scheme, we assume that all the domains hosting the K objects are known to the name resolvers. For each document k in OBJ we define an access sequence $\omega(k, n)$, where n counts the number of events that occur while accessing object $OBJ(k)$ from its storage (the storage may be an origin or a replica server). The set of all access sequences is given by $\phi = \{\ \omega_i(k, n) \mid i, n \in \mathbb{N} \text{ and } 0 < k \leq K\ \}$. Next, for each of the events occurring along the path of the access sequence ω_i, we determine the time values for the VCN-based routing scheme and the DNS-based routing scheme. The

following terminology is used to describe the events occurring for the sequence $\omega_i \in \omega$:
$T^{Rq}_{S,i}(x)$: time at which an entity x sends the first byte of the request i to the next hop
$T^{Rq}_{E,i}(x)$: time at which an entity x sends the last byte of the request i to the next hop
$T^{Rp}_{S,i}(x)$: time at which an entity x sends the first byte of the reply for request i
$T^{Rp}_{E,i}(x)$: time at which an entity x sends the last byte of the reply for request i
$T_{S,i}(x)$: time to receive the first byte of response for request i by an entity x from the next hop
$T_{E,i}(x)$: time to receive the last byte of response for request i by an entity x from the next hop
where, entity x $(0 < x \leq n)$ represents the entities involved in the routing schemes.
We also define the variables:

$d_{x,x+1}$: latency of the link between the entities x and $x+1$
$b_{x,x+1}$: bandwidth of the link between the entities x and $x+1$
Δ_x : latency at entity x
s^k: size of packet of type k

5.2 Assumptions

The performance of a content-aware routing scheme depends on many factors including the average round trip time (RTT) between the nodes, load status of content hosting servers and the architecture of the routing scheme. We are, however, interested to find the performance of a routing scheme in terms of the overall time required to deliver a content across a network. For this purpose, we make some assumptions that relaxes some conditions in our linear model. The assumptions made are as follows:

Full-site content delivery A single client request for a document often generates multiple requests for embedded objects in the document which may reside on different hosting sites. In our model, we assume a full-site content delivery, where each request is served from a single server. We assume, that all content needed by a requested document reside on the same content hosting site. Moreover, each content hosting site responds correctly to all the requests it receives from clients (i.e., failure to respond to a request is considered negligible)

Control packet loss In a congested network it is difficult to note the loss of the control packets (e.g., TCP connection setup packets or control messages used by the routing protocols). Such control packet loss impacts the total response time between the nodes. We assume that the loss of control packets is small compared to the total number of packets in the network and can be considered negligible.

Routers Routers are store-and-forward and use FIFO queueing. This is done to ensure that request and document delivery orderings are preserved. Moreover, we assume that the packet processing time at a router is less significant than the packet delivery time.

Average RTT between the nodes: Request routing and content delivery in any routing scheme depends on the average response times among the nodes. The response time between any two nodes can be measured by the following equation and is illustrated by the Figure 5.1.

$$t_1 = t_0 + (\frac{s}{b} + d)$$

where, s = size of the document being transported across a link, b = the bandwidth of the link between the two nodes and d = latency of the link between the two nodes.

Each link contributes some transmission delay $\frac{s}{b}$ and latency d. The transmission delay is due to the time a node needs to copy and serialize all the bytes of a packet onto a link. The link latency is due to the time required for a signal to travel across a link and other fixed per-packet delays that a node incurs before forwarding a packet.

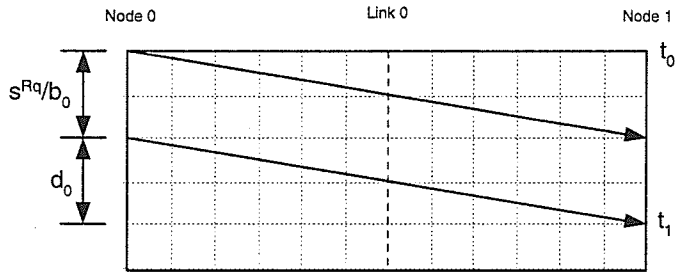

Figure 5.1: The figure shows the total amount of time required to deliver a document from one node to another. In this example, s is the size of a content, b_0 is the bandwidth of the link between the two nodes and d_0 is the latency of the link.

Congestion Each link delivers more than one packet at any given time. When the total amount of packets on a link becomes very large it results in congestion. This, in turn, increases the transmission delay.

5.3 Model for PICS Routing Scheme

Here, we present a pipeline model for the access sequence $\omega_i(k, n = 5)$ and assume that the packet processing time at the content routers is almost negligible. Figure 5.2

illustrates the timing relationships between the entities client (C), ingress CER (I), CSR (R), egress CER (E) and the server (S) where the entities are numbered in sequence starting from the client. The figure shows an access sequence with a single CSR along the routing path. However, the number of CSRs may vary depending up the path chosen by the routing protocol. We first determine the equations to compute the access latency and bandwidth for the given situation and then we present a more generalized expression for the equations such that they represent a practical situation with more number of entities along a routing path.

(a) client/VCN access latency

$$\begin{aligned} L^C &= T_{E,1}(1) - T^{Rq}_{S,1}(1) \\ &= \sum_{x=1}^{4}(T^{Rq}_{S,1}(x+1) - T^{Rq}_{S,1}(x)) + \Delta_S + \sum_{x=1}^{4}(T^{Rp}_{S,1}(x) - T^{Rp}_{S,1}(x+1)) \\ &= \sum_{x=1}^{4}(\frac{s^{Rq}}{b_{x,x+1}} + d_{x,x+1}) + \Delta_S + \sum_{x=1}^{4}(\frac{s^{Rp}}{b_{x,x+1}} + d_{x,x+1}) \\ &= \sum_{x=1}^{4}(\frac{(s^{Rq} + s^{Rp})}{b_{x,x+1}} + 2d_{x,x+1}) + \Delta_S \end{aligned}$$

(b) delay incurred in the VCN to route request

$$\begin{aligned} L^V &= T^{Rp}_{E,1}(2) - T^{Rq}_{S,1}(2) \\ &= \sum_{x=2}^{3}(\frac{(s^{Rq} + s^{Rp})}{b_{x,x+1}} + 2d_{x,x+1}) + \Delta_S \end{aligned}$$

(c) total bandwidth required by the VCN to deliver a content

$$B^V = max(s^{Rq}\sum_{x=2}^{3}\frac{1}{T^{Rq}_{E,1}(x) - T^{Rq}_{S,1}(x)}, s^{Rp}\sum_{x=2}^{3}\frac{1}{T^{Rp}_{E,1}(x) - T^{Rp}_{S,1}(x)})$$

The variables L^C, L^V and B^V are random in nature as the number of the CSRs may vary depending on the access sequence. For an access sequence $\omega(k, n = N)$ the access

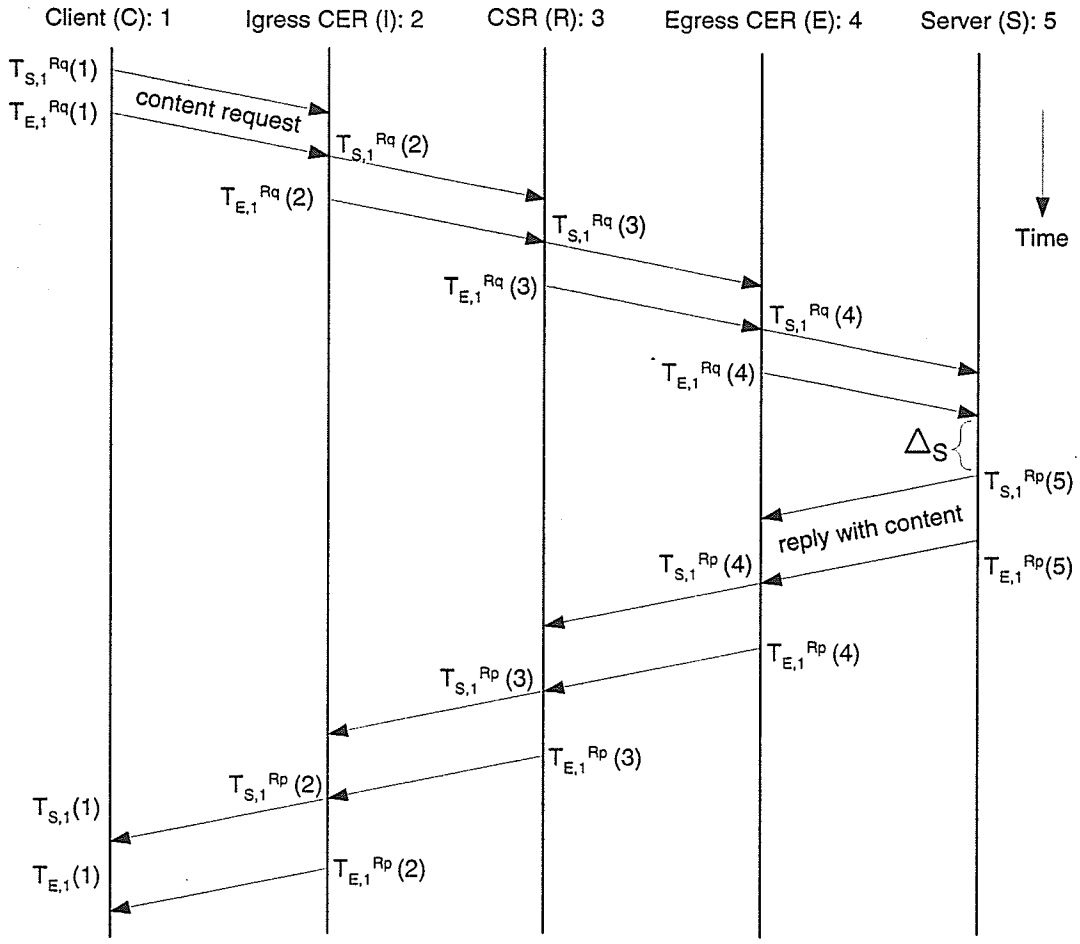

Figure 5.2: The time values for events for discovering and accessing content in PICS routing scheme, this is a pipelined model where we assume that the packet processing time for a content router is negligible.

latencies and bandwidth can be generalized as:

$$L^C = \sum_{x=1}^{N-1}\left(\frac{(s^{Rq}+s^{Rp})}{b_{x,x+1}} + 2d_{x,x+1}\right) + \Delta \tag{5.1}$$

$$L^V = \sum_{x=2}^{N-2}\left(\frac{(s^{Rq}+s^{Rp})}{b_{x,x+1}} + 2d_{x,x+1}\right) + \Delta \tag{5.2}$$

$$B^V = max\left(s^{Rq}\sum_{x=2}^{N-2}\frac{1}{T_{E,1}^{Rq}(x) - T_{S,1}^{Rq}(x)}, s^{Rp}\sum_{x=2}^{N-2}\frac{1}{T_{E,1}^{Rp}(x) - T_{S,1}^{Rp}(x)}\right) \tag{5.3}$$

where Δ represents the latency at the server entity.

Again, when no CSRs (entity 3) and egress CERs (entity 4) are not required for routing, the minimum VCN delay can be measured as: $L^V_{min} = (s^{Rq}+s^{Rp})(\frac{1}{b_{2,3}}+d_{2,3})+\Delta_S$, where server replaces the CER as the entity 3.

5.4 Model for DNS-based Routing Scheme

We now present a pipeline model for the DNS based routing scheme for the access sequence $\omega_i(k, n = 4) \in \omega$. Similar to the PICS scheme, the packet processing time of the WAN entities are negligible. Figure 5.3 shows the timing relationship between the entities client (C), name resolver (R), IP routers (R) and the server (S). Similar to the PICS scheme, the entities are numbered in sequence starting from the client. The figure shows an access sequence with a single IP router. Similar to the previous section, we first compute the equations for the given access sequence and then we compute a more generalized formula for a practical situation involving multiple IP routers.

(a) the client/WAN latency is measured by

$$\begin{aligned} L^C &= T_{E,2}(1) - T^{Rq}_{S,1}(1) \\ &= (\frac{s^{DNS_{Rq}} + s^{DNS_{Rp}} b_{1,2}}{b_{1,2}} + 2d_{1,2}) + \Delta_D + \Delta_C \\ &\quad +(\frac{s^{Rq} + s^{Rp}}{b_{1,3}} + 2d_{1,3}) + \Delta_S + (\frac{s^{Rq} + s^{Rp}}{b_{3,4}} + 2d_{3,4}) \end{aligned}$$

(b) delay incurred in the WAN

$$\begin{aligned} L^W &= T^{Rp}_{E,2}(3) - T^{Rq}_{s,2}(3) \\ &= \Delta_S + (\frac{s^{Rq} + s^{Rp} b_{3,4}}{b_{3,4}} + 2d_{3,4}) \end{aligned}$$

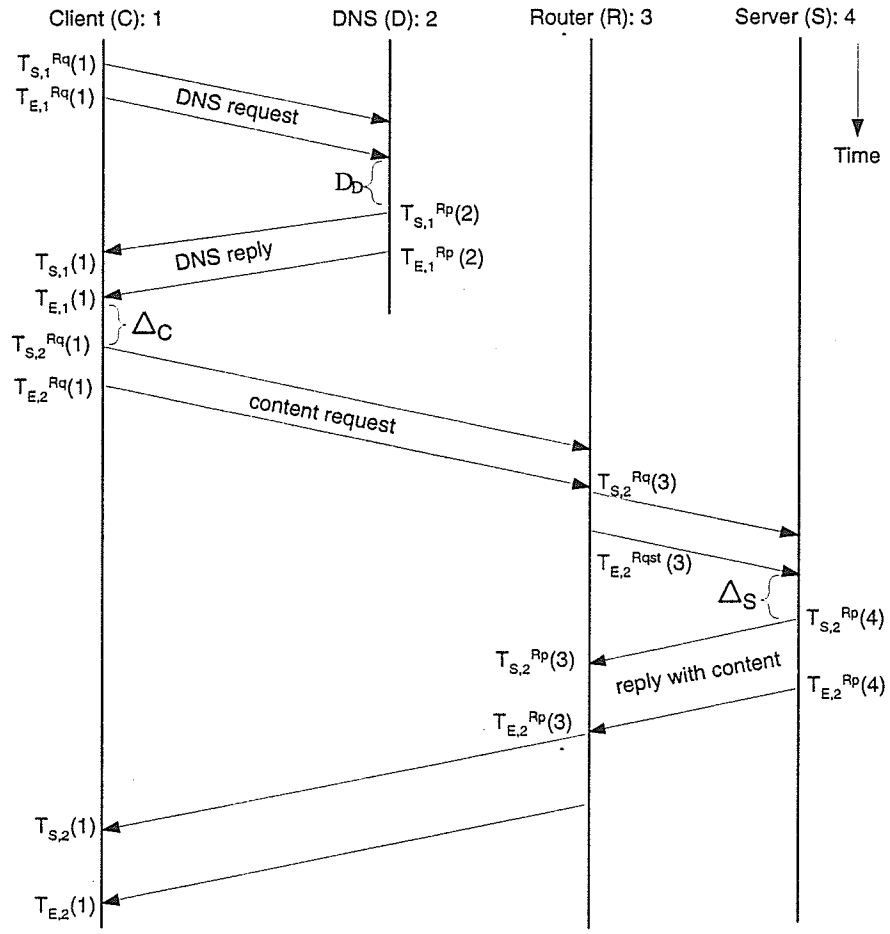

Figure 5.3: The time values for events to discover and access a content in DNS based routing scheme, this is a non-pipelined model where the packet processing time for each entity is negligible.

(c) total bandwidth required by the WAN to deliver the content

$$B^W = max(\frac{s_{Rq}}{T^{Rq}_{E,2}(3) - T^{Rq}_{S,2}(3)}, \frac{s_{Rp}}{T^{Rp}_{E,2}(3) - T^{Rp}_{S,2}(3)})$$

As mentioned earlier, in practice the number of nameservers used for name resolution and the number of IP routers along request and reply path are variable. For an access sequence $\omega(k, n = N)$, where the number of DNS servers be m and the number of IP routers be r such that $N = m + r$, the access latencies and bandwidth can be measured

by:

$$L^C = \sum_{x=1}^{m-1}\left(\frac{s^{DNS_{Rq}} + s^{DNS_{Rp}}}{b_{x,x+1}} + 2d_{x,x+1}\right) + \sum_{x=1}^{r-1}\left(\frac{s^{Rq} + s^{Rp}}{b_{x,x+1}} + 2d_{x,x+1}\right) + \Delta \quad (5.4)$$

$$L^W = \sum_{x=1}^{r-1}\left(\frac{s^{Rq} + s^{Rp}}{b_{x,x+1}} + 2d_{x,x+1}\right) + \Delta \quad (5.5)$$

$$B^W = max\left(\sum_{x=1}^{m-1}\left(\frac{s_{Rq}}{T^{Rq}_{E,2}(x) - T^{Rq}_{S,2}(x)}\right), \sum_{x=1}^{m-1}\left(\frac{s_{Rp}}{T^{Rp}_{E,2}(x) - T^{Rp}_{S,2}(x)}\right)\right) \quad (5.6)$$

5.5 Link Utilization

Let us consider a network comprising N routers connected by J links. The links have different bandwidth values. In case of PICS, we assume that the routers are content aware and the links as tunnels or fast logical paths that connect the CERs and CSRs. In case of DNS-based routing scheme, the routers are assumed to be IP routers and the links as the physical links. The edge routers in the network are connected to client and server sites and the network is used for transferring content from the server to the client sites. We assume that there is a link j ($j \leq J$) in the network which has the ability to deliver a specified fraction of requests from the server to the client. We also assume that the arrival rate of requests for I documents follow a Poisson process.

