Distributed Network Storage Service with Quality-of-Service Guarantees

John Chung-I Chuang
Carnegie Mellon University/
University of California, Berkeley

Marvin A. Sirbu
Carnegie Mellon University

Abstract

This paper envisions a distributed network storage service with Quality-of-Service (QoS) guarantees, and describes its architecture and key mechanisms. When fully realized, this service architecture would be able to support, in one integrated framework, network storage services ranging from best-effort caching to replication with performance guarantees. Content owners could, through the use of standardized protocols, reserve network storage resources to satisfy their application-specific performance requirements. They would be able to specify either the number and/or placement of the replicas, or higher-level performance goals based on access latency, bandwidth usage or data availability. The network storage provider would then optimally allocate storage resources to meet the service commitments, using leftover capacity for best-effort caching. Content consumers would then retrieve the nearest copy of the data object, be it from a replica, cache, or the original source, in a completely transparent manner. Furthermore, a distributed network storage infrastructure should be integrated with the existing transmission-based QoS framework so that applications can select the optimal combination of storage and transmission resources to satisfy their performance requirements.

This work identifies and discusses key research areas and problems that need to be tackled, including those in service specification, resource mapping, admission control, resource reservation and real-time storage management. The paper establishes a QoS framework upon which community discussion on this vision can proceed.

1. Introduction

The ideas of caching, replication, and memory hierarchy in general are well established in computer hardware, operating systems, distributed databases and distributed file systems design. The growth of Internet traffic, together with the strong references of locality observed in wide-area data access patterns, spurred the expansion of the memory hierarchy into the network itself. Network caching and replication involves the storing of multiple copies of data objects in distributed locations throughout the network. Accesses to data are satisfied by these nearby copies, saving the need to go all the way back to the original source. This results in four significant benefits:

- reduced access latency
- reduced bandwidth consumption
- server load balancing
- improved data availability/redundancy.

The traditional distinction between caching and replication is that of ex-post versus ex-ante data duplication. An initial data request is needed to trigger the caching of the data object, and subsequent requests for the same object are served from the cached copy until it is purged from the cache. A replicated copy of an object, on the other hand, is made in anticipation of its use at some future time. This anticipatory execution of replication can be based on a highly selective and speculative prefetching algorithm, or a complete duplication of the entire object-space (e.g., a mirror site).

An additional distinction between caching and replication derives from their differing economics. Caching is often implemented by a network service provider (e.g., ISP) purely as a means to reduce communications link costs. Any benefits (e.g., improved latency and availability) or costs (loss of content control, undercounting of advertisement impressions) to the content provider are not internalized in the ISP's decision. Replication, by contrast, is typically purchased by the publisher to achieve latency or availability goals, but the transport savings to the network service providers may not be internalized in the publisher's decision.

Tables 1 and 2 illustrate the cost-benefit tradeoffs of network caching and network replication from the perspectives of the network service provider and the content provider. In network caching (Table 1), the network service provider is the entity making the decision to set up and operate the cache, and therefore incurs the appropriable cost. It realizes appropriable benefits in the form of transport savings, since the average distance traveled by data packets is now reduced. On the other hand, the content provider is not the decision-making entity for cache installation and operation. Therefore, the costs and benefits it realizes are nonappropriable. The primary benefit to the content provider is improved performance in data delivery, including (i) reduced network latency, (ii) improved data availability, and (iii) a reduction in server load. However, the content provider loses control over the content stored in the network caches. First, they can no longer control the number and location of the cached copies. Second, the lifetime of the cached objects is now subject to the specific replacement policies instituted by the cache operator. Third, the content provider has no means of invalidating objects once they are in the cache, creating the possibility of stale data. Finally, and perhaps most important to some content providers, the hit-statistics collected at the server will be much lower than the actual hit-count of the objects. Lacking a consistent, reliable way to account for hits at the network caches, many content providers are actively engaged in "cache-busting" by tagging their objects "noncacheable." Indeed, legislative efforts are under way in various parts of the world to ban the use of caches on intellectual property rights infringement grounds.

Table 1. Cost-benefit tradeoffs of network caching from the perspectives of (i) network service provider and (ii) content provider.

	Network Service Provider	Content Provider
Appropriable Benefit	Transport savings	
Appropriable Cost	Cache setup/operation	
Non-appropriable Benefit		Better performance: · improved latency · improved availability · reduced server load
Non-appropriable Cost		Lose content control: · no access statistics · possible stale data · number/location/lifetime of object

In contrast, the decision to replicate is made by the content provider (Table 2). Today, the content provider may contract a Web-hosting company to operate one or more replicas (or mirrors), and incur the appropriable cost of replication. In return, it receives all the performance improvements as in the caching arrangement and avoids surrendering content control. From the network service provider's point of view, replication also results in transport savings, but does not incur any nonappropriable cost to the network. The major downside with network replication (more specifically Web-hosting) today is the high transaction costs associated with setting up a replica. This necessarily means that replicas are set up for longer durations (e.g., months rather than hours or days) and are therefore not as responsive to changes in traffic and/or demand.

Table 2. Cost-benefit tradeoffs of network replication from the perspectives of (i) network service provider and (ii) content provider.

	Network Service Provider	Content Provider
Appropriable Benefit		Retain content control; Better performance
Appropriable Cost		Cost of replication
Non-appropriable Benefit	Transport savings	
Non-appropriable Cost	---	

A properly constituted network storage market would internalize all of these costs and benefits and allow optimally efficient decision making by both publishers and service providers. Our goal in this paper is to show how these efficiency gains can be realized.

1.1 Caching and Replication: A QoS Perspective

This work suggests that new and important insights can be gained by looking at caching and replication from a Quality-of-Service (QoS) perspective. The QoS concept is not new; it comes from the transmission domain of data networking. What is new is the treatment of caching and replication as different QoS services within a unified network storage framework.

When applied to data transmission, QoS introduces the distinction between guaranteed service and best effort service. Best effort service is the default service used by most applications, and it does not offer any guarantees regarding packet delivery. Guaranteed service, on the other hand, offers performance guarantees based on latency, jitter and/or loss rates.

When the same QoS concept is applied to network storage, we recognize that caching can be considered to be a best-effort service, in contrast to replication which is a guaranteed service. Network caches perform best-effort service by storing a local copy of each object it sees (except those explicitly tagged uncacheable). Since caches have finite storage capacity, they have to evict old objects to make room for new ones. The fact that objects may be purged at any time means caches cannot provide any guarantees of data persistence. The possibility of a cache miss introduces uncertainty in the access latency of an object, similar to the introduction of jitter in data transmission.

Replication, on the other hand, represents a service commitment to keep a persistent copy of the object. There can be no misses at the replica (transmission analogy: no packet drops) in this guaranteed service. In order to provide guarantees, replication requires some form of resource reservation. Today, replication setups are mostly ad hoc and require manual intervention for want of a standardized reservation protocol. It is therefore extremely expensive, if not impossible, for these static replicas to respond to changing traffic patterns and network conditions.

The need for service guarantees in data transmission is driven by real-time network applications that cannot tolerate variations in packet delay. We believe that the demand for network storage services with guarantees will similarly come from applications that cannot tolerate the performance variations inherent in all caching schemes. These applications may be mission-critical, have stringent performance and/or availability requirements, or place high value in consistent data access latency, thereby requiring data objects to be kept in persistent storage even if they do not exhibit reference locality, or are rarely accessed at all. No amount of intelligent or adaptive caching, or overprovisioning (short of infinite cache size) can address the needs of these applications.