Let:

T: the time interval for which we measure the utilization of a link j

T_s/T_e: the starting/ending time for the interval T (i.e., $T = T_e - T_s, T_e > T_s$)

B_j: link speed (bytes/sec) of the link j

$b_j(t)$: amount of bytes delivered along a link j as a function of time

S_i: size of a document i delivered over the link j

The total amount of bytes that the link j can deliver over a time interval T is given by:

$$b_j^T = B_j \times T$$

The actual amount of bytes being delivered across the link j over time T is measured by:

$$b_j^{'} = \int_{T_s}^{T_e} b_j(t)dt$$

Then, the utilization of link j is measured by:

$$\rho_j = \frac{b_j^{'}}{b_j^T} == \frac{1}{B_j \times T} \int_{T_s}^{T_e} b_j(t)dt$$

The mean time taken to deliver a document i $(0 \leq i < I)$ across the link j is given by

$$\delta_{ij} = \frac{S_i}{B_j(1-\rho_j)}$$

5.6 Overhead Due to Tag Dissemination and Status Update

In both PICS and DNS based routing schemes, we need to update the status of the network and the servers at regular intervals. In the above subsections, we have already measured the total time and the amount of resources (in terms of bandwidth) required by both the routing schemes for delivering the content. However, a substantial amount of time and resources are required for updating the content-based routers and nameservers with the current conditions of the network and the hosting servers. In this subsection. we

measure this overhead in terms of bandwidth required for tag and status updates. The analytical model developed here is used to calculate the overhead incurred using different conditions in the simulator and we deduct the overhead calculated from the resources available in the simulator before the content routing process starts. This way, we can incorporate network overhead but keep the simulator design simple.

5.6.1 PICS Routing Scheme

In the VCN-based routing scheme, the status updates are disseminated to all the CERs and CSRs in the VCN. As an initial case, the status packets are flooded along the CSPs (as implemented in our simulator). However, in practice, status and tag updates should follow some optimized scheme for dissemination. When a status packet originates at a server-side CER, its content reflects the load status of the server to which this CER is connected. As the status packet moves across the CSPs, its content is updated at the CSRs connected across the CSPs to include the status of the CSP along which the updating CSR forwards the packet. This way, when a status packet reaches a client-side CER the status packet reflects the state of an entire path (i.e., a set of CSPs) between the client and server side CERs.

A CSP (CSP_j) always connects to two CSRs (i.e., one at each end). Then at any given time, the maximum number of status packets that can be carried by CSP_j is two i.e., one packet pushed on CSP_j by each of the CSRs on either ends of CSP_j. If the size of a status packet is fixed to s_s (bytes) and the status update interval be T_s seconds,

then maximum bandwidth required by CSP_j to carry the status packets is given by:

$$b_j^s = \frac{2s_s}{T_s}$$

The tag update packets (i.e., the content-to-tag bindings) which also originate at the CERs are piggybacked on to the status packets for simplicity. Normally, tag updates occur after much longer intervals than the status updates (i.e., $T_s \ll T_t$). By piggybacking, we can reduce the dissemination overhead. This technique reduces the total amount of bandwidth used by the status and tag update messages.

5.6.2 DNS-based Routing Scheme

In DNS-based routing, a hierarchical system of nameservers are used to translate between IP addresses and domain names. Before the actual name resolution begins the name-servers have to be updated with all the domain names that are present in the Internet. In our simulations, we consider a 3-tier Internet topology where the nameservers are placed hierarchically across the three tiers and are used to translate the domain names to IP addresses. Our simulation setup is described in detail in Section 6.1. In this section, we calculate the bandwidth overhead caused by updating the name-servers with all available domain names and server load status information. We consider the following variables to compute the overhead:

s_d: size (bytes) of the DNS update packets

s_s: size (bytes) of the status update packets

T_d: DNS update interval (seconds)

T_s: status update interval (seconds)

N_A: number of AS(s) present in the network

N_M: number of MAN(s) per AS

N_L: number of LAN(s) per MAN i

H_j^{os}: number of origin servers per LAN j

H_j^{ss}: number of surrogate servers per LAN j

In our model of DNS-based routing, the clients and the content hosting servers are located at the LAN level. The lowest level of nameservers are placed in the MAN level and these act as the local nameservers for the clients and the content hosting servers. We assume that each client and content hosting server connects to a single nameserver but a MAN level nameserver may be shared by multiple clients and hosting servers. Each origin server update their local nameserver about all the server names they host. The surrogate servers update the local nameservers of the origin servers about the replicas they host. Usually, the status updates are much more frequent than the DNS updates (i.e, $T_s \ll T_d$). Thus, similar to the PICS scheme, the DNS update packets are piggybacked on to the status update packets for better efficiency.

The LAN level hosts are mostly connected by a single link to the LAN level IP routers. The LAN level IP routers connect a LAN to the MAN level hosts. The maximum overhead of the links in LAN_i that connect the origin and surrogate servers to the LAN level IP routers is caused by the status updates from each server host. This overhead is given by:

$$b_{L_i} = \frac{s_s}{T_s}$$

and the maximum overhead of the links (or the paths) that connect the IP routers in LAN_i to the nodes in a MAN_j is caused by all the status update packets originating at the LAN_j. This overhead is measured by:

$$b_{L_i}^{M_j} = \frac{s_s(H_i^{os} + H_i^{ss})}{T_s}$$

The MAN level nameservers also receive updates from surrogates which host copies of the origin servers but are not placed in the same LAN as the origin servers. If we assume that the rate of replicating the origin servers is α and the surrogates for origin servers in LAN_i are located in a LAN which connected to a different MAN (MAN_k), then the links that connect the MAN_j to MAN_k is given by:

$$b_{M_j}^{M_k} = \frac{s_s \alpha H_i^{os}}{T_s}$$

Then, the maximum overhead that a link in MAN_j can experience is measured by:

$$b_{M_j} = \frac{s_s}{T_s} \sum_{i=1}^{N_L} [(1+\alpha)H_i^{os} + H_i^{ss}]$$

The nameservers in the MAN level update their zonal nameservers which are located in the WAN level about all the second level domain names. The zonal nameservers know about all the domains (e.g., umanitoba.ca) that are hosted in their zone. In our simulations, we consider there is one zonal nameserver per AS. Therefore, the overhead incurred due to DNS updates along the links connecting a name server in MAN_j to its zonal nameserver in WAN_p is given by $\frac{s_d}{T_d}$ *bytes/second*. The maximum overhead that a link leading to a zonal nameserver in WAN_p from the nameservers in it's zone (i.e., in

MAN level) is given by:

$$b_{M_j}^{W_p} = \frac{s_d.N}{T_d}$$

where N = number of nameservers in one zone and $N = N_M$ for our simulations.

The maximum overhead that a link in WAN_p can experience is computed by:

$$b_{W_p} = \frac{s_d.N.N'}{T_d}$$

where N' = number of zonal nameservers in WAN_p and $N' = N_A$ for our simulations.

Chapter 6

Simulations

6.1 Simulation Setup

Our simulator uses a hierarchical three level network topology that is divided into multiple autonomous systems (AS) and closely resembles the traditional Internet infrastructure comprising the WAN, MAN and LAN levels. The network topology is generated by first creating a flat AS network at the WAN level, using BRITE [MeL01], and then expand each AS into MAN and LAN levels, using Tiers [Doa96] topology generator. The network entities used by the routing schemes being modeled are mapped on the three level network using heuristics. A C-based discrete-event simulation language [BaM98] is used to implement the network entities.

Even though the simulator deploys a network model that is similar to the traditional infrastructure of the backbone networks, for simplicity, we limit the placement of the

content hosting servers (i.e., origin and surrogate servers) at the lowest level (i.e., LAN level) of the 3-tier network. The IP routers are placed at the edge of WAN, MAN and LAN levels connecting a WAN with a MAN level and a LAN with a MAN level. IP routers are also placed inside each WAN, MAN, and LAN to provide connectivity among the nodes on each plane. Bandwidths are assigned to the network links by the topology generators, BRITE and Tiers. Keeping this setup, the remaining nodes are placed as follows.

DNS-based Routing Model Setup

For the DNS-based routing scheme, the nameservers are placed hierarchically across the MANs and WANs and resembles the hierarchical structure of DNS servers and their databases on the Internet [LiA01]. Each client and server in the LAN level connects to a nameserver at the MAN level. These nameservers act as local name servers for the hosts in the LAN level. The next higher level name server are the zonal nameservers which are located at the WAN level. For simplicity, we maintain one zonal nameserver per AS. In case of a DNS *miss* the zonal nameservers contact a root name server which is also placed in the WAN level. The root nameserver knows about the authoritative nameservers which are also placed in the MAN level. The authoritative nameservers hold the top level domain (TLD) names. The authoritative nameserver knows about all the zonal servers and the second level domain names known to the zonal nameservers. The clients contact their local nameserver at the MAN level. In case of a DNS *miss*, the MAN level nameserver refers the request to its zonal nameserver. If the zonal nameserver cannot

resolve the request, it refers the request to the root nameserver which directs the request to an authoritative nameserver. The authoritative nameserver informs the requesting client's local name server at the MAN level about the zonal nameserver which hosts the requested name. The client's local nameserver contacts the zonal name server which directs the request to the local nameserver of the requested site. The local nameserver of the requested site returns an IP address for the requested site to the client's local name server. This IP address may be an origin site or any surrogate hosting the copy of the origin site. The client's local nameserver returns the IP address to the requesting client. The client on receiving an IP address contacts the address. Routing of the packets between the nodes are done using the OSPF routing protocol.

VCN-based Routing Model Setup

In the PICS routing schemes, the content routers are placed at the edge of the WAN, MAN, and the LAN. The CERs are located at the MAN level and acts as the gateway to the LANs. A CER may be connected to one or more LANs. The CSRs are placed at WAN level and are connected to one or more MANs. The CERs and CSRs together form the VCN and the content-based routers are connected by logical fast paths. Each origin and surrogate server update their local CER (i.e., the CER to which the LAN comprising the server is connected) with all the contents that it hosts. The CERs generate the tags and update all other CERs with content-to-tag bindings and the CSRs with tags. At this stage, the tag is disseminated within the VCN by flooding. The overhead for the flooding method is computed in the previous chapter. The routing between the content-

based routers is handled by the PICS forwarding algorithm while the routing between the physical nodes is done by the OSPF protocol.

Workload

The simulator is driven by traces generated by the ProwGen [BuW01], a synthetic trace generator. Content characteristics like the file size and popularity of the documents, which are generated by the synthetic trace generator, are used for content replication within the simulated network. The content placement is random but the number of copies of the documents is proportional to the demand (i.e., popularity) of the requests. The replication of the contents are handled in two ways: (a) *site-based*: set of all documents hosted by an origin server site is copied on to the surrogate servers (b) *set-based*: a subset of documents from each origin server are replicated. The subsets are chosen depending on the popularity of the documents. The network overheads for both routing protocols are calculated as per the equations given in Section 5.6. However, in this version of the simulator, the server bandwidth is considered to be infinite. The storage capacity of surrogate servers are also assumed to be unlimited. The requests for the simulator are generated using a Poisson distribution over a period of one hour. We perform test runs with the simulator for different sizes of the network (i.e., number of nodes in the network) and for different trace files. We run the tests for 50, 150, 250, and 500 nodes. The total number of distinct documents that are handled by both the routing schemes is 3000 for all the runs. Each test run use trace files for 10,000 and 100,000 requests. Tests are run using both set-based and site-based replication.

6.2 Results and Discussion

We use the following metrics to measure the performance of the VCN-based and DNS-based routing scheme: (a) resolution time: the time required to resolve a client request on to a content hosting site, (b) delivery time: the time required to deliver the requested content from the content hosting site to the requesting client, (c) throughput: the amount of bytes delivered by the routing network per unit time, (d) link utilization: the amount bytes transfered through each link with respect to the total capacity of the links, and (d) the impact of delivering streaming and non-streaming content using the VCN-based routing scheme. Each graph used to present the simulation result is titled by the rate of replication (i.e., number of copies for a content expressed in percentage) and the number of requests used in the experiments. Each graph presents the results of three different setups (i) DNS site-based: DNS-based routing scheme where the popular web sites are replicated on surrogates as a whole (ii) PICS site-based: VCN-based routing scheme where the popular web sites are replicated on surrogates as a whole and (iii) PICS set-based: VCN-based routing scheme where the sets of documents are replicated depending on their popularity (i.e., more popular document sets have more copies in the network than the less popular documents).

6.2.1 Resolution Time

In this section section, we measure and compare the resolution time for both the routing schemes. Figures 6.2, 6.3 and 6.4 shows the resolution time with 10,000 requests generated

per LAN. As the number of content replications increase we see that resolution time decrease. Figure 6.2 shows an approximate reduction of 17% in resolution time for PICS, Figure 6.3 shows an approximate reduction of 23% in resolution time for PICS and Figure 6.4 shows an approximate reduction of 27% in resolution time for PICS. Figures 6.5, 6.6 and 6.7 shows the resolution time with 100,000 requests generated per LAN. Figure 6.5 shows an approximate reduction of 30% in resolution time for PICS, Figure 6.6 shows an approximate reduction of 36% in resolution time for PICS and Figure 6.7 shows an approximate reduction of 38% in resolution time for PICS. The DNS-based routing, however, was executed without any caching of the DNS decisions. In our simulations, in absence of a DNS caching scheme approximately 95% of the requests traverse the nameserver hierarchy for request resolution and approximately 38% of the requests contact the root nameserver. By caching DNS decisions the name resolution latency can be cut down to an considerable extent. However, in order to accurately respond to the fast varying server conditions, the client-side nameservers must avoid caching of the DNS decisions [ShT01].

We attribute the improvement in name resolution performance of the VCN-based routing as compared to the DNS-based routing to the fact that in the latter case most of the requests traverse the nameserver hierarchy placed across the 3-tiers network topology for mapping the request on to an IP and then routing the request to this IP. This results in considerable delay due multiple queries, replies and referrals along the nameserver hierarchy. Figure 6.1(a) shows a worst case for name resolution by traversing the entire

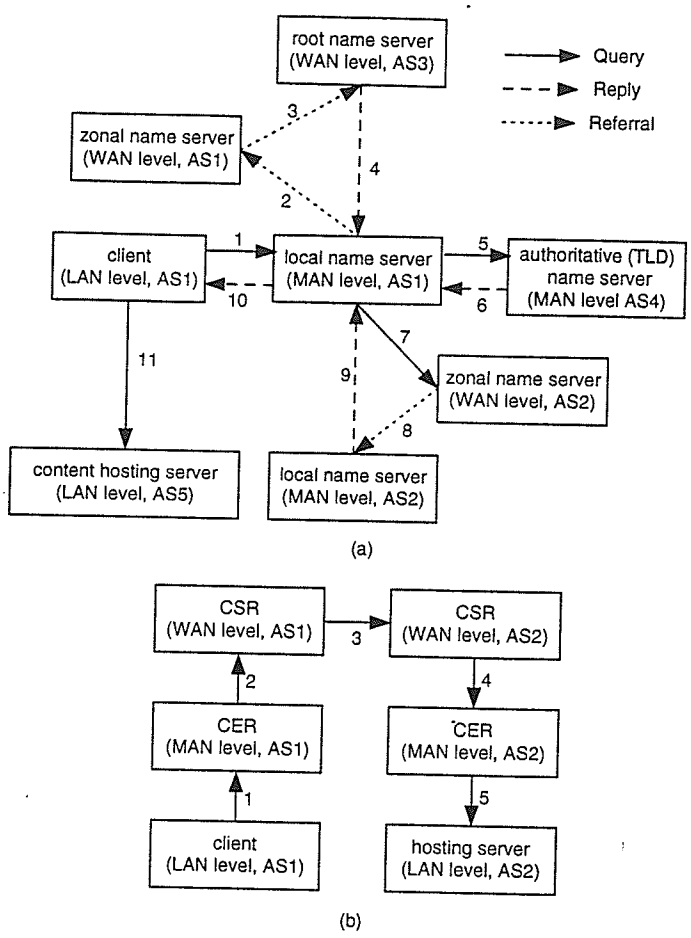

Figure 6.1: (a) The DNS name resolution process (b) Name resolution process in the VCN.

DNS nameserver hierarchy. A DNS request has to move up the entire DNS hierarchy from the LAN to the WAN to query the root nameserver located in the WAN level and return a reply to the client's local nameserver in the MAN level. The client's local name server then queries the authoritative (TLD) nameserver, referred by the root nameserver and located in the WAN level. The referred authoritative nameserver may be located in a different autonomous system. This authoritative nameserver returns the zonal nameserver for

the domain requested by the client. The client's local name server then queries the requested domain's zonal nameserver which refers the query to the local nameserver for the requested domain. This referred zonal nameserver may be located in a different AS. The referred local nameserver (may be located in a different AS) returns an IP address (i.e., the final destination for client request) for a content hosting server holding the content requested by the client. After receiving an specific IP address for the destination (i.e., the content hosting server) the client's local nameserver informs the client about the new destination IP address. The client request is then routed to this IP address which may also be located in a different AS. In comparison, in the VCN-based routing, the request is encapsulated and routed across the VCN which overlaps the 3-tiers of the network topology. Figure 6.1(b) shows a worst case in which the request is processed by maximum two CERs and two CSRs in our simulations. The CERs are at the MAN level and are the gateways to the client and hosting server site. The CSRs are strategically located in the WAN level and connects the different MANs connected to the LANs in which the client and server resides. The name resolution in VCN, however, assumes, that the CSRs placed in different autonomous systems have a all to all connection.

The scalability and performance of the DNS-based name resolution scheme is mainly dependent on the manually delegated hierarchical namespace and caching of DNS decisions [CoH01]. If the size of the namespace increases, the load on the DNS nameservers also increase and the performance degrades. Parsing of the variable length request packets incur considerable delay in decision making at the nameservers. Caching of DNS

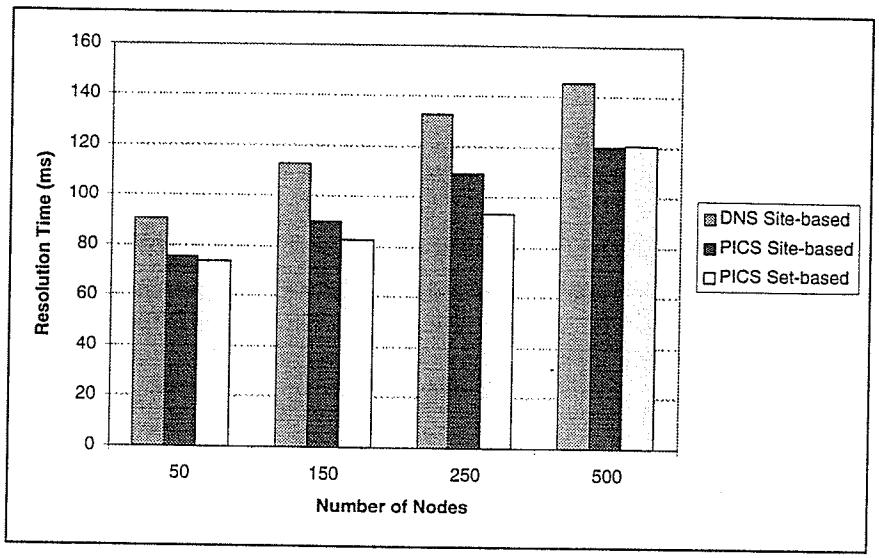

Figure 6.2: Resolution time with 0% replication and 10,000 requests.

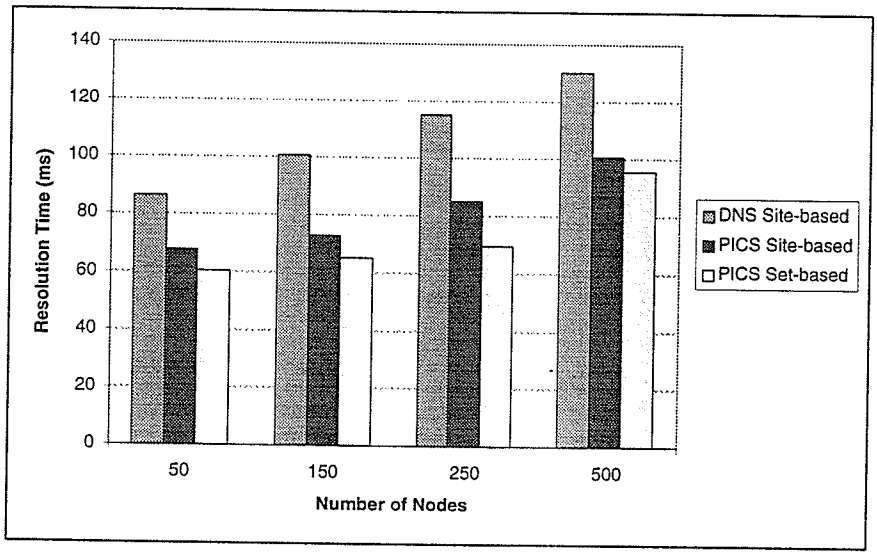

Figure 6.3: Resolution time with 5% replication and 10,000 requests.

decisions also effects the performance and scalability of the DNS. In case of a CDN using DNS-based name resolution, DNS caching should be negligible but in practice studies prove that DNS caching improves the name resolution latency to an large extent. We

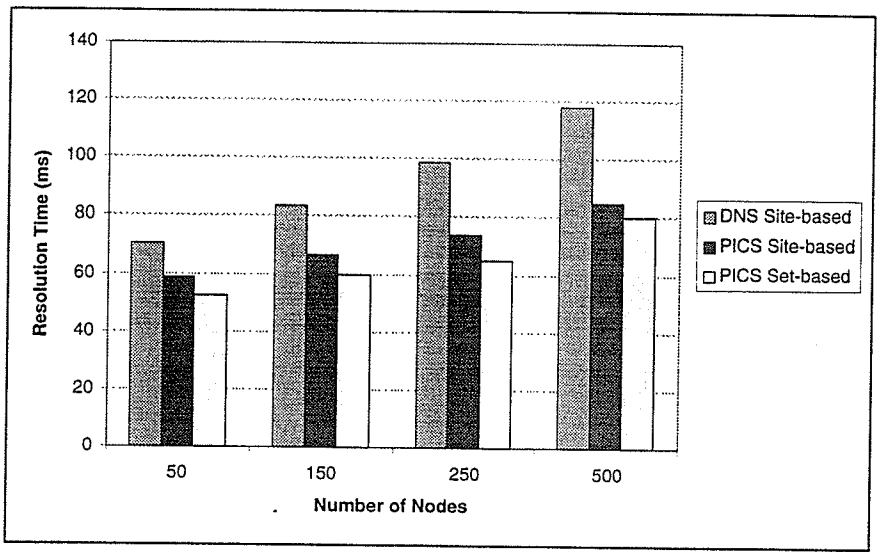

Figure 6.4: Resolution time with 25% replication and 10,000 requests.

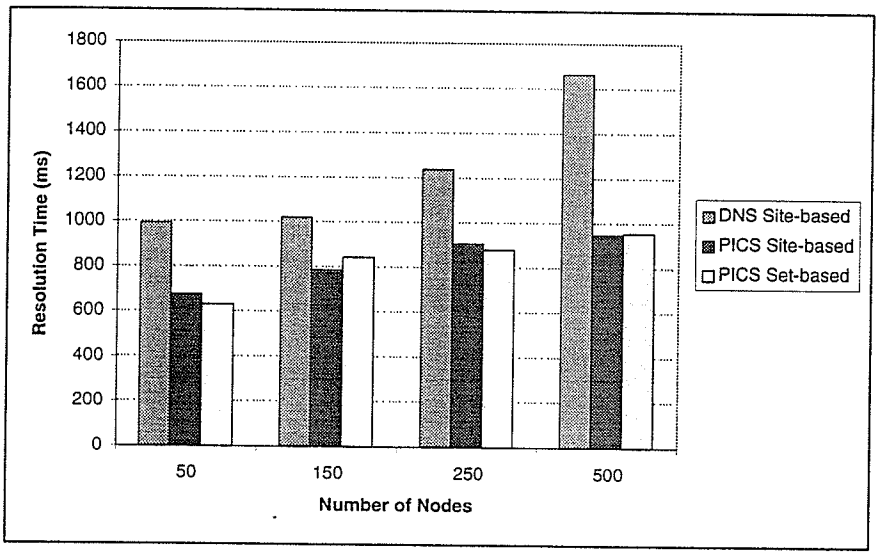

Figure 6.5: Resolution time with 0% replication and 100,000 requests.

intend to include the caching of DNS decision in the future versions of our simulator.

The VCN-based routing scheme depends on the content derived tag space generated for all documents subscribed to the VCN. The content-to-tag bindings for the VCN

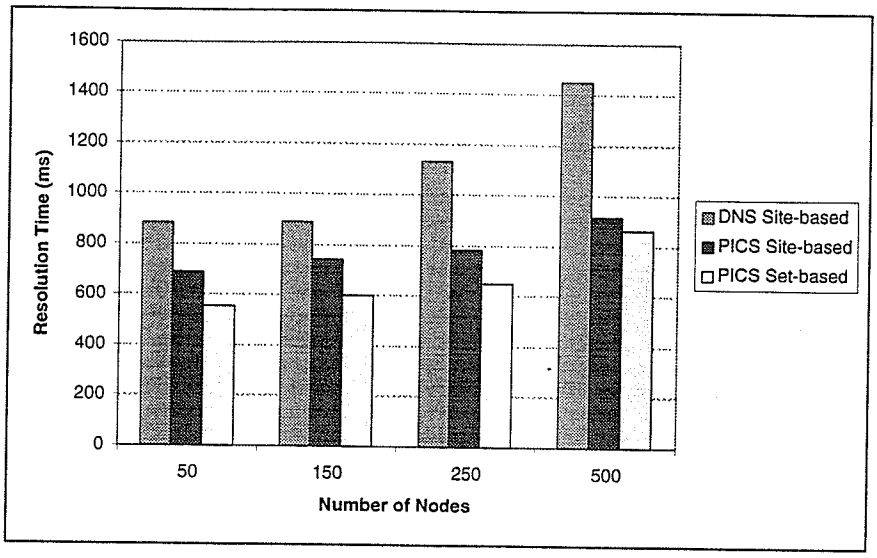

Figure 6.6: Resolution time with 5% replication and 100,000 requests.

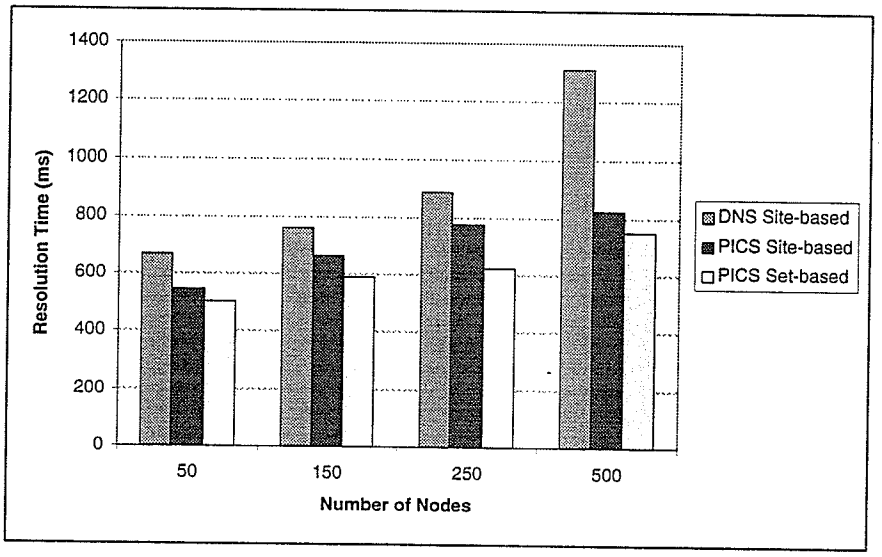

Figure 6.7: Resolution time with 25% replication and 100,000 requests.

increase with the increase in the number of distinct documents subscribed to the VCN. However, this overhead is mostly limited to the edge of the VCN. In the VCN core, the total number of tags is limited to an finite number at all times. Moreover, the packet

parsing time in the VCN core reduced as the CSRs use only fixed length the tag in the content header to select a suitable the next hop. This way the routing decision latency increase at the edge of the VCN but does not effect the performance at the core. Scalability issue in the VCN is addressed by the content classification process which allocates a finite and manageable number of distinct tags for very large data sets.

6.2.2 Content Delivery Time

In this section we obtain the results for deliverying the content. The figures 6.8, 6.9 and 6.10 shows the delivery time obtained with a 10,000 requests generated per LAN. For no content replica, we see an reduction of 24% in content delivery time for both PICS site and set-based schemes. For no replicas, the PICS set and site-based represents the same configuration, meaning there is only one copy of all documents. For a 5% replication, we observe a reduction of 26% for PICS site-based reduction and 33% reduction for set-based replication. For a higher replication of 25%, we observe that PICS site-based replication achieves about 28% reduction in delivery time while set-based achieves an 37% reduction as compared to the DNS-based scheme. We observe an similar set of results using a higer workload of 100,000 requests per LAN, iluustrated by Figures 6.11, 6.12 and 6.13. For no replication we observe an 23% reduction in delivery time for PICS site-based and a 20% reduction for set-based. In practice, both the PICS set and site-based replication should be showing an equal performance with no replication of content. When the replication of content is increased to 5%, we observe that the PICS site-based replication scheme

reduces by 24% and set-bsed reduces by 31%. For a 25% replication, PICS site-based achieves a 28% reduction and set-based achieves an 33% rreduction in content delivery as compared to the DNS-based scheme.

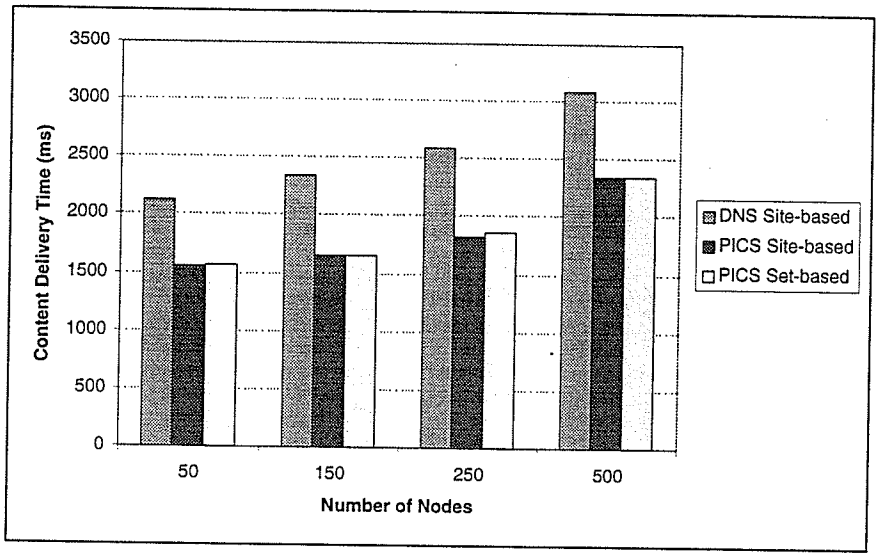

Figure 6.8: Content Delivery time with 0% replication and 10,000 requests.

We attribute the inprovement (i.e., reduction) in delivery time for VCN-based routing to the fact that the PICS forwarding protocol has the ability to perform resource management of the content delivery network more efficiently than the DNS-based routing schemes. In case of the DNS-based scheme, mostly the closest content hosting server with least load is chosen. However, two key issues are involved in DNS-based routing. Firstly, the content hosting server chosen by DNS resolution scheme is closer (in terms of network hops) to the client's nameserver but in turn may not be close to the client. Second, the routing decision takes into consideration the content hosting server's load condition but does not take into account the condition of the network between the client

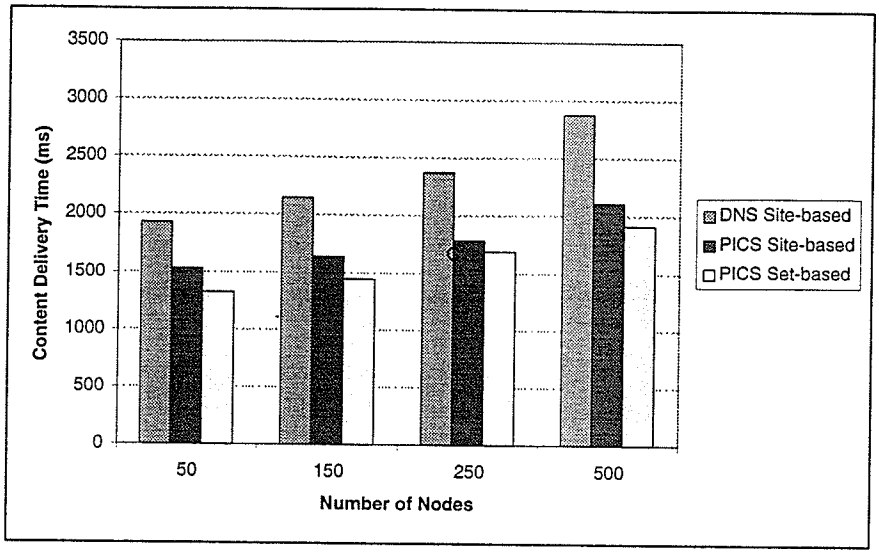

Figure 6.9: Content Delivery time with 5% replication and 10,000 requests.