1.2 Distributed Network Storage Infrastructure with QoS Guarantees

This paper calls for the building of a distributed network storage infrastructure with QoS guarantees[1]. This service model will support, in one integrated framework, network storage services ranging from best-effort caching to replication with performance guarantees. Content owners can, through the use of standardized protocols, reserve network storage resources to satisfy their application-specific performance requirements. They can specify either the number and/or placement of the replicas, or higher-level performance goals based on access latency, bandwidth usage or data availability. The network storage provider will optimally allocate storage resources to meet the service commitments, using leftover capacity for best-effort caching. Content consumers retrieve the nearest copy of the data object, be it from a replica, cache, or the original source, in a completely transparent manner.

The network storage resource thus reserved will be available for housing objects ranging from web pages, audio and video files, to databases, applets and executables. These storage nodes can also support scripts and processes that generate dynamic objects and maintain logs of access statistics.

Furthermore, this distributed network storage service can be integrated with the existing transmission-based QoS framework so that applications can select the optimal combination of storage and transmission resources to satisfy their performance requirements. While the focus of this paper is on services based on network storage resources, we have explicitly adopted and adapted design philosophies and terminology from the transmission domain so as to facilitate a seamless integration of the two to create an economic marketplace for optimally trading off storage and transmission resources.

Figure 1 illustrates the process of turning performance requirements into performance realization via intelligent allocation of network and storage resources. Consider a publisher with application-specific performance requirements and a-priori information about the probable pattern of information access by consumers (across objects, space, and time), the publisher should be able to use some standardized semantics to express its formal QoS requirements. These requirements are then conveyed to the

[1] This paper is based on work reported in [19].

network storage service provider using some well-established resource reservation protocol.

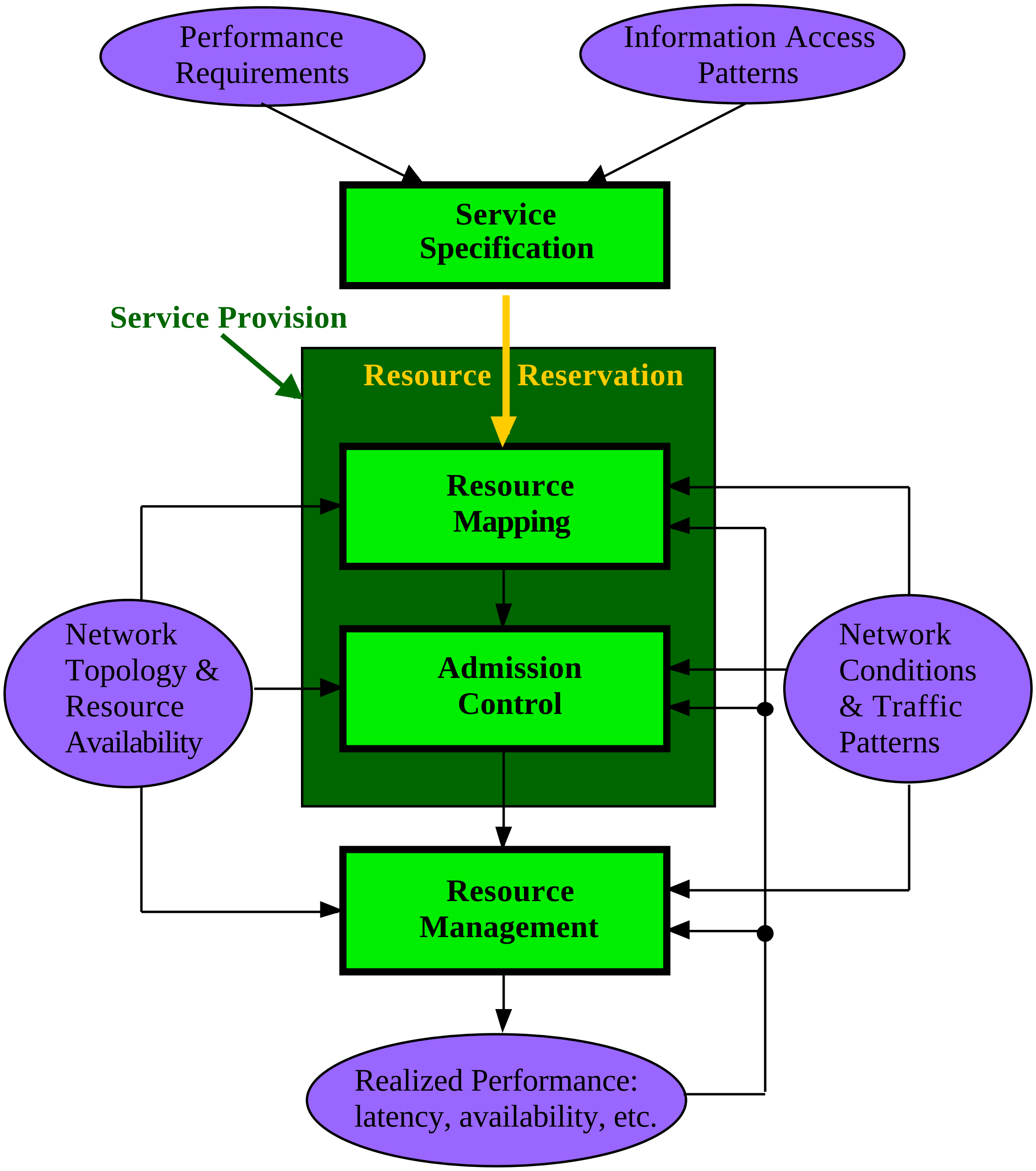

Figure 1. From performance requirements to performance realization: the process flow of establishing a network storage service with QoS guarantees.

The provider, using information available to it regarding network topology, storage resources, and other current and projected resource demands, maps the QoS requirements into an optimal set of specific resource requirements for both distributed storage and transmission capacity to meet the QoS requirements at minimum resource cost. This cost may be either more (due to the cost of storage replicas) or less (due to network savings) than the cost of disseminating content from a single server attached to the network at a single point; performance will always be better.

Having calculated this optimal resource mapping, the service provider attempts to reserve the specific link and storage resources as determined by the mapping. Depending upon the extent of previous resource reservations, individual transmission and storage facilities may admit or deny the reservation request. If the requests are denied, then an alternative resource mapping must be computed, and the process repeated.

Individual storage nodes execute real-time resource management policies (e.g., local cache replacement, replica update policies, etc.) to maximize local resource utilization while meeting all service commitments.

The result is a performance realization, measured using the same metrics as those used to express the QoS requirements in the first place.

Finally, like all other multi-attribute services, appropriate pricing models must be put in place, so that prices can act as market signals to optimize the use of the network and storage resources at the demanded level of performance.

Given this framework, we can identify the key components of the distributed network storage service:

· service specification	(Section 3)
· service provision	(Section 4)
- resource reservation	(Section 4.1)
- resource mapping	(Section 4.2)
- admission control	(Section 4.3)
· real-time resource management	(Section 5)
· metadata management	(Section 6)

Each of the components and their associated research problems will be described in the sections indicated above, following a review of relevant literature in Section 2. There are other mechanisms necessary for the distributed network storage infrastructure, including (i) resource discovery (with issues like naming, name resolution and distance estimation), (ii) security, (iii) accounting, billing and payment. While important in their own right, these mechanisms do not pertain to the QoS aspect of the infrastructure, and have been the subjects of substantial research efforts elsewhere. Therefore they are outside the scope of this paper. The details of QoS pricing and industrial organization of the infrastructure are also reserved for future work.