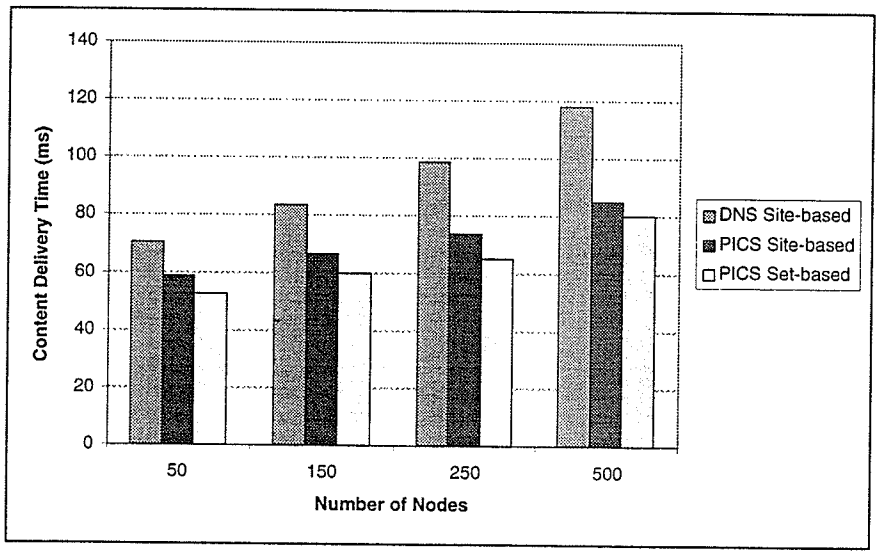

Figure 6.10: Content Delivery time with 25% replication and 10,000 requests.

and content hosting server. Moreover, the routing protocol has no knowledge of the content being delivered. As a result, for a given set of content hosting servers, the physical links leading to these servers may be loaded with connections from preceding requests

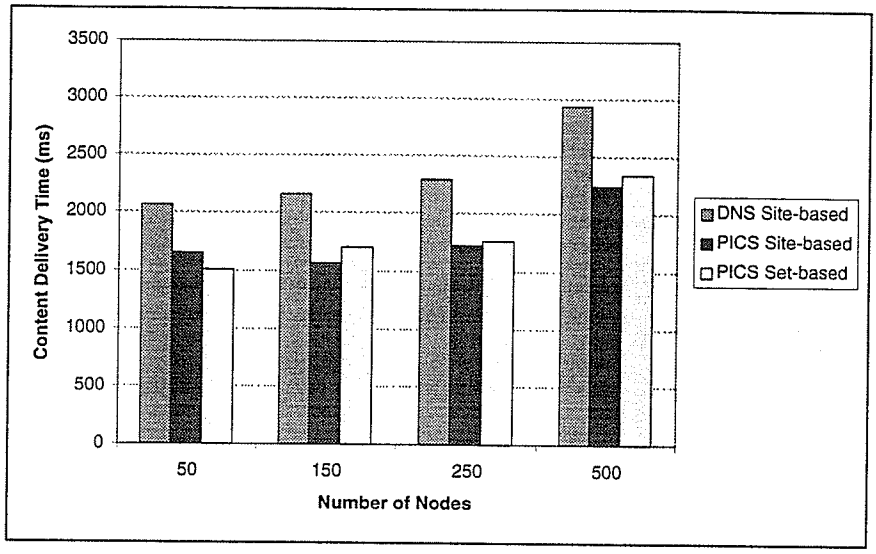

Figure 6.11: Content Delivery time with 0% replication and 100,000 requests.

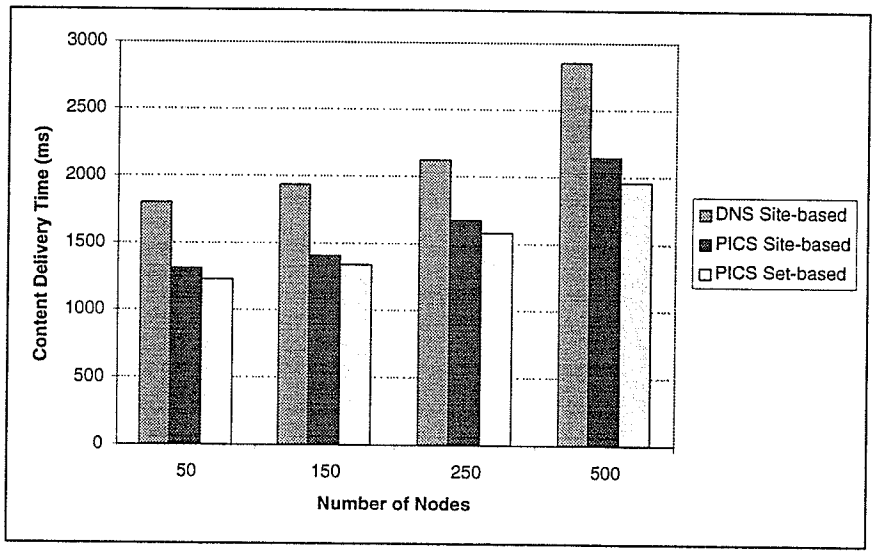

Figure 6.12: Content Delivery time with 5% replication and 100,000 requests.

submitted to the network. This leads to congestion around the links that are common to paths leading to different hosting servers. The result is the extra delay incurred due to congestion while routing requests to the content hosting sites and also while delivering

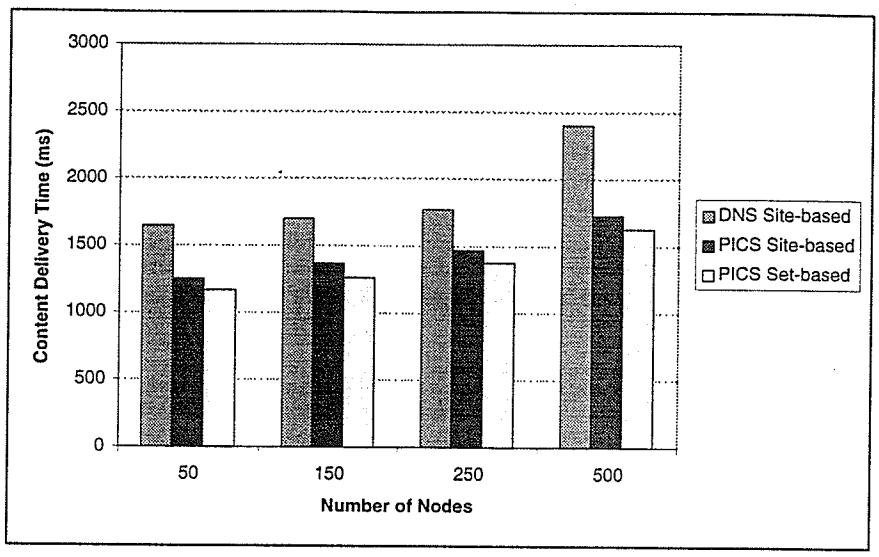

Figure 6.13: Content Delivery time with 25% replication and 100,000 requests.

the content back to the clients. However, in case of the VCN-based routing scheme, the path to the content hosting server from the client is chosen depending on the availability of the resources along the path. The VCN disseminates the server and network status at regular intervals. These status updates reflect the load condition of the hosting server as well as the path leading to the hosting server. The PICS forwarding algorithm use the network status information to ensure the availability of resources immediately or in near future. The status information also helps to balance the delivery load across the different paths that lead to the replicas for a content. Moreover, the knowledge about a content's resource requirements further help the PICS routing protocol to balance the delivery load across the different links. As a result, even though the content hosting site selected by the PICS routing protocol may not be the closest, still the resources are ensured along the path selected for content delivery. This in turn results in continuous delivery of content

in the least possible time. In other words, ensuring the availability of resources along a network path while routing a request to a content hosting server makes it possible to avoid the congestion points while delivering the content to the client. This is not possible in DNS-based routing scheme which takes into account the network condition only after the server is selected.

6.2.3 Throughput

In this section we show our attempts to quantify the throughput of the network for both the routing schemes. Throughput is defined as the number of bytes being transfered by the routing network per unit time. Figures 6.14 and 6.15 shows an experiment setup with no replication. We observe that the network throughput of the set and site-baed replication of PICS are almost same and yield higher results as compared to the DNS-based routing. Figures 6.16 and 6.17 shows an experiment setup for 10,000 requests generated per LAN and Figures 6.18 and 6.19 shows a setup with 100,000 requests generated per LAN. We observe that in almost all cases where content is being replicated, there is steady increase in the network throughput. For smaller networks (e.g., 50 nodes) we not see any difference but for larger networks. the VCN-based routing scheme always perform better by about 13% to 23% increament in the throughput.

The results clearly impress on the inability of the DNS-based scheme to monitor and control both server and network condition as compared to the VCN-based scheme. We reason that the PICS forwarding protocol is able to control the content flow (and reduce

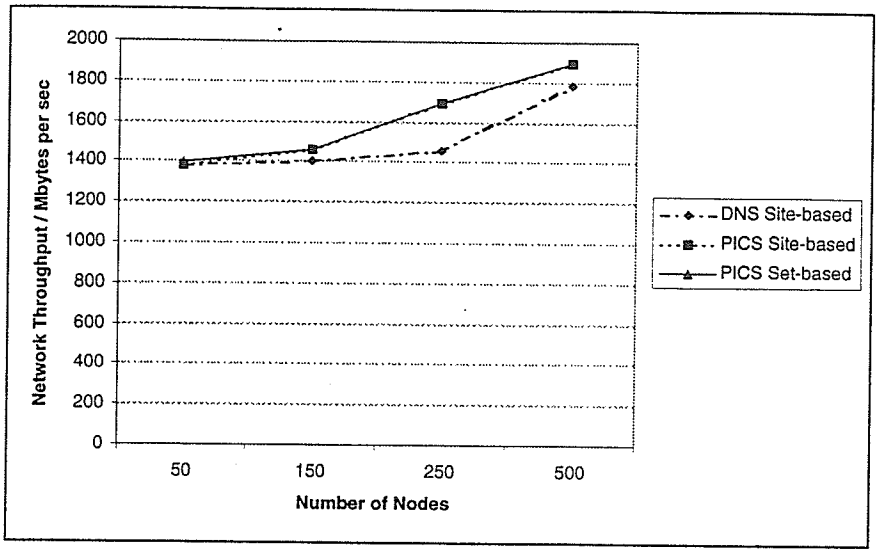

Figure 6.14: Throughput with 0% replication and 10,000 requests.

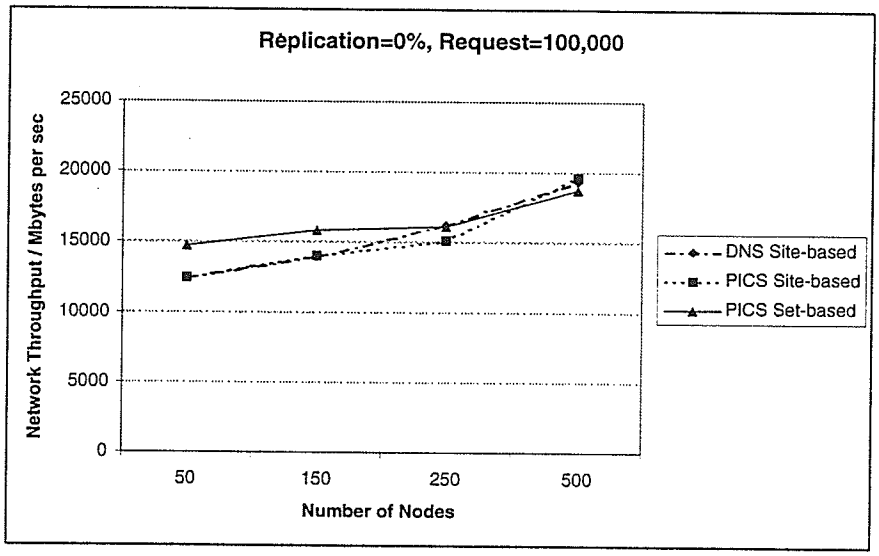

Figure 6.15: Throughput with 0% replication and 100,000 requests.

possibility of congestion) in the network by being able to equally distribute the delivery load across the paths leading to all the replica sites for a content. This is made possible by distributing the network and server status at regular and frequent intervals across the

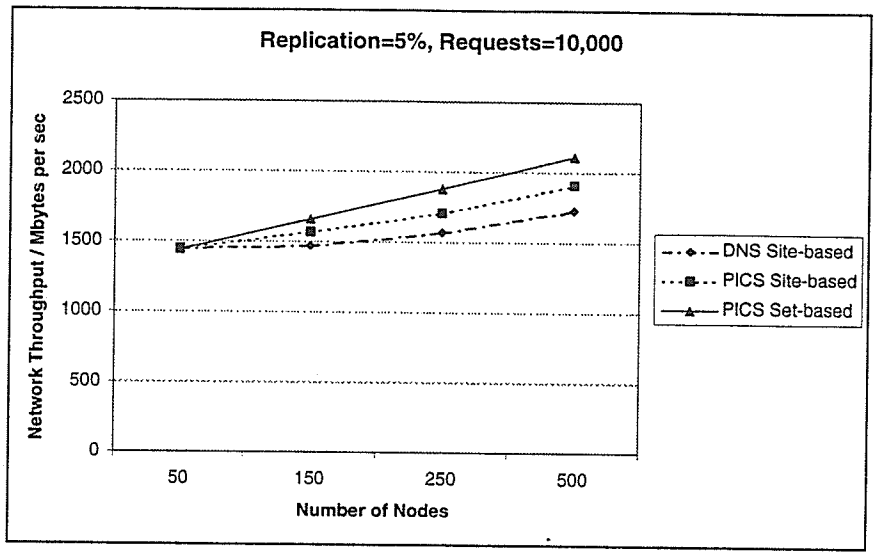

Figure 6.16: Throughput with 5% replication and 10,000 requests.

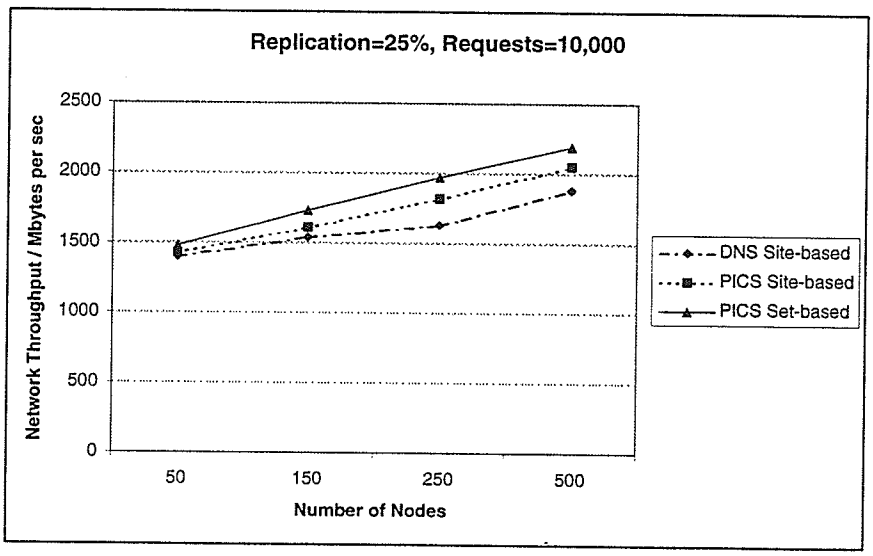

Figure 6.17: Throughput with 25% replication and 10,000 requests.

overlay network. These status updates reflect the server status as well as condition of a path leading to the server. Selecting the path with the best status condition also ensures that a server with low load condition is being selected. This in turn ensures continuous

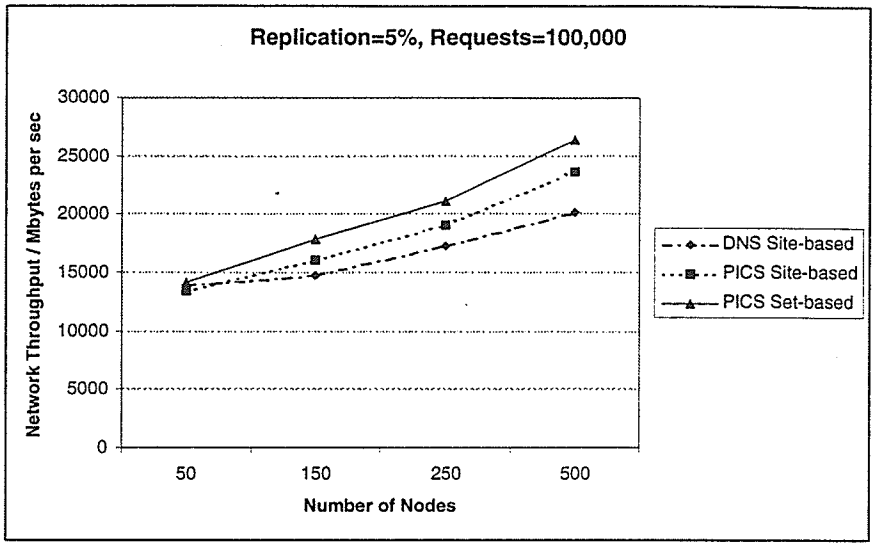

Figure 6.18: Throughput with 5% replication and 100,000 requests.

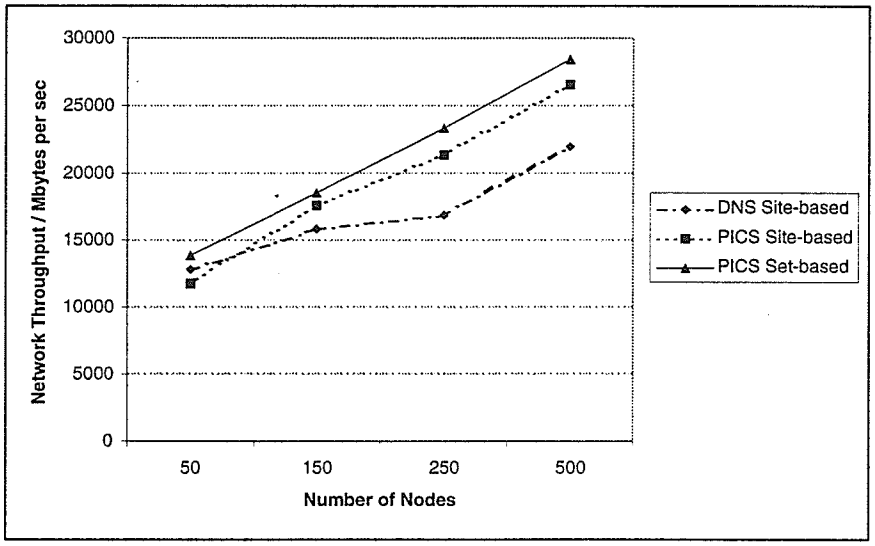

Figure 6.19: Throughput with 25% replication and 100,000 requests.

flow of data in the least possible time. This way, not only the load on the links are equally distributed but also the load on the hosting servers are balanced. In comparison, in the DNS-based routing scheme, a request is resolved to the closest hosting server with

least load. While routing the request, the path to this server is selected by the traditional routing protocols (e.g. OSPF). Mostly, the shortest path is selected. But even though the server is least loaded at the moment, the path between the client and the server is not guaranteed to be the least loaded at the same instant. This is a possibility that definitely increases the delivery time as compared to the VCN which guarantees the best path to the best replica at any given time.

6.2.4 Link Utilization

In this section, we compare the utilization of the network links for both the routing schemes. Figures 6.20, 6.21, 6.20 and 6.21 shows an setup with no replication. We observe that as expected, link utilization of the VCN-based schemes are almost equal for all size of networks. Figures, 6.22, 6.23, 6.24 and 6.25 illustrates the utilization of active links and shows an approximately 31% to 50% increase. Figures, 6.28, 6.29, 6.30 and 6.31 illustrates the utilization of active links and shows an similar increase. However, the graphs show that the utilization of all links reduce by 8% to 10% for VCN-based routing and an approximate 10% to 16% reduction for DNS-based routing. In either routing schemes, not all links present in the network are used by the routing protocols. But the results indicate that the VCN is able distribute the delivery load by using more number of links for content delivery than the DNS-based routing. This means that the VCN is able to find alternate paths for the requests, where as the DNS-based routing use only a selected set of network paths and thereby lead to congestion much faster than the VCN.

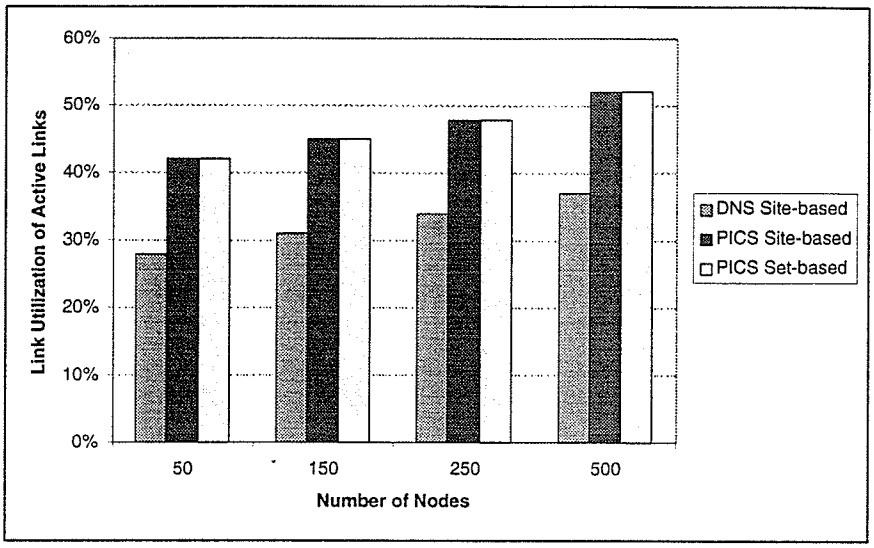

Figure 6.20: Link utilization for active links with 0% replication and 10,000 requests.

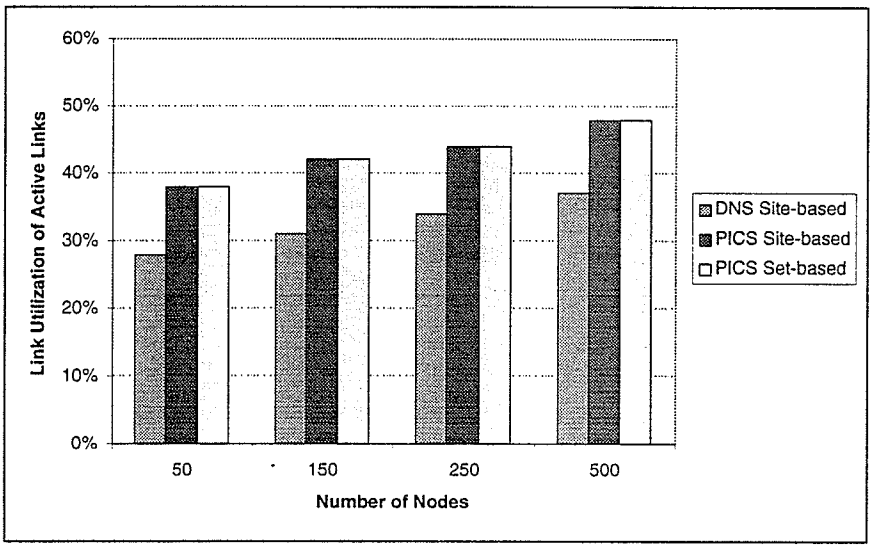

Figure 6.21: Link utilization for active links with 0% replication and 100,000 requests.

The improvement in the utilization of links for the VCN in comparison to the DNS-based scheme may be attributed to the reasons stated in the previous subsection. The VCN-based routing scheme chooses the best path and in turn a considerably less loaded

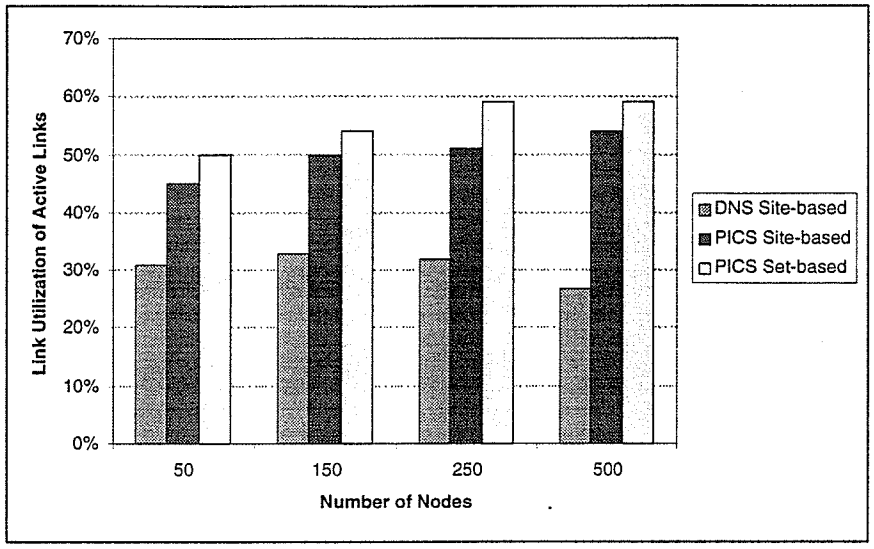

Figure 6.22: Link utilization for active links with 5% replication and 10,000 requests.

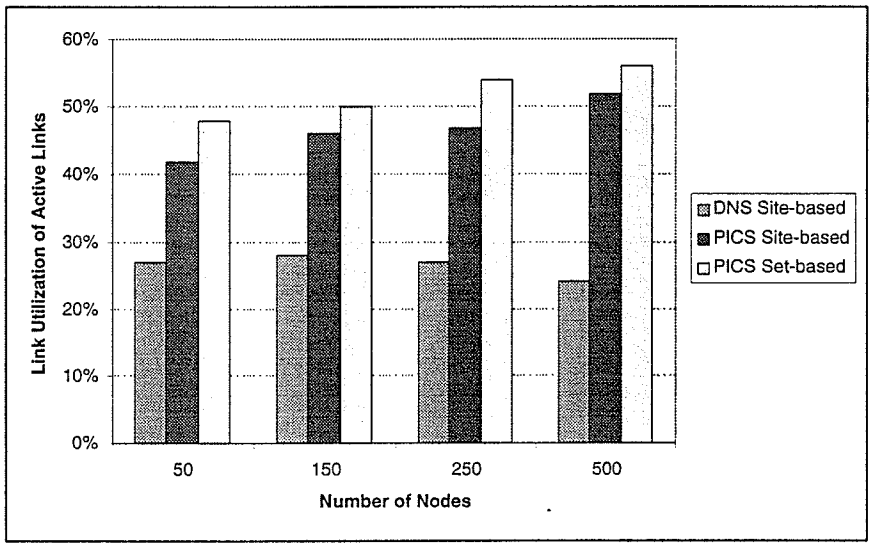

Figure 6.23: Link utilization for active links with 5% replication and 100,000 requests.

server for content delivery. This way, it is possible to distribute the delivery load in the most cost effective manner (i.e., load is distributed across the paths to all the replicas of a content in almost equal proportions) for all the links in the network. On the other

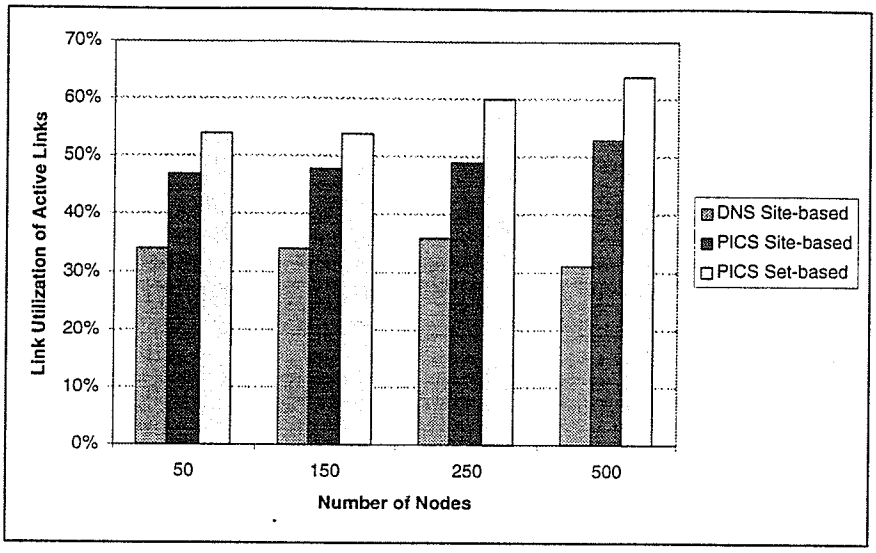

Figure 6.24: Link utilization for active links with 25% replication and 10,000 requests.

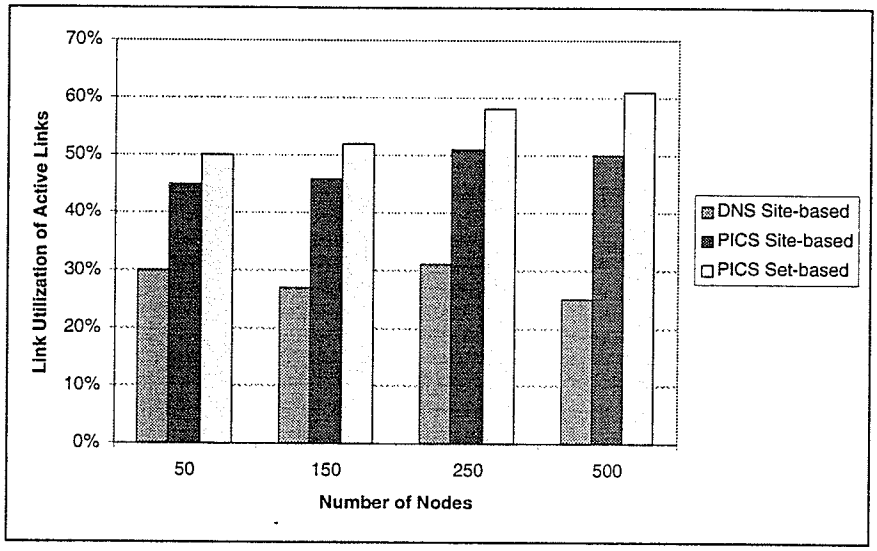

Figure 6.25: Link utilization for active links with 25% replication and 100,000 requests.

hand, DNS-based routing first chooses the best server and then tries to find the best path to the selected server. This way it is not always possible to select the best path or distribute the delivery load in equal proportions along the network links. In turn this

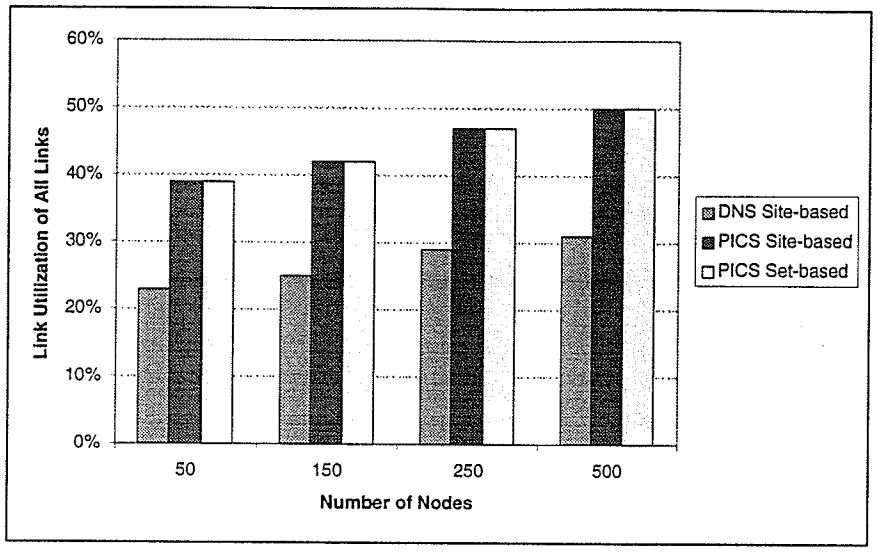

Figure 6.26: Link utilization for all links in the network with 0% replication and 10,000 requests.

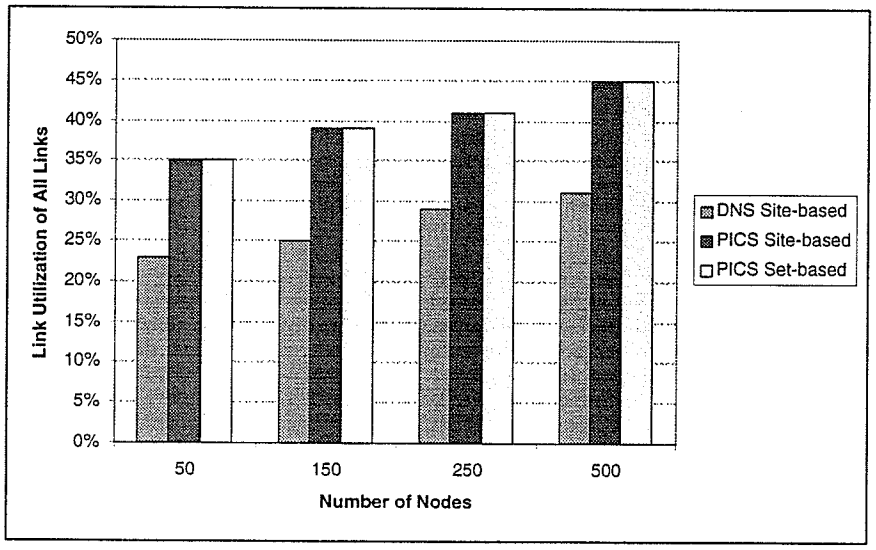

Figure 6.27: Link utilization for all links in the network with 0% replication and 100,000 requests.