2. Related Work

This work benefits from the cross-fertilization of two fields, namely (i) network caching and replication, and (ii) transmission-based QoS. While active research continues apace in both fields, this is the first proposal that introduces the notion of QoS-based services to the network storage domain. Caching alone is a “best effort” service that cannot provide service guarantees to a publisher.

The ideas of caching, replication, and memory hierarchy in general are well established in computer hardware, operating systems, distributed databases and distributed file systems design. The growth of Internet traffic (file transfer protocol [FTP] and hypertext transfer protocol [HTTP] in particular) spurred the expansion of the memory hierarchy into the network itself. Caching and replication started out at the edge of the network. Caching proxies [1, 2] are installed at campus gateways and at the ISP’s metropolitan points-of-presence (POPs); mirror sites are installed, with manual intervention, as replicated servers of popular FTP/HTTP sites.

2.1 Network Caching

Before long, caching progressed into the wide-area network (WAN) itself. Motivated by strong references of locality observed in wide-area data access patterns [3-5], network caches are organized into hierarchies [6]. Despite a dynamic hierarchical caching proposal [7], the dominant network-caching infrastructure is still a manually configured hierarchy of object caches. The static hierarchy is limited to no more than three levels for latency considerations. Some proposals call for object sharing among neighboring caches via inter-cache communication protocols [8-11]; others call for network caches that can handle dynamic objects [12, 13]. There is currently a flurry of adaptive, self-organizing caching proposals that promises intelligence, scalability and adaptability for network caching [14-18]. Under the rubric of “active networking”, DARPA has recently funded a project looking at the adaptation of protocols developed for cache management in network attached storage devices to the larger problem of unreliable WANs [19].

There has been a proliferation of novel replacement policies for network caches. These policies are all variants of the Least Recently Used (LRU) or Least Frequently Used (LFU) policies. But in addition to object popularity, these policies also incorporate object size, distance, latency and/or cost factors into the decision-making process [20]. On the other hand, there are applications whose objects are managed independent of how frequently or recently they were accessed. For example, objects that have stringent performance requirements or are mission-critical need to be kept in persistent storage even if they do not exhibit reference locality, or are rarely accessed. In these instances, the object owners would seek to secure network storage resources with QoS guarantees not available with network caching.

2.2 Network Replication

There have been many different proposals for network replication, though the only ubiquitous scheme is also one of the earliest: NNTP [21]. Network News Transfer Protocol involves massive replication of news-articles to NNTP servers (on the order of 10,000's) throughout the network. However, NNTP only offers weak consistency, providing no guarantees regarding on-time replication of articles. Subsequent work based on this massive replication concept uses either multicast [22] or the hierarchical organization of servers [23, 24]. The Internet2 initiative is proposing a distributed storage infrastructure which allows massive replication to a system of replicated servers, though the content publisher would have no control over the placement of the objects nor receive any performance guarantees [25]. Wolfson et al. examine adaptive replication algorithms for selecting web or database replica sites [26], but does not consider the problem of meeting specific QoS criteria.

Other proposals for network replication tend to focus on the ex-ante vs. ex-post distinction of data duplication [27, 28]. Therefore, they should be more accurately characterized as proactive or push caching [29, 30], selective pre-fetching [31, 32], or demand-driven replication [33]. The prior research most closely aligned with our proposal is that of [34, 35]. However, this work focuses on relatively stable data, and does not examine algorithms for optimal placement, alternatives along the spectrum from guaranteed service (mirrors) to best effort (caches), nor tradeoffs between reserving storage and reserving transmission capacity. Markatos and Chronaki argue for a hierarchical combination of selective replication (pre-fetching) and caching [36].

Today, mirroring and contracting to web-hosting services remain the only viable replication options that provide some form of persistence guarantee to a publisher. Both involve high degrees of customization and human intervention, and are therefore limited to static, long-term arrangements,[2] involving entire sites as opposed to individual data objects. Responding to changing traffic patterns and network conditions is extremely costly, if not impossible, in these cases.[3]

[2] Official sites for the Olympic Games, World Cup are notable exceptions.

[3] There is a recent proposal to bring differential service to web servers and content-hosting servers [37]. This scheme calls for the preferential scheduling and processing of requests, but does not offer any guarantees with regard to object persistence. In this it is similar to notions of differential service for network transmission.

2.3 Transmission-based QoS

The need for network support for multiple service levels has been long recognized [38-40]. Real-time network applications require some form of performance guarantees that are not available from a single-class best-effort infrastructure. Therefore, the concept of QoS was introduced at the IETF and ATM Forum organizations and became embodied in standards such as the *intserv* framework [41] and the Traffic Management Specification [42]. These standards specify the different service classes and the service guarantees available to network applications. The realization of these schemes requires advances in traffic specification [39, 43], resource reservation [44], resource mapping [45-47] and admission control [48, 49], scheduling algorithms[4] [39, 51-54] and queue management [55], along with various other control and management mechanisms such as traffic policing. Pricing design for multi-service networks has also witnessed a flurry of research activity [56-65]. While the transmission QoS literature provides a useful starting point for identifying the mechanisms needed for a distributed network storage infrastructure, we believe that there are fundamental differences between the two infrastructures that require more than simple adaptation of designs and architectures.[5]

Having identified the relevant literature in network caching, replication and transmission-based QoS, we are now ready to describe the key components of the distributed network storage infrastructure.

3. Service Specification

The first step towards creating a useful distributed network storage infrastructure is to identify service classes that may be of value to applications. In the previous sections we have simply identified caching and replication as the two basic service classes. In reality, different applications have diverse needs and performance goals and will therefore demand different flavors of network storage services. A service specification standard or API (application programming interface) will allow the content owners and the network storage providers to communicate, using unambiguous metrics, the requirements and expectations of a service commitment.

There are two chief elements to a service specification: *traffic profile* and *performance requirements*. In data transmission, the traffic profile of the source is

[4] Zhang provides a comprehensive survey of packet scheduling disciplines [50].

[5] For example, transmission buffers generally follow a FIFO discipline (or some variant of FIFO), but network storage is usually random access. Therefore, the nature and cost of congestion is not the same. Also, we expect the nature and degree of traffic burstiness to be different between data transmission and network storage demand.

usually expressed as some combination of peak and average rates, maximum burst length, token bucket filter rate, etc. [39, 43]. Performance requirements, on the other hand, are usually specified in delay bounds, acceptable loss rates, etc. When a service contract is established, the network is responsible for meeting the performance requirements, so long as the source transmits data within the prescribed traffic profile.

The specification of a network storage service also consists of a traffic profile and performance requirements. The traffic profile declares the amount of storage capacity to be reserved, the time and duration of the reservation, and the distribution of data accesses, if known. The performance requirements can be expressed along one or more of the following (sometimes overlapping) dimensions:

- data access latency (mini-sum, mini-max)
- data access jitter
- acceptable miss rate (including 0%)
- data availability/redundancy
- coverage area
- bandwidth savings
- cost

The distributed network storage infrastructure has to be able to accommodate new service classes and new performance metrics as the market demands them. We provide some example services here for illustrative purposes (Table 3).

Table 3. Some examples of network storage services.[6]

	Service	Description (traffic profile, performance requirements)
#1	Deterministic	1GB storage capacity for 1 hour, 100ms maximum latency
#2	Average	1GB storage capacity for 1 hour, 50ms average latency
#3	Combination	1GB storage capacity for 1 hour, 50ms average latency, 100ms worst case latency
#4	Stochastic	1GB storage capacity for 1 hour, Probability[latency > 100ms] ≤ ε
#5	Geographic	1GB storage capacity for 1 hour, 100ms latency bound for all receivers in specific domain or region, or to specific set of receivers
#6	Budget-constrained	1GB storage capacity for 1 hour, minimizing worst-case latency, subject to budget constraint of no more than *K* replicas
#7	Placement-oriented	1GB storage capacity for 1 hour, at *N* specific nodes
#8	Advance reservation	1GB storage capacity from 2330hr, December 31 1999 to 0029hr, January 1 2000, 100ms latency bound

[6] While most of these example services are specified with latency requirements in milliseconds, they can also be specified in terms of network hops. Alternatively, the performance requirements may not be latency-based at all.

These services are just a small sample of the many possible services that may be offered over the distributed network storage infrastructure. Clearly, the more types of services to support, the richer the specification semantics need to be. The challenge, as always, is in achieving the right balance between simplicity and flexibility. While these example services offer a glimpse into the many dimensions along which services may be classified, we choose to highlight one particular dimension in the following sub-section.

3.1 Deterministic vs. Statistical Guarantees

Services can be differentiated by the "firmness" of their guarantees. The QoS work in the data transmission arena provides ample illustrations [40, 66]. The IETF (Internet Engineering Task Force), for example, has specified three classes of services as part of their integrated-services framework: guaranteed service (GS), controlled load service (CLS) and best effort service (BES) [41, 67-69]. Similarly, the ATM Forum has specified four classes: constant bit rate (CBR), variable bit rate (VBR), available bit rate (ABR) and unspecified bit rate (UBR) [42]. These service classes can be characterized as providing one of the following performance guarantees: deterministic, statistical, or no guarantee.

Applying this to our example services, we see that services #1, #5 and #8 provide deterministic guarantees on access latency. All data accesses are guaranteed to experience no more than the stipulated 100ms delay. Services #2 and #4, on the other hand, offer statistical guarantees. Service #2 makes latency guarantees only for data accesses in the aggregate, but not for individual data accesses. For service #4, up to ε of data accesses may fall outside the latency bound without violation of the commitment. Service #3 offers a combination of deterministic and statistical guarantees. Finally, the best effort service of network caching corresponds to the base case of offering no guarantees.

It is important to recognize that services #2-4 do not necessarily represent the full range of services with statistical guarantees. The exact specification of statistical guarantee services may be dependent on the stochastic nature or the source of burstiness of the traffic load in question.

There are two sources of burstiness in terms of demand for network storage capacity. First, it is conceivable that some content owners may experience fluctuations in the size of their corpus. News publishers, for example, may have a relatively stable corpus size for ordinary news days but an explosion of additional news articles on days with extraordinary world events or stockmarket activity. These publishers may wish to characterize their traffic load with average and peak capacity numbers. Second, data access patterns may be bursty with respect to the objects requested, the geographic locations of the consumers, etc. These patterns may or may not be amenable to characterization using some demand distribution function (across objects, space and time).

To the extent that these stochastic behaviors or burstiness can be accurately characterized and made available to the network, appropriate statistical multiplexing techniques can be applied to improve storage utilization. On the other hand, for those applications with no burstiness in storage demand, they cannot hope to realize any statistical multiplexing gains, and are better off with deterministic-guarantee services.

Finally, in addition to these guaranteed services, there is also effort at the IETF to introduce differential or differentiated service to the Internet [70]. This service provides no performance guarantees, but offers some notion of a premium service where packets are given preferential treatment over best effort packets. If we wish to apply this differential service concept to network storage, then some form of cache replacement policy that takes priority into account will be required. Alternatively, a "premium" data object may be tagged and initiated with a negative number in its age field when it is first cached, so as to lengthen its cache residency.

4. Service Provision

Having identified some possible network storage service classes, we turn to the mechanisms for providing these services. As in transmission-based QoS provision, there are three main components of network storage service provision: *resource reservation*, *resource mapping* and *admission control* [71].

4.1 Resource Reservation Protocol

A resource reservation protocol allows the service requester and the service provider to communicate and negotiate the reservation of transmission and storage resources according to the service specifications. The protocol must be able to support the various types of services to be offered, including both performance-oriented and placement-oriented services. It would also be desirable for the protocol to include provisions for returning to the requester delivery logs and other indications that service level agreements are being met.

The resource reservation protocol for network transmission services, RSVP [44], serves as a useful starting point for discussion. One possibility might be to extend the current RSVP protocol so that it can support reservation requests for storage resources as well as transmission resources. However, we foresee some difficulties with this approach. First of all, the concepts of the routing path and end-to-end reservation do not apply to storage. Secondly, in the case of replication, the “receivers” or the content consumers may not be known at reservation time.[7] Whereas both sender and receiver(s) are involved in transmission-based resource reservation, only the content owner is involved in the storage-based case. This goes against the fundamental design philosophy of receiver-initiation in RSVP. The specification of the resource reservation protocol is outside the scope of this work, and should be postponed until the overall network storage service provisioning architecture has been defined.

4.2 Resource Mapping

Resource mapping is the translation of high-level service specifications into low-level resource requirements. To be able to make optimal resource allocation decisions, the resource mapping entity has to be constantly updated with the status and availability of a heterogeneous set of resources at a global level. It may need to maintain a knowledge-database with information such as network topology, storage capacity, link capacity, link delay, network condition, and predictions of future traffic patterns (possibly based on measurements of current traffic patterns).

For a storage-based QoS service provider, such as a web-hosting service that does not control transmission resources, the resource mapper will map QoS requirements into storage resources only. It does so by assuming that only best effort transmission service is available, and this service is characterized by some delay distribution on each link. On the other hand, for a unified transmission-storage QoS infrastructure, the resource mapper may map QoS requirements into a combination of storage and transmission resources. These transmission resources may range from dedicated transmission capacity (e.g., leased lines), QoS services based on intserv, diffserv, to IP “overnet” services that provide single-hop connectivity between specified end points.[8]

For storage services with deterministic guarantees, resource mapping has to be performed based upon the peak or worst-case resource requirements. The demand distribution of data accesses is irrelevant; the resource mapper simply identifies the set of

[7] Conversely, the installation and use of local caches by the end user (or organization) may be considered a form of receiver-based storage resource reservation, but it is usually performed without explicit involvement of the content owners (senders). Finally, network caching may be performed by the network provider in complete transparency to both senders and receivers, and without the need for resource reservation.

[8] Digital Island, an Internet Service Provider, offers single hop connectivity between major network access points throughout the world by selective provisioning of network capacity. This service is used by online publishers, for example, to achieve performance targets for their information dissemination applications [72].

network nodes at which storage capacity needs to be reserved in order to meet latency and/or other performance requirements for any object requested by any consumer.

For storage services with statistical guarantees, the resource mapper can take into consideration the probability distribution of data accesses when determining the optimal set of network nodes. To the extent that demand for network storage can be characterized as Markovian, it may be possible to apply the effective bandwidth or equivalent capacity concepts from the data transmission domain [46, 47].