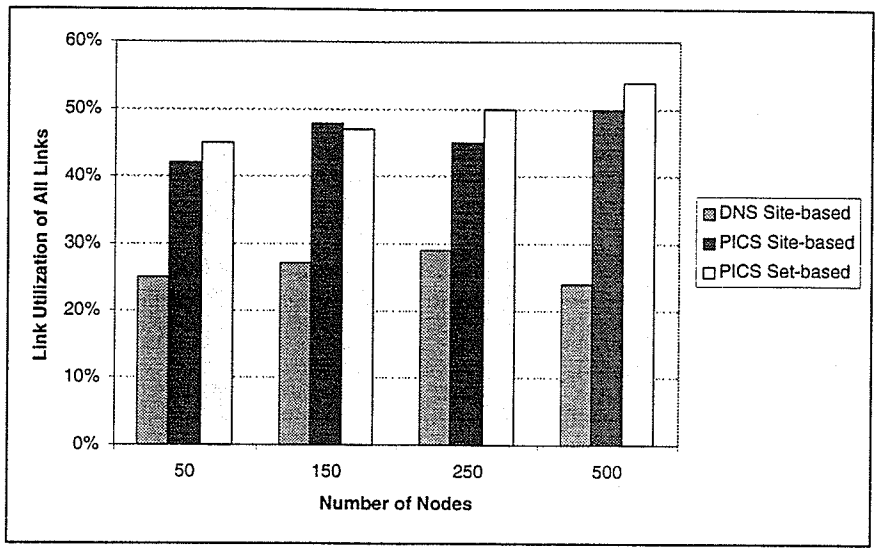

Figure 6.28: Link utilization for all links in the network with 5% replication and 10,000 requests.

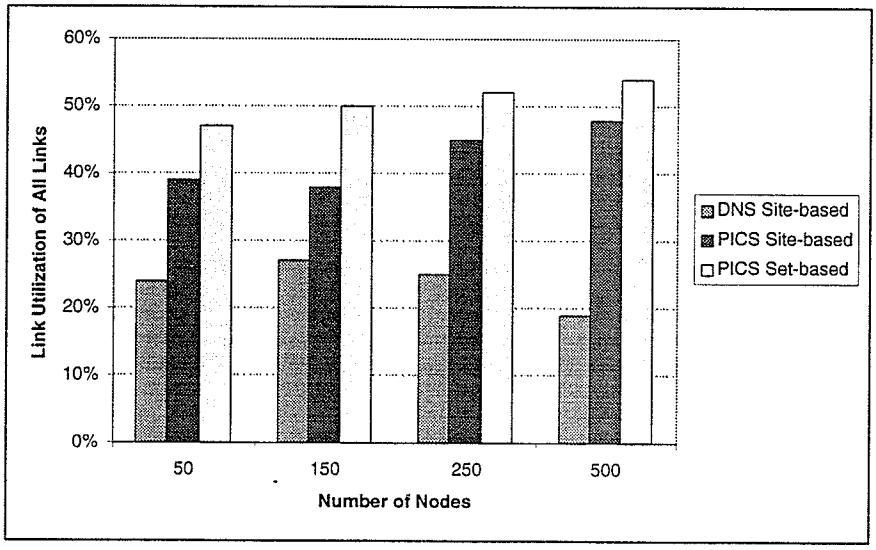

Figure 6.29: Link utilization for all links in the network with 5% replication and 100,000 requests.

results in congestion of the prime links, which connect multiple content servers to the network and acts as the bottleneck for rest of the network. If the most of the content replicas are located in the same section of the network (e.g., same LAN) then the VCN is able to identify the conditions of the path to those and link and distribute the entire delivery load among the replicas. The DNS, however, identifies the server status. It cannot identify the LAN's load condition while selecting a path. Such a case definitely leads to congestion for the DNS-based routing scheme,

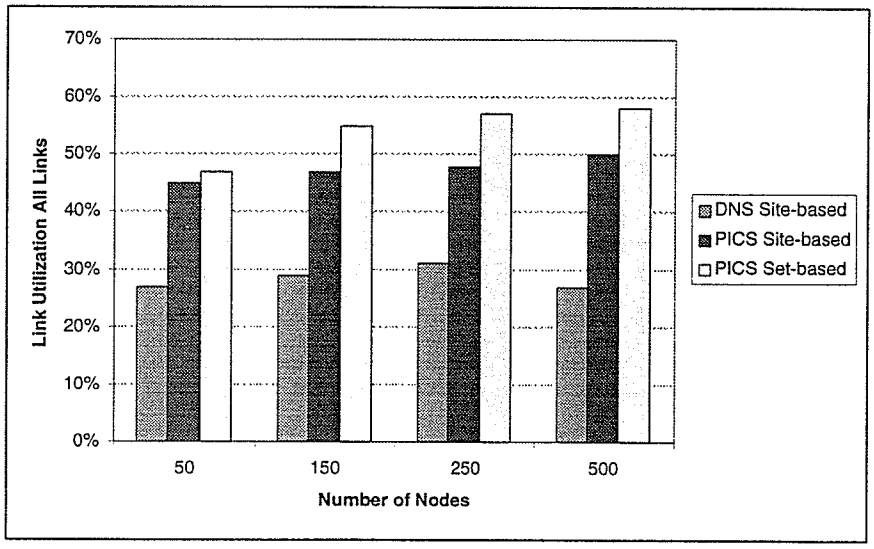

Figure 6.30: Link utilization for all links in the network with 25% replication and 10,000 requests.

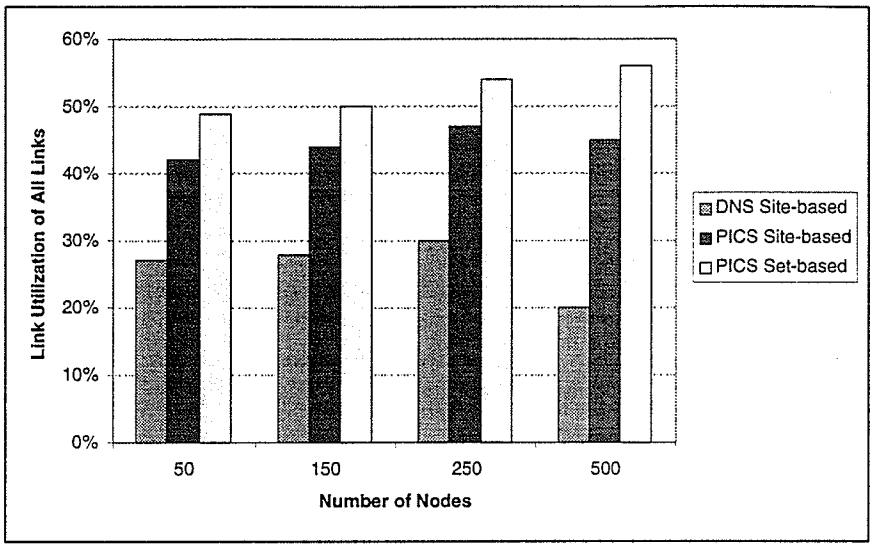

Figure 6.31: Link utilization for all links in the network with 25% replication and 100,000 requests.

6.2.5 Impact of Streaming and Non-streaming Documents in PICS

In the earlier subsections, we presented the results of experiments which were conducted with non-streaming content. In this subsection, we compare the impact of delivering both non-streaming and streaming content using the PICS routing protocol. Figures 6.32, 6.33, 6.34, 6.35, 6.36 and 6.37 shows the results obtained from the set-based replication. Figures 6.38, 6.39, 6.40, 6.41, 6.42 and 6.43 shows the results obtained from the site-based replication. The results show a increase in throughput for the VCN site and set-based replication in almost all cases. However, our simulations results observed approximately 12% of the requests for streaming content were rejected by the servers.

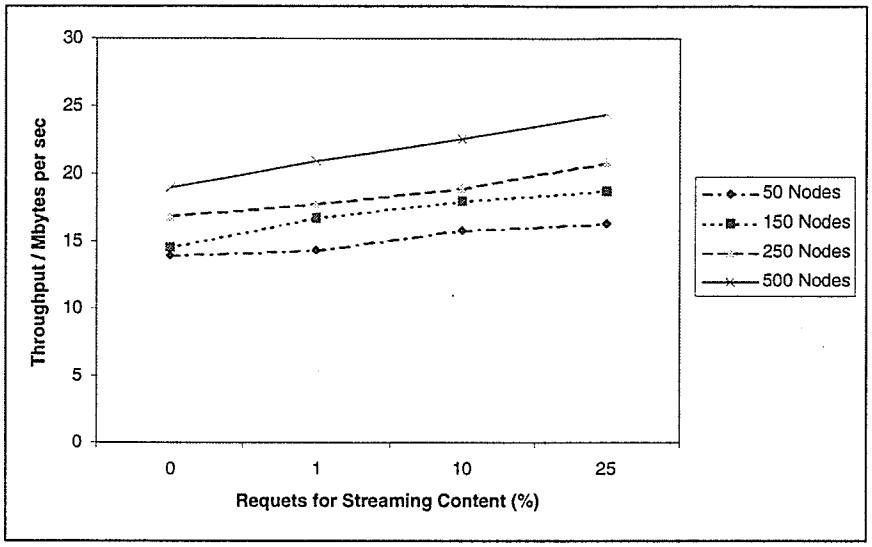

Figure 6.32: Impact of mixed traffic with set-based 0% replication and 10,000 requests.

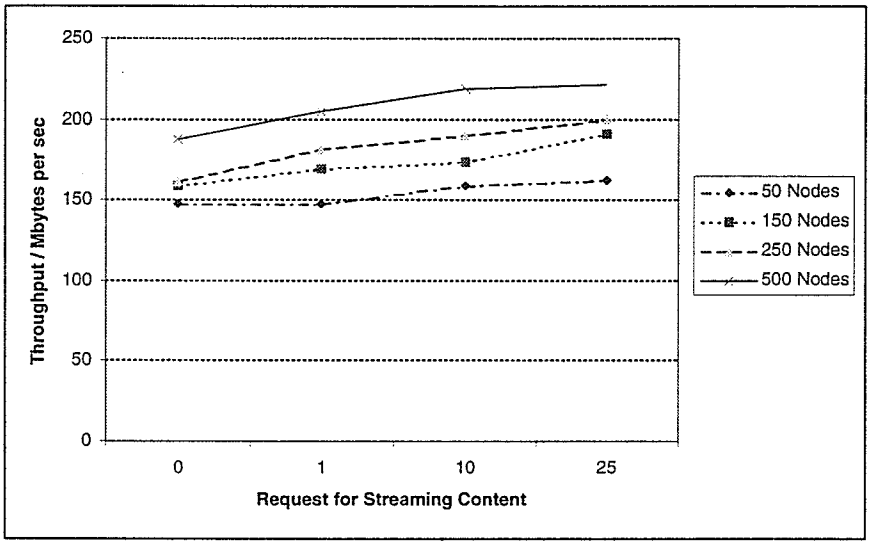

Figure 6.33: Impact of mixed traffic with set-based 0% replication and 100,000 requests.

Streaming content require a fixed amount of bandwidth for delivery for the duration of its download (i.e., the playing time of the audio/video content). The PICS forwarding algorithm limits the maximum bandwidth on each link that can be used by the streaming

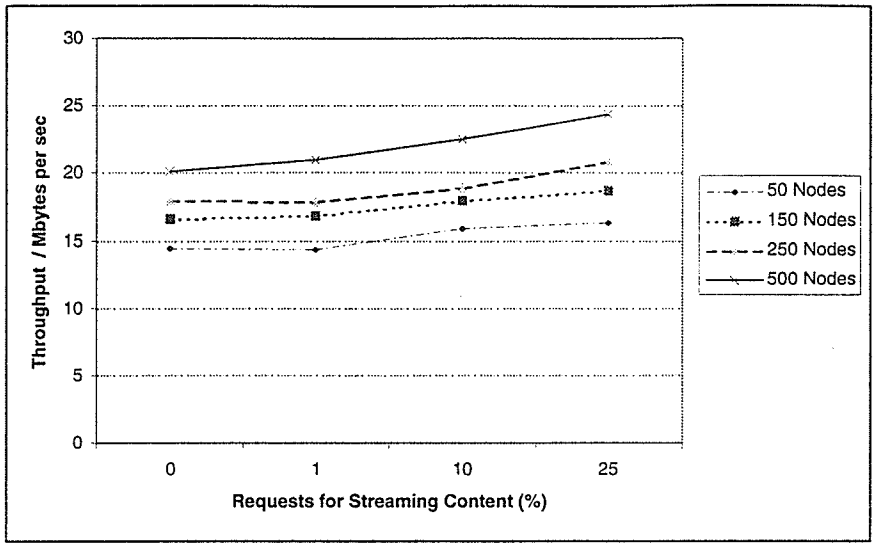

Figure 6.34: Impact of mixed traffic with set-based 5% replication and 10,000 requests.

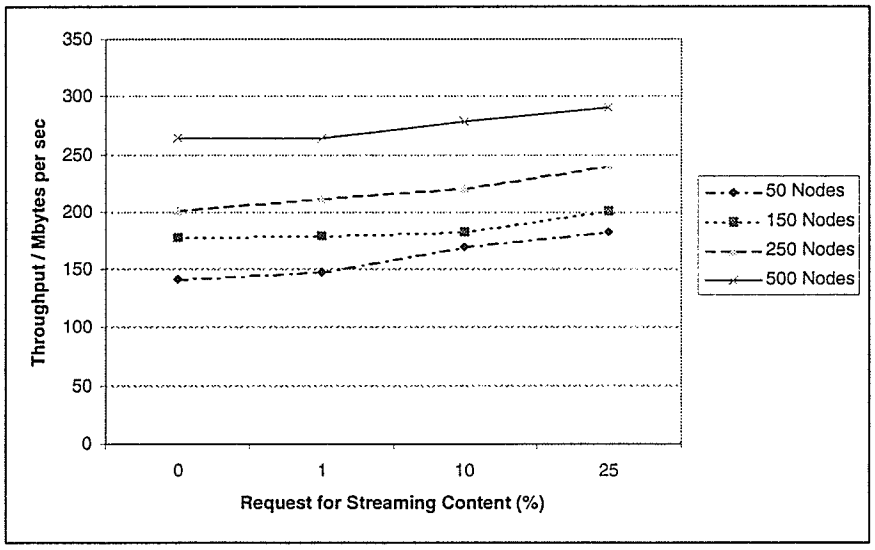

Figure 6.35: Impact of mixed traffic with set-based 5% replication and 100,000 requests.

contents for delivery. For the simulations, this limit is restricted to 20% of the link bandwidth. All connections for streaming content on a particular link use maximum 20% of the bandwidth on that link for content delivery. If the number of such requests

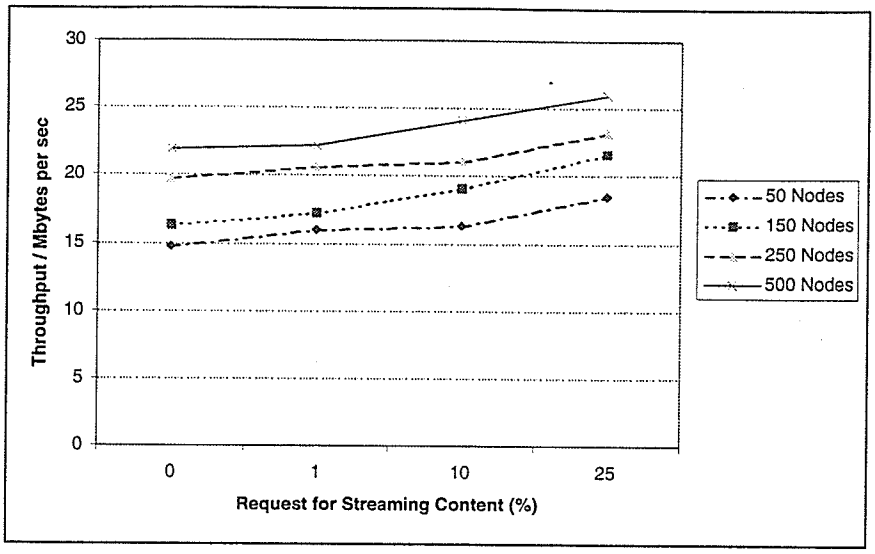

Figure 6.36: Impact of mixed traffic with set-based 25% replication and 10,000 requests.