In [73] we show that the resource mapping problem can be formally characterized and solved as a facilities-location problem, as in the location theory literature [74]. In particular, the mapping of a service with deterministic guarantee can be described as a weighted k-center problem [75, 76], while that of a service with statistical guarantee can be described as a weighted k-median problem. In either problem instance, the objective is to find the optimal number and placement of replicas such that the delay or distance bound is met.

By applying the formal model to the early ARPANET topology as an example network, we are able to demonstrate the operation of the resource-mapping process, including the mapping into an optimal combination of storage and transmission resources.

Qualitatively, the results agree with one's intuition:

- There is an inverse relationship between the delay bound and the number of replicas required to achieve it;
- Optimal replica placement changes with the number required: the optimal locations for two replicas may not overlap the optimal locations for three replicas;
- Statistical delay bounds can be met with fewer replicas by exploiting locality of reference to place fewer replicas near concentrations of demand; and
- Much of the statistical benefit of full replication can be achieved with smaller, partial replicas which contain the most heavily accessed objects.

The primary contribution of our formal model is to quantify precisely these intuitions; for example, we show how to calculate the magnitude of the performance degradation when replicas are constrained to be located at predetermined server sites as opposed to the optimal locations.

4.3 Admission Control

Because network transmission and storage capacities are finite, not all service requests can be accepted without adversely degrading the performance of the network.

Therefore, admission control is needed to reject those requests whose service contracts could not be fulfilled by the resources available at the time.

Admission control occurs in two stages. First, individual resource nodes (network switches or storage nodes) make local decisions as to whether a service request can be accommodated given the current availability of local resources. If all local decisions are positive, then a global check on aggregate requirements (e.g., aggregate delay bound) is performed (if necessary) before the final accept/reject decision is made.

In the case of transmission, admission control occurs along the routing path between sender and receiver (or receivers in the case of multicast). Switching nodes make local conditional acceptances and forward the request downstream, or send a reject message back to the sender. If a conditional acceptance is made, the switching node is obliged to set aside the requested capacity until the aggregate admission control decision is made, at which point the capacity is either fully committed or returned to the available pool. Therefore, the local admission control decisions have to occur sequentially on a hop-by-hop basis, and are finally followed by the aggregate decision.

In the case of storage, there is no notion of a path within a service request, and so all of the local admission control decisions can occur independently and in parallel. Furthermore, there is no need for an aggregate admission control decision, since there are no end-to-end requirements to be met. Therefore, all that is needed is a central entity to transmit admission control queries to and collect responses from the storage nodes. This role may be played by the resource mapper, or in the case of placement-oriented services, by the service requester itself.

There is clearly a tightly-coupled relationship between admission control and resource mapping. Therefore, it is important to recognize and leverage the possible synergy that may exist between the two entities. When resource utilization level is high, and the likelihood of a service request being rejected by the individual resource nodes is high, the resource mapping and admission control process may be iterated several times before a success is finally encountered. In this situation, it may be appropriate for the resource mapping and admission control functions to switch to a “greedy” algorithm or a quorum-based algorithm.

Both approaches reduce the number of possible iterations by sending admission control queries to more than enough nodes at the first attempt. In the “greedy” algorithm, the resource mapper will provide multiple sets of nodes that can satisfy a particular service request. The sets may or may not have common elements. The admission controller will send queries to the union of the sets, and declares the request admitted as soon as it receives positive responses from all the nodes of any given set. In the quorum-based algorithm, the resource mapper will provide a set of candidate nodes to which queries are sent. The service request will be declared admitted as soon as a quorum number of nodes returns a positive response.

5. Real-Time Resource Management

After the establishment of network storage services, the service provider has to perform real-time resource management in order to meet and enforce all service commitments.

In network transmission, resource management crudely means deciding which packets to transmit next (scheduling management) and which packets to drop (buffer management). The simplest queue discipline is FIFO (first-in first-out), which results in best effort transmission. To accomplish QoS guarantees, a combination of packet scheduling such as fair-weighted queuing [52] and traffic shaping at the edge of the network (e.g., token bucket with leaky bucket rate control) is necessary [77]. This paper will not deal with resource management in the data transmission context.

In network storage, resource management means deciding which data objects to keep in memory, which objects to purge. The most common replacement policy is LRU (least recently used) and it results in the implementation of best effort caching. To support QoS in network storage, we need to support the coexistence of data objects from both best-effort caching and guaranteed-service replication. Replicated objects have to be kept in memory for the entire duration of their service contract, while cached objects are aged and purged according to some object replacement policy. In addition to the variety of network cache replacement heuristics being proposed [20, 78, 79], cache replacement strategies can also include directives from the publisher (HTTP 1.1's *nocache* pragma), and ad hoc rules for identifying dynamic pages [80]. The techniques for marking and keeping replicated objects in memory might be adapted from virtual memory management (e.g., page locking) or distributed file system design (e.g., hoarding [81]). Finally, cache consistency mechanisms and replication update policies have to be put in place, and techniques for accomplishing these are readily available from distributed databases and file systems design.

5.1 Local Storage Management

Several important research questions have to be addressed with regard to local storage management. First, is there an optimal mix between replicated and cached objects in a network storage node? If so, what is the optimal mix? Alternatively, should a minimum fraction of storage be dedicated to caching? Intuitively, it makes sense not to commit all resources to replication, even though replication is expected to generate higher revenue than caching. A healthy supply of caching capacity will better deal with the burstiness in traffic and minimize the likelihood of thrashing.

5.2 Traffic Policing

Another local storage management issue is traffic policing. What happens when the content owner sends content in excess of the reserved amount? The storage manager exercises jurisdiction over this “non-conformant” traffic, and decides whether these objects should be discarded immediately, put into cache space (if available), or replace some existing objects in replication memory. Alternatively, the content owner may be sending an updated version of an object, in which case the stale object has to be identified and replaced.

The concept of *committed information rate* (CIR) from frame relay may be applied here. In data transmission, performance guarantees are provided for traffic transmitted at up to the committed information rate, while traffic in excess of the CIR are delivered as best-effort traffic. This guarantees each sender a minimum share of a link resource, while allowing them to send additional traffic when other senders are idle. An analogous concept of a *committed storage rate* (CSR) may be developed, such that a publisher is guaranteed a minimum fraction of a multi-publisher storage facility, and can store additional objects if free space is available. An alternate service might guarantee a minimum object lifetime before cache swap-out. The feasibility of these alternatives will have to be verified through modeling and simulation using cache trace data.

5.3 Hierarchical Resource Sharing

Hierarchical resource sharing or dynamic storage allocation also finds its analogy in link-sharing in the network transmission context [54, 82]. A content owner may have different classes of objects in its corpus, and wishes to assign different QoS levels for the different classes. The owner can make separate storage reservations, each with different performance requirements, for the different object classes. Alternatively, it can make a single storage reservation that allows real-time control over the allocation of reserved storage resources to different classes of data objects.

Consider the example of a popular news web-site (Figure 2). The size of the entire corpus is 2.5GB, and the publisher classifies the objects into one of three groups. The first group comprises of objects deemed critical by the publisher, such as the homepage and its navigational bars, the headline news articles, and the advertising banners. While its current size is 250MB, the publisher expects the size to fluctuate, but not to exceed 500MB. The bulk of the news content (2GB) makes up the second group.

Finally, 250MB of corporate information (e.g., press releases, job openings, mugshots of CEO and VP's) constitute the third group.

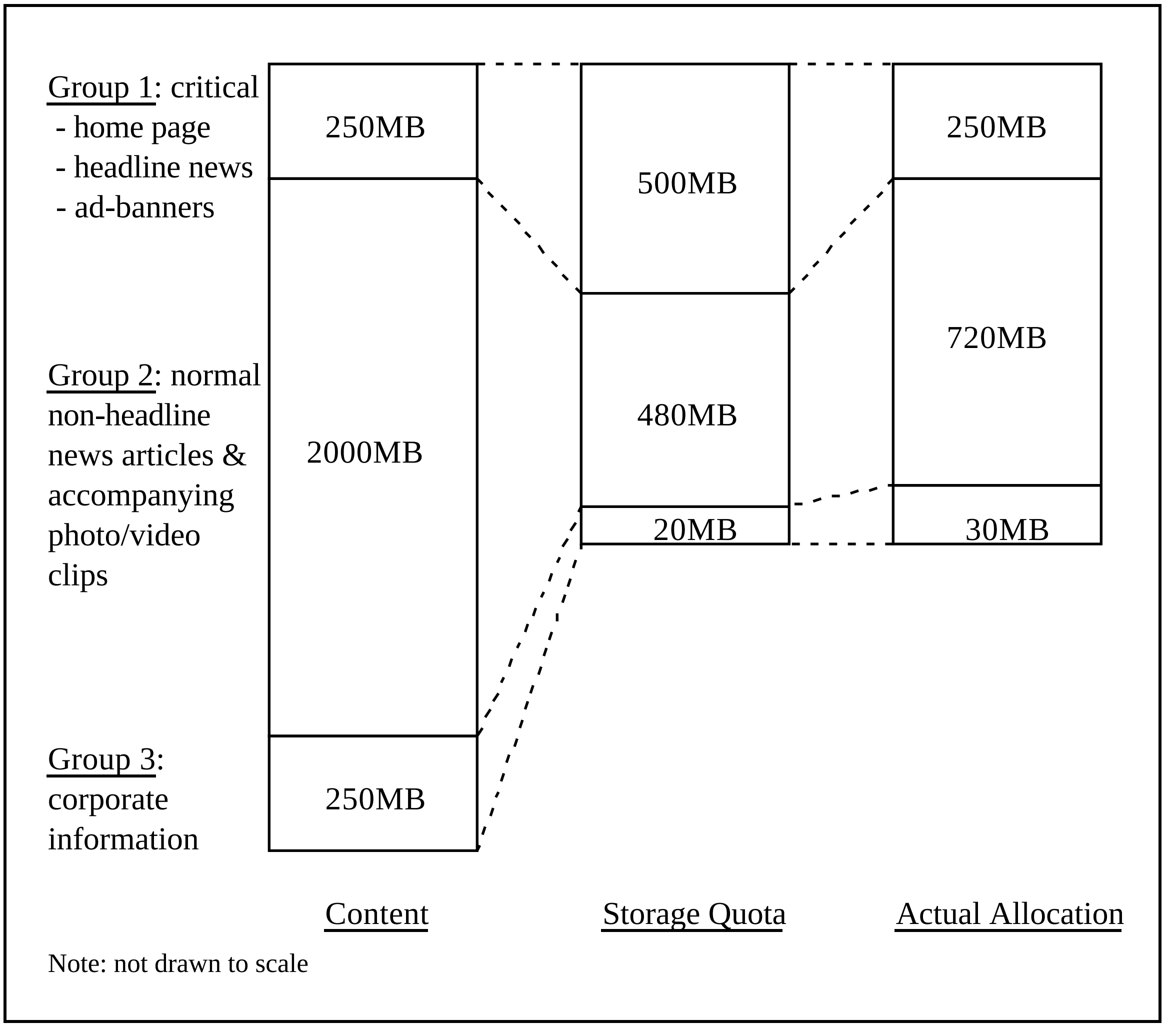

Figure 2. Hierarchical resource sharing example.

The publisher reserves 1GB of storage capacity and specifies the proportion to which storage will be allocated among the three groups. The publisher wants 100% of the group 1 objects to be in memory, even if the size of the group grows to 500MB. Therefore, group 1 is allotted 500MB or 50% of the storage quota. Groups 2 and 3 are then assigned 48% and 2% of the quota respectively.

Since there are currently only 250MB of group 1 objects, all of these objects are guaranteed to be in memory. The extra 250MB of group 1's quota will be proportionately shared (at a ratio of 24:1) between groups 2 and 3. Therefore, group 2 gets 480 + 240 = 720MB of storage and group 3 gets 20 + 10 = 30MB of storage. Should additional objects be added to group 1, storage capacity will be reclaimed from groups 2 and 3. This ensures that group 1 objects are always in memory, up to 500MB. Without this resource sharing scheme, the publisher would have to reserve and dedicate 500MB of storage capacity to group 1 objects, even when there are less than 500MB of objects most of the time.

Using this resource sharing scheme, the publisher can also control the degree of statistical multiplexing to take advantage of reference localities in data access patterns. In the same example, the publisher is able to achieve 100% coverage of group 1 objects (no statistical multiplexing), 36% coverage for group 2 objects, and 12% coverage for group 3 objects. The publisher can increase or decrease the storage quota for groups 2 and 3 to control the respective hit rates.

From this example, it is clear that hierarchical resource sharing is attractive because it gracefully absorbs the "burstiness" in object-class-sizes and facilitates user-controlled statistical multiplexing.

5.4 Global Storage Management

While the previous subsections deal with management issues local to the storage nodes, there are also global storage management issues that require study. In the normal operation of the distributed network storage infrastructure, there may be situations that require movement of data objects between storage nodes even after resource mapping and reservation. For example, changes in network status (e.g., network congestion, down nodes or links) may necessitate the movement of objects to maintain the existing service commitments. Alternatively, there may arise opportunities (e.g., termination of existing commitments, addition of new capacity) where data movement can lead to improved resource utility or load balancing. The scheduling of data migration, replication and de-replication constitutes the scope of global storage management [83].

6. Metadata Management

Publishers contracting with a network storage service are interested in more than simply timely delivery of content. They also need detailed information about the services actually provided: page impressions delivered, access patterns, traffic distribution, and so on. These data may be essential to recoup advertising revenues or control access to pay-per-view data. Thus, over and above protocols for managing the provisioning of network storage service, there must be protocols for reporting back to publishers on actual service usage.

7. Conclusion

In this paper we describe both a distributed network storage service with quality-of-service guarantees and its technical and economic mechanisms. When fully realized, this service model will support, in one integrated framework, network storage services ranging from best-effort caching to replication with performance guarantees, and optimal tradeoffs between storage and transport resources. Content owners can, through the use of standardized protocols, reserve network storage resources to satisfy their application-specific performance requirements. They can specify either the number and/or placement of the replicas, or higher-level performance goals based on access latency, bandwidth usage, or data availability. The network storage provider will optimally allocate storage resources to meet the service commitments, using leftover capacity for best-effort caching. Content consumers retrieve the nearest copy of the data object, be it from a replica, cache, or the original source, in a completely transparent manner. The price of this service will reflect an optimal balancing of storage and transport resources to achieve the publisher's content distribution objectives.

This paper establishes a QoS framework upon which community discussion on this vision can proceed. It also identifies key research areas and problems that need to be tackled, including those in service specification, resource mapping, admission control, resource reservation, storage management, location transparency, accounting, pricing and industrial organization.