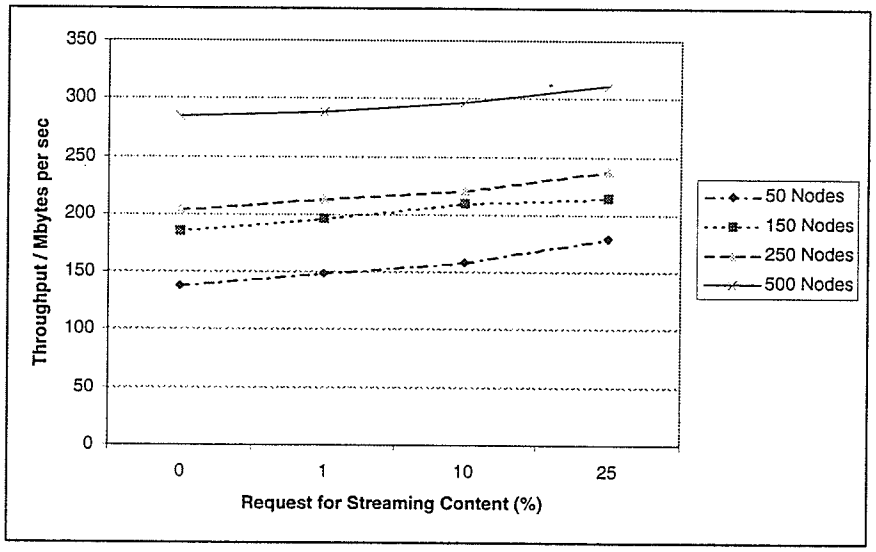

Figure 6.37: Impact of mixed traffic with set-based 25% replication and 100,000 requests.

increase such that 20% bandwidth limit is no longer sufficient to support delivery of all the streaming requests, then the requests for streaming content are rejected. The results show that in case of a mixed traffic (i.e., of streaming and non-streaming content), by

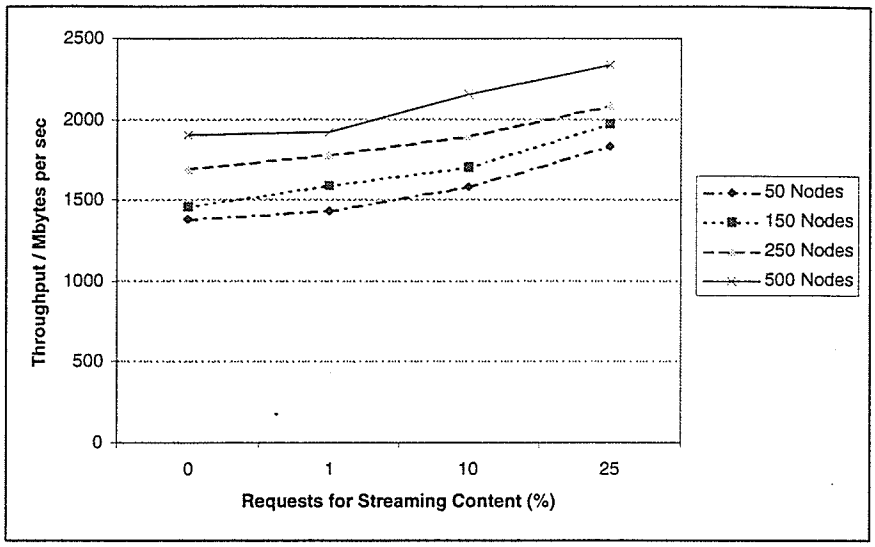

Figure 6.38: Impact of mixed traffic with site-based 0% replication and 10,000 requests.

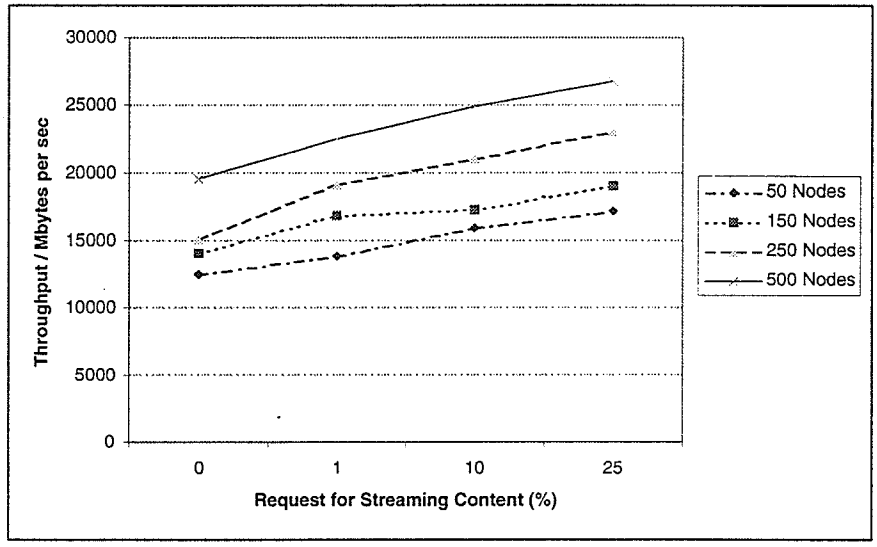

Figure 6.39: Impact of mixed traffic with site-based 0% replication and 10,000 requests.

rejecting the requests the throughput can be increased by approximately 15% to 20%. The rejection of some requests ensure bandwidth availability for other requests. The remaining 80% of the link bandwidth on each link are shared by the requests for non-

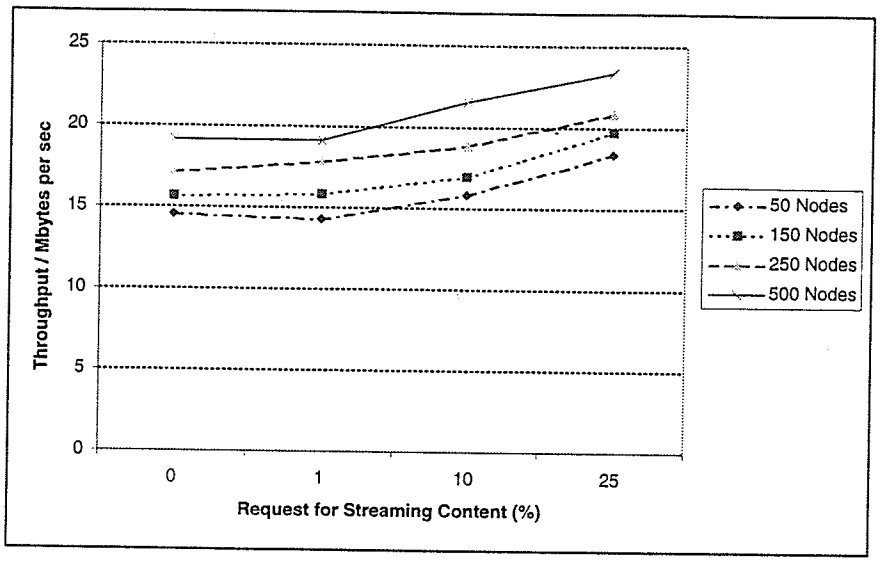

Figure 6.40: Impact of mixed traffic with site-based 5% replication and 10,000 requests.

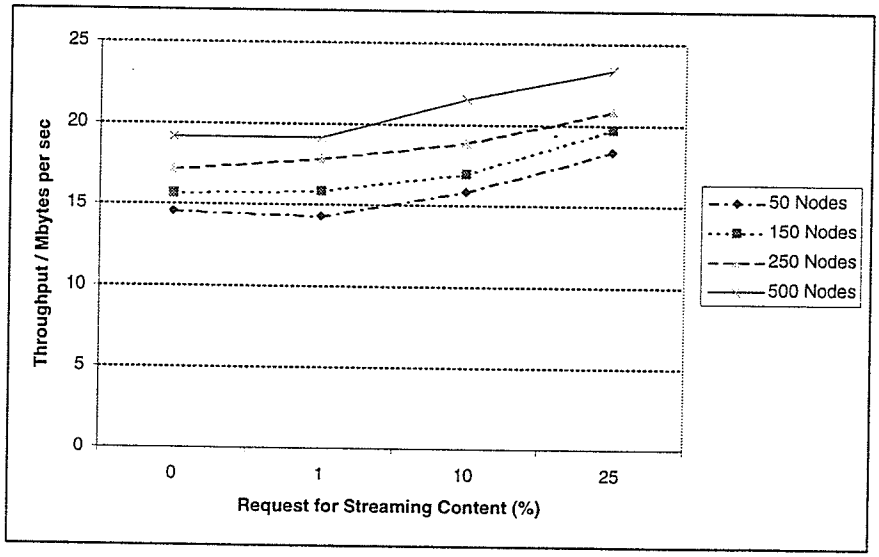

Figure 6.41: Impact of mixed traffic with site-based 5% replication and 10,000 requests.

streaming content. The results show, that this technique used by PICS is quite effective in handling requests for mixed traffic.

In practice, the amount of link bandwidth that should be reserved for streaming

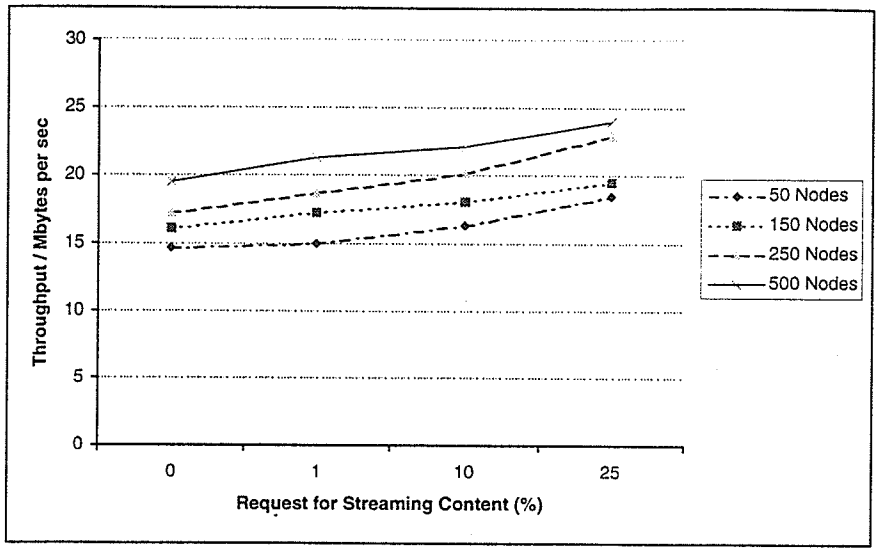

Figure 6.42: Impact of mixed traffic with site-based 25% replication and 10,000 requests.

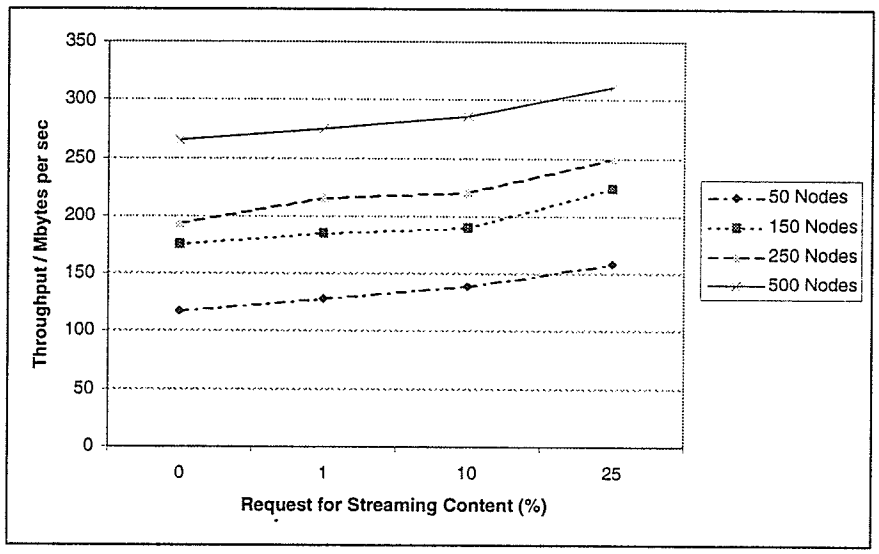

Figure 6.43: Impact of mixed traffic with site-based 25% replication and 100,000 requests.

content should be determined from the pattern of the link usage obtained from prior traffic results. Our simulator lacks such a traffic analysis mechanism. For simplicity, we set this limit to 20% of the total bandwidth for each link in all our simulations. For our

future works, we plan to develop a traffic manager for collection and analysis of VCN traffic to investigate the flow of content in a real time scenario.

Chapter 7

Applications of Content-based Routing

7.1 Overview

As explained earlier, the benefits of the proposed content-routing architecture include: (a) ability to scale content-based routing to span geographically distributed servers and clients, (b) flexibility to handle content with wider range of popularity and hence different levels of replication, (c) ability to detect surges in demand due to increases in popularity, and (d) ability to detect and contain certain types of intrusion and other malicious activities. Below we briefly discuss some of the applications of the proposed content-routing architecture.

7.2 Content Delivery Networks

The *Content Delivery Networks* (CDNs) are emerging as popular alternatives for conventional web caching for disseminating important content on the Internet. The popularity of the CDNs is primarily driven by the desire of the "service originators" to control QoS that is delivered to the end user. The content-based routing architecture discussed here can be used to implement scalable CDNs. The VCN can provide a way of discovering and accessing the available content on the network. The content-based routing infrastructure provided by the VCN is one part of a CDN. CDNs would require support for acquiring and managing resources that are required for storing and delivering the content at locations that can satisfy most of the demand.

7.3 Scalable Service Deployment on the Internet

The CDNs are a special case for scalable network services where the service is focused on delivering the content. In general, an Internet service may want to scale itself up or down depending on the demand. Further, the service should be replicated and/or migrated to locations that are close to the "demand points" for the service. Once the service is replicated or migrated, it is essential to redirect the client side requests to the best replica of the service provider so that replication could provide the benefit to the end user. The VCN can be used in this aspect of scalable service deployment. In particular, the scalability of our content-routing architecture enables the deployment of

scalable wide-area services.

7.4 Video-on-Demand Services

Video-on-Demand (VoD) is another application that could benefit from a content-based routing scheme such as the one proposed in this paper. One of the goals of a VoD system is to deliver videos in a wide area whenever users want to watch them. In order to make the deployment economically viable it is necessary for the deployment to span a large user population on a wide network area. One of the major issues with VoD is serving clients that submit their requests in an uncoordinated fashion while conserving the server and network resources. Content-based routing may facilitate the deployment of VoD type of systems by providing a transparent technique for redirecting requests. The redirections may be initiated based on where and when the requests are coming from.

7.5 Intrusion Detection and Security Systems

A network intrusion detection system monitors the packets in a network and attempts to determine whether an intruder (e.g., hacker) is trying to break in to a system or trying to cause a denial of service attack. Content-based routing helps to improve upon the intrusion detection through a more exhaustive form of packet parsing. Such a technique enables an ID system to look at the entire content of a packet stream and then identify any signature or string that can force the host to release information about its configuration

or provide protection against insertion and evasion attacks [PtT98]. By looking at a packet stream in the same way as the end-hosts (i.e., the servers) the ID system deployed on the content routers will protect the end-hosts from the exploitation of the TCP/IP protocol suite by DoS attacks. In our method of content-based routing, the network ID system can be deployed on the VCN and use the CERs for packet filtering. Exhaustive packet analysis at the edges of the VCN in conjunction with the traffic measurement and pattern matching techniques used in the core the VCN will help to identify and filter an attack.

Chapter 8

Conclusions and Future Work

In this thesis we have introduced Protocol Independent Content Switching (PICS), a novel method for content discovery and access over geographically distributed networks. This chapter concludes this dissertation with a summary of its contributions and performance benefits, comparison with some of the existing content routing models and the directions for future work.

8.1 Thesis Contribution

The PICS routing scheme creates a content-aware overlay network, called the Virtual Content Network (VCN), on top of the physical routing plane. The VCN acts a virtual content-based router that uses content-based tags to route client requests to the best content hosting server along the best path available to the server. We have developed a discrete event simulator to compare the performance of the proposed PICS routing

scheme with the existing domain name based routing scheme. The simulator is based on a simple analytical model that we have developed to measure the performance of our proposed model. The content-based tags used for routing are generated based on the attribute values of content that will be delivered using the VCN. We introduce a content classification scheme to generate the tags. It uses a content classification scheme to identify the attributes of content from a content-based routing perspective. The routing algorithm uses different metrics like server and network load condition. One important advantage of our proposed architecture is that the tag based routing protocol is not tied down to the underlying architecture. This means that different greedy heuristics can be used for routing.

We have used the simulator to examine different routing scenarios. We found that the PICS routing scheme outperforms the DNS-based routing scheme is almost all the case. The simulations results show that the VCN is able to efficiently handle the flow of content in the routing domain and thereby reduce congestion of the links in the routing network. In the following sections we summarize the PICS design components and discuss its performance benefits.

8.1.1 Architecture and Design Benefits

Content Characterization: Content characterization is a process for identifying the key attributes of content which are used to generate an accurate description of content. The key attributes include both the structural and semantics properties of

content. In general, the content description specifies the resources (e.g., bandwidth) required for delivery of the content. This description includes information about the size and type of the document, popularity of the content in the routing domain and the intended usage of the content by the requesting application. This fact is also reflected by results showing the utilization of the network links.

Content Classification: Content classification is a scheme that uses the content attributes identified by the characterization process to create content classes called *Content Equivalence Classes* (CECs). Each CEC is collection (i.e., aggregate) of documents with similar characteristics. Each CEC is identified by a tag generated based on the characteristics of the constituent documents in the CEC. Since, each CEC identifies a distinct group of contents with a specific set of characteristics, the tag generated for each CEC will be unique within the VCN. The tag is used by the PICS routing algorithm for discovery and access of documents.

Scalability: By creating content classes (or content aggregates) the total number of distinct tags used by the content-based routers in the VCN is always limited to a finite number. Moreover, the content-based tags used to describe the resource requirements of a content requested. This way, the content-based routers only need to maintain the content-based tags. The content description can be inferred from the tags at the time of routing. The VCER tree used by the VCN further helps to distribute the tag management with the VCN. The routing factor used by the PICS scheme indicate load condition of the server as well as the paths (i.e., network

links) leading the server. This information helps the content-based routers to make suitable routing decisions and thereby allow load balancing at the content hosting servers as well as the network links.

Efficiency: The CERs at the edge of the VCN maintain the content-to-tag bindings while the CSRs in the core of the VCN maintain only the tags along with the possible next hops for the tag. The CERs encapsulate a client request with a content header which holds the tag that identifies the CEC holding the content being requested. The CSRs examine the fixed-length tag in content header of the encapsulated packet to find the next hop for the request. This is unlike the traditional routing schemes which examine variable length request for making routing decisions. Parsing of variable length request packets is a time consuming process. Moreover, in traditional routing schemes, the request packets are examined (i.e., parsed) at all routers along the routing path. This increases the overall time required to route client request. In comparison, the fixed length tag-based routing will be much more efficient.

Routing Algorithm: The PICS routing (or forwarding) algorithm use the content-based tags for discovery and access of content. The forwarding decision is based on the resource requirement of the client request and the resource availability in the routing domain. This enables the routing protocol to control the content flow within the routing domain. It is be able to reduce the bandwidth starvation (i.e., congestion) in the network much more efficiently than other routing schemes which

route depending only on the location information and ignores of the resource requirements of the content. This is evident from our simulations results which shows that PICS routing protocol considerably improves the resolution time, content delivery time, throughput and link utilization of the network. The results impress on the fact that content-aware networking is much more scalable and efficient in terms of network resource management in comparison to the existing DNS-based routing schemes.

8.1.2 Performance Benefits: VCN vs DNS Based Routing

In a practical routing scenario, the true challenge is not only to select an optimal server but the path chosen to deliver data from the server should also be optimal in comparison to other servers and paths in the routing network. Looking at the simulations results, we can conclude that the VCN-based routing scheme is able to choose best server and the best path to the server more efficiently than the traditional DNS-based routing scheme, even though the selected path and the server may not be the closest to the client. But the guaranteed resource availability along the path to the server ensures the delivery of content in the least possible time. On the other hand, the DNS scheme picks the best server closest to the client. But the path to this server is not always optimal in terms of resource availability.

The improvement in the throughput for the VCN, for larger networks, shows that the VCN is able to control the data flow and reduce congestion in the routing network quite

efficiently. For smaller networks, however, we do not see any significant improvement in network efficiency. This is because, the VCN tries to balance the delivery load equally among all the available paths. For smaller networks, the number of such available paths are limited and thus leads to congestion while handling the client requests. However, for larger networks, the VCN is able to spread the delivery load across the entire network thereby reducing the congestion of individual links in the network. This fact is also reflected by the percent of utilization of the network links. For the VCN, we see that almost all the paths that lead to a replica are being utilized by the routing protocol. The DNS, on the other hand, tries to use only a selected number of paths and replicas for serving the clients. This most certainly leads to congestion.

Although, the VCN mostly performs well at the content delivery task, we observed that in most of the cases, the VCN selects an server that is not closest to the client, even though there may be a replica of the requested content closer to the client than the selected server and is capable of serving more requests. This is a clear disadvantage of the routing heuristics followed by our content-based routers. Our content-based routers performs very well in balancing equally the content delivery load across all the available paths and content replicas an any given time. However, one may argue that it will be more optimal to serve a request from a replica closer to the client and is not highly congested rather than traversing the longer network paths trying to balance the load on the paths. We are currently working to make the content routers use different routing heuristics depending on the condition of the hosting servers and network congestion factor.

One prime motivation for the proposed content-based routing scheme was to reduce the overall time required to serve a client with respect to the existing domain name based routing schemes. From the resolution time and the content delivery time obtained from the simulations, we can conclude that VCN-based load balancing of the hosting servers and the network links yields better content download time than traditional DNS-based routing scheme. Our experiments to examine the impact of streaming media content on the VCN showed that the VCN is able to handle requests for streaming content quite effectively, even though we see a moderate percent of streaming connections being rejected by the VCN. We conclude that the VCN has the capability to consistently maintain an high level of network efficiency even though at the cost of losing certain amount of revenue by rejecting the client requests. However, at any given instant the routing protocol at least guarantees that the majority of clients are served rather than losing a larger number of customers due to the congestion caused if trying to serve all the requests.

8.1.3 Next Generation Networking

The content-based tags generated from content characterization and classification schemes lay the foundation for the next generation of routing techniques. We conjecture, the next generation of routing techniques will route requests based on the data being transfered rather than any specialized addressing and routing information attached to, or otherwise associated with, the data. The tag based routing used by the VCN is the first step towards such an IP less routing architecture. Routing in this way will bear the least or

almost no relationship to the hierarchical structure of the underlying transport network. Advanced content characterization and classification schemes will help to develop an accurate description of content and lead to more optimized tagging and routing schemes.

8.2 Comparison with Existing Routing Methods

Similar to the INS [AdS99] name specifiers, we create content based tags for data sets served by content providers. However, unlike the attribute based naming, content tags in our architecture are derived from the structural and semantical properties of the content (explained in Section 3.4). Further, in our approach, the routing information is disseminated in a directed "feed-forward" manner unlike the flooding-based dissemination process used in the INS. Use of *content equivalent classes* further enhances content aggregation as contents or services can be advertised as a group. Our content-based routing process performs tag matching once the tag is derived. This is expected to be more efficient than "tree walking" at each router.

In comparison to the NBRP [GrC01], we have further reduced the content delivery time by combining the name resolution and content access methods. We use a portion of the URL namespace (and some other attributes of contents, as explained in later sections) to create fixed length tags that are used to encapsulate the packets. The tag is then used to route the packets across the VCN. We expect that the cost of transferring the encapsulated packet will be less or will balance the RTT time incurred by the traditional methods. Also, the encapsulation is created independent of the Internet protocols and is

not affected by the different subnets the encapsulation has to traverse. The usage of URL namespace increases the granularity of access as compared to the NBRP. Further, our method creates content equivalence classes that are used to compose content-aggregates which helps in managing very large sets of data.

Unlike the CRP [MiN00], our routing scheme allows routing at the granularity of URL but also provides a solution for a more comprehensive form of name-based routing using content derived tags and virtual content routers. The use of fixed length tags reduce the complexity and cost incurred due to walking the namespace tree at each router/cache. Further, the content equivalence classes used by our scheme helps in managing large sets data.

Similar to CAN [RaF01], our routing protocol eliminates the cost of parsing and identifying the content at each content router. However, in comparison, our proposed architecture performs a deeper study of the content to describe contents and create fixed length tags that are used to locate and access the content. Our method provides a more scalable and generalized scheme for partitioning content hosting sites and handling large volumes of data by creating content equivalence classes.

The routing technique used by Pastry [RoD01] does not necessarily select the shortest path nor the best path from a source to a destination. It does not take into account the condition of the network between the next hops while routing a message. However, it generates a loop free routing path where a message will always be forwarded along a path with no possibility of returning to any of the nodes it had crossed along the routing

path. A Pastry node uses only the local routing information to route a message. This is in contrast to our routing technique which uses a global routing metric and will always ensure the best path (in terms of resource availability) being chosen between the source and destination. This method, however, incurs an extra overhead for disseminating the routing information globally within the routing domain. But using suitable optimization schemes, this overhead can be minimized.

Unlike SLR [AnH99], our method maps a set of flows on to a tag and uses the tag to find the best path. This eliminates the cost of mapping a flow onto a path at each content router and also allows handling large volumes of data by using the content equivalence classes. The VCN updates the content routers about the current status of the attached hosting sites and of the network. This helps in routing using the most current metrics. The network administrator can also explicitly shape a route. We expect that it will be easier to perform traffic engineering techniques (including RSVP) as compared to the SLR method.

In our content routing scheme, we try to incorporate the idea of CNP [ChL99], whereas the most popular document is accessed more easily/faster than the other less popular documents. Our content based routing model provides a scalable and generalized method for routing on geographically distributed wide-area networks. We implement a storage mechanism to handle the changing popularity of the contents on the time-of-day basis. CNP allows expanding its server/caching spans by dissemination of content information. We use a similar idea by creating fixed length tags and disseminating them across the

VCN. In this thesis, we present some efficient methods for tag dissemination and content discovery and access methods.

8.3 Future Research Directions

Our analysis and simulation based experiments reveal that our proposed content routing architecture has the capability of delivering content on a wide area distributed network with quality of service guarantees and at a finer granularity than the existing content routing architectures. However, several issues need to be addressed further before a system based on PICS can be deployed. Some of the issues include: (a) improved tag dissemination protocols, (b) efficient content characterization and classification schemes, and (c) mechanisms for interpreting and exploiting the staleness in status information of remote servers. Optimized techniques for improved traffic engineering and security functions along with a more flexible content description scheme will lead to a more efficient performance. We are currently working on a PICS model for improved tag distribution techniques to further reduce the network overhead. We also intend to extend the forwarding algorithm enabling the content-based routers to make decisions using a greedy heuristic to maximize the utilization of the content switching paths. For this we intend to examine the performance of our proposed routing architecture by using different routing protocols.

In our proposed scheme, we have described a technique for creating, storing and handling the content-to-tag bindings. However, as the number of distinct content increases,

we foresee a scalability problem in managing and disseminating the content-to-tag information in the VCN. To handle this problem, we are working to find an optimized method for handling the tags within the VCN. We also want to include more information while creating the tags using the content characterization and classification schemes. For example, by including information obtained from cookies we will be able to provide more support applications that require secure and persistent connection. We are working to develop an formal and generalized design for our proposed content classification and characterization schemes. Furthermore, we also have plans for developing PICS to create a complete content-aware multi-vendor content delivery network that will allow content providers and users to benefit from the cumulative service benefits of multiple service providers.

Bibliography

[AdS99] W. Adjie-Winoto, E. Schwartz, H. Balakrishnan, and J. Lilley, "The Design and Implementation of an Intentional Naming System," *17th ACM Symposium on Operating Systems Principles (SOSP'99)*, Dec. 1999, pp. 186–201.

[AhD96] J. S. Ahn and P. B. Danzig, "Packet Network Simulation: Speedup and Accuracy Versus Timing Granularity," *IEEE/ACM Transaction on Networking*, Vol. 4, Oct. 1996, pp. 743–757.

[AnH99] N. Anerousis and G. Hjalmtysson, "Service Level Routing on the Internet," *IEEE Globecom*, Vol. 1b, Dec. 1999, pp. 553–559.

[AnK98] A. Anastasiadi and S. Kapidakis, "A Computational Economy for Dynamic Load Balancing and Data Replication," *1st International Conference on Information and Computation Economies*, Oct. 1998, pp. 166–180.

[BaM98] R. Bagrodia, R. Meyer, M. Takai, Y. Chen, X. Zeng, J. Martin, B. Park, and H. Song, "Parsec: A Parallel Simulation Environment for Complex Systems," *IEEE Computer*, Vol. 31(10), Oct. 98, pp. 77–85.

[BlB98] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, "An Architecture for Differentiated Services," RFC 2475, Dec. 1998.

[Bla00] D. Black, "Differentiated Services and Tunnels," RFC 2983, Oct. 2000.

[Bro99] S. Brown, *Implementing Virtual Private Networks*, McGraw Hill, Berkeley, CA, 1999.

[BuG98] R. Buff, A. Goldberg, and I. Pevzner, "Rapid, Trace-Driven Simulations of the Performance of Web Caching Proxies," *Computer Science Department, New York University (Available at http://www.cs.nyu.edu/artg/)*, Sep. 1998.

[BuW01] M. Busari and C. Williamson, "On the Sensitivity of Web Proxy Cache Performance to Workload Characteristics," *IEEE INFOCOM*, Vol. 3, July 2001, pp. 1225–1234.

[ChL99] C. Chiang, M. Liu, and M. Muller, "Caching Neighborhood Protocol: A Foundation for Building Dynamic Caching Hierarchies with WWW Proxy Servers," *International Conference on Parallel Processing (ICPP'99)*, Sep. 1999.

[CoH01] E. Cohen and H. Kaplan, "Proactive Caching of DNS Records: Addressing a Performance Bottleneck," *Proceedings of the Symposium on Applications and the Internet*, Jan. 2001.

[DaC01] M. Day, B. Cain, G. Tomlinson, and P. Rzewski, "A Model for Content Internetworking (CDI)," Internet-Draft (work in progress): draft-day-cdnp-model-09.txt, November 2001.

[DaR00] B. Davie and Y. Rekhter, *MPLS: Technology and Applications*, Morgan Kaufmann Publishers, San Diego, CA, 2000.

[Doa96] M. B. Doar, "A Better Model for Generating Test Networks," *IEEE Globecom*, Nov. 96, pp. 86–93.

[FaL00] D. Farinacci, T. Li, S. Hanks, D. Meyer, and P. Traina, "Generic Routing Encapsulation (GRE)," RFC 2784, Mar. 2000.

[FrB96] N. Freed and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types," RFC 2046, Nov. 1996.

[GrC01] M. Gritter and D. R. Cheriton, "An Architecture for Content Routing Support in the Internet," *USENIX Symposium on Internet Technologies and Systems*, Mar. 2001.

[Her00] S. Herzog, "RSVP Extensions for Policy Control," RFC 2750, Jan. 2000.

[JoC01] K. L. Johnson, J. F. Carr, M. S. Day, and M. F. Kaashoek, "The Measured Performance of Content Distribution Networks," *Computer Communications*, Vol. 24(2), May 2001, pp. 202–206.

[JuS01] J. Jung, E. Sit, H. Balakrishnan, and R. Morris, "DNS Performance and the Effectiveness of Caching," *Proceedings of ACM SIGCOMM Internet Measurement Workshop*, Aug. 2001.

[KeA98] S. Kent and R. Atkinson, "Security Architecture for the Internet Protocol," RFC 1825, Nov. 1998.

[Kes97] G. Kessler and S. Shepard, "A Primer On Internet and TCP/IP Tools and Utilities," RFC 1251, Mar. 1997.

[KrW01] B. Krishnamurthy, C. Wills, and Y. Zhang, "On the Use and Performance of Content Distribution Networks," *Proceedings of ACM SIGCOMM Internet Measurement Workshop*, Aug. 2001.

[LaZ84] E. D. Lazowska, J. Zahorjan, G. S. Graham, and K. C. Sevcik, *Quantitative System Performance Computer System Analysis Using Queueing Network Models*, Prentice-Hall Inc., New Jersey, NJ, 1984.

[LiA01] C. Liu and P. Albitz, *DNS and BIND*, O'Reilly & Associates, Sebastopol, CA, 2001.

[LiF01] B. Liu, D. R. Figueiredo, Y. Guo, J. Kurose, and D. Towsley, "A Study of Networks Simulation Efficiency: Fluid Simulation vs. Packet-level Simulation," *IEEE INFOCOM*, Vol. 3, Apr. 2001, pp. 1244–1253.

[Mal94] G.Malkin, "RIP Version 2 Protocol Analysis," RFC 1721, Nov. 1994.

[MeL01] A. Medina, A. Lakhina, I. Matta, and J. Byers, "BRITE: Universal Topology Generation from a User's Perspective," *International Workshop on Modeling, Analysis and Simulation of Computer and Telecommunications Systems(MASCOTS'01)*, Aug. 2001.

[MiN00] B. S. Michel, K. Nikoloudakis, P. Reiher, and L. Zhang, "URL Forwarding and Compression in Adaptive Web Caching," *IEEE INFOCOM*, Vol. 2, Mar. 2000, pp. 670–678.

[Moc87] P. Mockapetris, "Domain Names - Concepts and Facilities," RFC 1034, Nov. 1987.

[Moy98] J.Moy, "OSPF Version 2," RFC 2328, Apr. 1998.

[NiB98] K. Nichols, S. Blake, F. Baker, and D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers," RFC 2474, Dec. 1998.

[PaA98] V. Pai, M. Aron, G. Banga, M. Svendsen, P. Druschel, W. Zwaenepoel, and E. Nahum, "Locality-Aware Request Distribution in Cluster-based Network Servers," *8th ACM Conference on Architectural Support for Programming Languages and Operating Systems*, Oct. 1998, pp. 205–216.

[Pos94] J. Postel, "Domain Name System Structure and Delegation," RFC 1591, Mar. 1994.

BIBLIOGRAPHY

[PtT98] T. H. Ptacek and T. N. Newsham, "Insertion, Evasion, and Denial of Service: Eluding Network Intrusion Detection," *Technical report, Secure Networks*, Jan. 1998.

[RaF01] S. Ratnasamy, P. Francis, M. Handley, and R. Karp, "A Scalable Content-Addressable Network," *ACM Conference on Applications, Technologies, Architectures and Protocols for Computer Communications*, Aug. 2001, pp. 161 – 172.

[RoD01] A. Rowstron and P. Druschel, "Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems," *18th IFIP/ACM International Conference on Distributed Systems Platform (Middleware'01)*, Nov. 2001, pp. 329–350.

[ShT01] A. Shaikh, R. Tewari, and M. Agrawa, "On the Effectiveness of DNS-based Server Selection," *IEEE INFOCOM*, Vol. 3, Apr. 2001, pp. 1801–1810.

[TaD00] W. Tang, F. Du, M. W. Mutka, L. M. Ni, and A. Esfahanian, "Supporting Global Replicated Services by a Routing-Metric-Aware DNS," *2nd International Workshop on Advanced Issues of E-Commerce and Web Based Information Systems*, June 2000, pp. 67–74.

[YaG97] A. Yan and W. Gong, "Fluid Simulation for High Speed Networks," *15th International Teletraffic Congress*, June 1997.

[ZhF98] L. Zhang and S. Floyd and V. Jacobson, "Adaptive Web Caching: Towards a New Global Caching Architecture," *3rd International Web Caching Conference*, June 1998.