The distributed network storage infrastructure represents a completely new economy with its unique set of cost structure, market agents, industrial organization and economic rules. Therefore its architects and designers have to be cognizant of the economic implications of different technical design choices, and consistently select the alternatives that promote competition, efficiency and equitability.

References

[1] A. Luotenen and K. Altis, “World-wide web proxies,” presented at 1st International Conference on the WWW, 1994.

[2] M. Abrams, C. R. Standridge, G. Abdulla, S. Williams, and E. A. Fox, “Caching proxies: limitations and potentials,” presented at 4th International World Wide Web Conference, Boston MA, 1995.

[3] P. B. Danzig, R. S. Hall, and M. F. Schwartz, “A case for caching file objects inside internetworks,” presented at ACM SIGCOMM, 1993.

[4] A. Almeida, A. Bestavros, M. Crovella, and A. de Oliveira, “Characterizing reference locality in the WWW,” presented at IEEE Conference on Parallel and Distributed Information Systems, Miami Beach FL, 1996.

[5] B. A. Huberman, P. L. T. Pirolli, J. E. Pitkow, and R. M. Lukose, “Strong regularities in world wide web surfing,” in *Science*, 1998.

[6] A. Chankhunthod, P. B. Danzig, C. Neerdaels, M. F. Schwartz, and K. Worrell, “A hierarchical internet object cache,” University of Southern California 95-611, March 1995.

[7] M. A. Blaze and R. Alonso, “Dynamic hierarchical caching in large-scale distributed file systems,” presented at 12th International Conference on Distributed Computing Systems, Yokohama Japan, 1992.

[8] R. Malpani, J. Lorch, and D. Berger, “Making world wide web caching servers cooperate,” presented at Fourth International World Wide Web Conference, Boston MA, 1995.

[9] D. Wessels and K. Claffy, “Internet Cache Protocol (ICP), version 2,” RFC 2186, September 1997.

[10] D. Wessels and K. Claffy, “ICP and the Squid web cache,” *IEEE Journal on Selected Areas in Communications*, vol. 16, pp. 345-357, 1998.

[11] L. Fan, P. Cao, J. Almeida, and A. Z. Broder, “Summary cache: a scalable wide-area web cache sharing protocol,” presented at ACM SIGCOMM, 1998.

[12] A. Iyengar and J. Challenger, “Improving Web server performance by caching dynamic data,” presented at USENIX Symposium on Internet Technologies and Systems, 1997.

[13] P. Cao, J. Zhang, and K. Beach, “Active cache: caching dynamic contents on the web,” presented at Middleware ‘98, 1998.

[14] S. Bhatacharjee, K. L. Calvert, and E. Zegura, “Self-organizing wide-area network caches,” Georgia Institute of Technology GIT-CC-97/31, 1997.

[15] A. Heddaya, S. Mirdad, and D. Yates, “Diffusion-based caching along routing paths,” presented at NLANL Web Caching Workshop, Boulder CO, 1997.

[16] Z. Wang and J. Crowcroft, “Cachemesh: A distributed cache system for world wide web,” presented at NLANL Web Caching Workshop, Boulder CO, 1997.

[17] L. Zhang, S. Floyd, and V. Jacobson, “Adaptive Web Caching,” Initial proposal, February 1997.

[18] S. Michel, K. Nyugen, A. Rosenstein, L. Zhang, S. Floyd, and V. Jacobson, “Adaptive web caching: towards a new global caching architecture,” presented at Third International WWW Caching Workshop, Manchester England, 1998.

[19] D. e. a. Nagle, “Active networking for storage: exploiting active networks for network-attached storage,” Carnegie Mellon University Proposal to DARPA BAA 98-03, 1998.

[20] P. Cao and S. Irani, “Cost-aware WWW proxy caching algorithms,” presented at USENIX Symposium on Internet Technologies and Systems, 1997.

[21] B. Kantor and P. Lapsley, “Network News Transfer Protocol,” RFC 977, February 1986.

[22] K. Lidl, J. Osborne, and J. Malcolm, “Drinking from the firehose: multicast USENET news,” presented at USENIX 1994 Winter Conference, 1994.

[23] P. B. Danzig, D. Delucia, and K. Obraczka, “Massively replicating services in wide-area internetworks,” University of Southern California 1994.

[24] K. Obraczka, “Massively replicating services in wide-area internetworks,” : University of Southern California, 1994.

[25] M. Beck and T. Moore, “The Internet2 distributed storage infrastructure project: an architecture for Internet content channels,” presented at Third International WWW Caching Workshop, Manchester England, 1998.

[26] O. Wolfson, S. Jajodia, and Y. Huang, “An adaptive data replication algorithm,” *ACM Transactions on Database Systems*, vol. 22, pp. 255-314, 1997.

[27] M. Baentsch, L. Baum, G. Molter, S. Rothkugel, and P. Sturm, “Enhancing the web's infrastructure - from caching to replication,” *IEEE Internet Computing*, vol. 1, pp. 18-27, 1997.

[28] M. e. a. Baentsch, “Quantifying the overall impact of caching and replication in the web,” University of Kaiserslautern February 1997.

[29] J. Gwertzman and M. Seltzer, “The case for geographical push-caching,” presented at 5th Annual Workshop on Hot Operating Systems, 1995.

[30] J. Gwertzman and M. Seltzer, “Autonomous Replication Across Wide-Area Internetworks,” *SOSP*, pp. 234-, 1995.

[31] T. M. Kroeger, D. D. E. Long, and J. C. Mogul, “Exploring the bounds of Web latency reduction from caching and prefetching,” presented at USENIX Symposium on Internet Technologies and Systems, 1997.

[32] Z. Wang and J. Crowcroft, “Prefetching in the world wide web,” presented at IEEE Global Internet, London UK, 1996.

[33] A. Bestavros, “Demand-based document dissemination to reduce traffic and balance load in distributed information systems,” presented at IEEE Symposium on Parallel and Distributed Processing, San Antonio TX, 1995.

[34] A. Bestavros and C. Cunha, “Server-initiated document dissemination for the WWW,” in *IEEE Data Engineering Bulletin*, vol. 19, 1996, pp. 3-11.

[35] A. Bestavros, “WWW traffic reduction and load balancing through server-based caching,” in *IEEE Concurreny*, vol. 5, 1997, pp. 56-67.

[36] E. Markatos and C. Chronaki, “A top-10 approach to prefetching the web,” presented at Internet Society INET'98, Geneva Switzerland, 1998.

[37] J. Almeida, M. Daby, A. Manikutty, and P. Cao, “Providing differentiated levels of service in web content hosting,” presented at Sigmetrics Workshop on Internet Server Performance, 1998.

[38] D. D. Clark and D. L. Tennenhouse, “Architectural considerations for a new generation of protocols,” presented at ACM SIGCOMM, 1990.

[39] D. Ferrari and D. C. Verma, “A scheme for real-time channel establishment in wide-area networks,” *IEEE Journal on Selected Areas in Communications*, vol. 8, pp. 368-379, 1990.

[40] D. Ferrari, “Client requirements for real-time communication services,” *IEEE Communications Magazine*, vol. 28, pp. 65-72, 1990.

[41] R. Braden, D. Clark, and S. Shenker, “Integrated services in the Internet architecture: an overview,” RFC 1633, June 1994.

[42] ATMForum, “Traffic management specification version 4.0.,” ATM Forum Technical Committee April 1996.

[43] H. Ohnishi, T. Okada, and K. Noguchi, “Flow control schemes and delay/loss tradeoff in ATM networks,” *IEEE Journal on Selected Areas in Communications*, vol. 6, pp. 1609-1616, 1988.

[44] L. Zhang, S. Deering, D. Estrin, S. Shenker, and D. Zappala, “RSVP: a new resource ReSerVation Protocol,” *IEEE Network*, vol. 7, pp. 8-18, 1993.

[45] J. Y. Hui, “Resource allocation for broadband networks,” *IEEE Journal on Selected Areas in Communications*, vol. 6, pp. 1598-1608, 1988.

[46] R. Guerin, H. Ahmadi, and M. Naghshineh, “Equivalent capacity and its application to bandwidth allocation in high-speed networks,” *IEEE Journal on Selected Areas in Communications*, vol. 9, pp. 968-981, 1991.

[47] F. Kelly, “Effective bandwidth at multi-class queues,” *Queuing Systems*, vol. 9, pp. 5-16, 1991.

[48] J. M. Hyman, A. A. Lazar, and G. Pacifici, “A separation principle between scheduling and admission control for broadband switching,” *IEEE Journal on Selected Areas in Communications*, vol. 11, pp. 605-616, 1993.

[49] S. Jamin, P. Danzig, S. Shenker, and L. Zhang, “A measurement-based admission control algorithm for integrated service packet networks,” *IEEE/ACM Transactions on Networking*, vol. 5, pp. 56-70, 1997.

[50] H. Zhang, “Service disciplines for guaranteed performance service in packet-switching networks,” *Proceedings of the IEEE*, vol. 83, pp. 1374-1399, 1995.

[51] A. Demers, S. Keshav, and S. Shenker, “Analysis and simulation of a fair queueing algorithm,” in *Journal of Internetworking: Research and Experience*, vol. 1, 1990, pp. 3-26.

[52] S. Keshav, “On the efficient implementation of fair queueing,” *Internetworking: Research and Experiences*, vol. 2, pp. 157-173, 1991.

[53] A. K. Parekh, “A generalized processor sharing approach to flow control in integrated services networks,” in *Department of Electrical Engineering and Computer Science*. Cambridge MA: Massachusetts Institute of Technology, 1992.

[54] S. Floyd and V. Jacobson, “Link-sharing and resource management models for packet networks,” *IEEE/ACM Transactions on Network*, vol. 3, 1995.

[55] S. Floyd and V. Jacobson, “Random early detection gateways for congestion avoidance,” *IEEE/ACM Transaction on Networking*, vol. 1, pp. 397-413, 1993.

[56] R. Cocchi, D. Estrin, S. Shenker, and L. Zhang, “A study of priority pricing in multiple service class networks,” presented at ACM SIGCOMM, 1991.

[57] S. H. Low and P. P. Varaiya, “A new approach to service provisioning in ATM networks,” *IEEE/ACM Transactions on Networking*, vol. 1, pp. 547-553, 1993.

[58] M. Honig and K. Steiglitz, “Usage-based pricing and quality of service in data networks,” presented at IEEE INFOCOM, 1995.

[59] J. Sairamesh, D. F. Ferguson, and Y. Yemini, “An approach to pricing, optimal allocation and quality of service provisioning in high-speed packet networks,” presented at IEEE INFOCOM, 1995.

[60] S. Shenker, “Service models and pricing policies for an integrated services Internet,” in *Public Access to the Internet*, B. Kahin and J. Keller, Eds.: MIT Press, 1995, pp. 315-337.

[61] D. Clark, “Internet cost allocation and pricing,” in *Internet Economics*, L. McKnight and J. Bailey, Eds.: MIT Press, 1997, pp. 215-252.

[62] A. Gupta, D. O. Stahl, and A. B. Whinston, “Priority pricing of integrated services networks,” in *Internet Economics*, L. McKnight and J. Bailey, Eds.: MIT Press, 1997, pp. 323-352.

[63] D. Songhurst and F. Kelly, “Charging schemes for multiservice networks,” presented at International Teletraffic Congress 15, 1997.

[64] Q. Wang, J. Peha, and M. A. Sirbu, “Optimal pricing for integrated services networks,” in *Internet Economics*, L. McKnight and J. Bailey, Eds.: MIT Press, 1997, pp. 353-378.

[65] G. de Veciana and R. Baldick, “Resource allocation in multi-service networks via pricing: statistical multiplexing,” *Computer Networks and ISDN Systems*, vol. 30, pp. 951-962, 1998.

[66] D. Clark, S. Shenker, and L. Zhang, “Supporting real-time applications in an integrated services packet network: architecture and mechanism,” presented at ACM SIGCOMM, 1992.

[67] S. Shenker, C. Partridge, and R. Guerin, “Specification of guaranteed quality of service,” RFC 2212, September 1997.

[68] S. Shenker and J. Wroclawski, “General characterization parameters for integrated service network elements,” RFC 2215, September 1997.

[69] J. Wroclawski, “Specification of the controlled-load network element service,” RFC 2211, September 1997.

[70] Diffserv, “An architecture for differentiated services,” IETF Diffserv Working Group Work in progress, 1998.

[71] C. Aurrecoechea, A. Campbell, and L. Hauw, “A survey of QoS architectures,” in *Multimedia Systems Journal*, 1998.

[72] J. Rendelman, “Reducing web latency -- Stanford University tries web hosting to boost 'net access,” in *Communications Week*, 1997, pp. 9-.

[73] J. C.-I. Chuang, “Economies of scale in information dissemination over the Internet,” in *Engineering and Public Policy*. Pittsburgh: Carnegie Mellon University, 1998.

[74] M. Labbé, D. Peeters, and J.-F. Thisse, “Location on networks,” in *Network Routing*, vol. 8, *Handbooks in Operations Research and Management Science*, M. O. e. a. Ball, Ed.: Elservier Science B.V., 1995.

[75] S. L. Hakimi, “Optimum locations of switching centers and the absolute centers and medians of a graph,” *Operations Research*, vol. 12, pp. 450-459, 1964.

[76] S. L. Hakimi, “Optimum distribution of switching centers in a communication network and some related graph theoretic problems,” *Operations Research*, vol. 13, pp. 462-475, 1965.

[77] A. Parekh and R. G. Gallagher, “A generalized processor sharing approach to flow control in integrated service network - the multiple node case,” *ACM/IEEE Transactions on Networking*, vol. 2, pp. 137-150, 1994.

[78] P. Lorenzetti, L. Rizzo, and L. Vicisano, “Replacement policies for a proxy cache,” Universita di Pisa October 1996.

[79] S. Williams, M. Abrams, C. R. Standridge, G. Abdulla, and E. A. Fox, “Removal policies in network caches for world-wide web documents,” presented at ACM SIGCOMM, 1996.

[80] Inktomi, “Traffic server's compatibility with advertising and dynamic content,” 1998.

[81] J. J. Kistler and M. Satyanarayanan, “Disconnected operation in the Coda file system,” *ACM Transactions on Computer Systems*, vol. 10, pp. 3-25, 1992.

[82] J. Bennett and H. Zhang, “Hierarchical packet fair queueing algorithms,” *IEEE/ACM Transactions on Networking*, vol. 5, pp. 675-689, 1997.

[83] A. Schill, “Migration, caching and replication in distributed object-oriented systems: an integrated framework,” *IFIP Transactions C (Communication Systems)*, vol. C-6, pp. 309-329, 1992.