Resource Allocation for *stor-serv*: Network Storage Services with QoS Guarantees

John Chung-I Chuang
University of California at Berkeley
chuang@sims.berkeley.edu

Abstract

There is increasing demand from content providers for distributed network storage services that go beyond traditional caching and replication. Through the *stor-serv* architecture [1], content providers can obtain storage services with Quality-of-Service (QoS) guarantees to satisfy their application-specific performance requirements. This paper presents a formal resource allocation model for the *stor-serv* architecture. The model can also be extended to solve network storage capacity planning problems. By applying the model to the ARPANET network topology, we are able to make the following observations: (i) services with deterministic guarantees require more network resources than those with statistical or stochastic guarantees; (ii) knowledge of non-uniformity in data access patterns may be exploited to achieve more efficient usage of network resources; (iii) partial replication of collections can improve mapping efficiency; (iv) the number and placement of storage servers in the network can significantly affect mapping efficiency for individual collections; and (v) depending on the relative costs of storage versus transmission resources, and the data access rates of the collection, a transmission-plus-storage solution may result in significant cost savings over a storage-only solution.

1. Introduction

Network caching and replication (or mirroring) are different techniques employed by network service providers (as well as content providers themselves) to facilitate efficient information dissemination to a global Internet audience [2-9]. These techniques rest on the premise that distributing multiple copies of an information object across the network results in: (i) reduced access latency, (ii) reduced bandwidth consumption, (iii) load balancing, and (iv) improved data availability/redundancy.

Caching and replication can be viewed as different classes of distributed network storage services within a Quality-of-Service (QoS) framework [1,10]. Caching as a best-effort service is simple to implement and adaptive to traffic changes. However, objects may be purged from cache at any time, resulting in cache misses (not unlike dropped packets in the transmission domain). In contrast, replication is a guaranteed service which offers predictability, albeit at the cost of explicit resource reservation. A distributed network storage infrastructure such as *stor-serv* would allow content providers to secure storage resources, from a menu of QoS choices, to satisfy their application-specific performance requirements.

This paper extends the work in [1] by considering the resource allocation problem for the *stor-serv* architecture. The resource allocation process is initiated by the service requester (e.g., content provider), who forwards a *service specification* to the service provider. The service specification consists of two components, namely *traffic profile* and *performance requirements*. The service provider performs a *resource mapping* function, where the high-level service specification is mapped into low-level resource requirements. Finally, the service provider performs *admission control* to determine if the service can be admitted given current resource availability.

Section 2 provides a formal model of the resource allocation problem. The model is applied to an example network in Section 3 to illustrate its operation, and some general observations are drawn. Extensions to the basic model are presented in the next two sections: capacity planning in Section 4 and the mapping into combination of storage and transmission resources in Section 5. Section 6 concludes the paper.

2. Problem Formulation

Consider a network $G(V,E)$ with vertices V and edges E. Each of the vertices v_i V is a demand point, i.e., it has one or more content consumers that issue requests for data objects. Let S V be the set of supply points, i.e., network nodes where storage servers are installed to host data objects. The number, location and capacity of these servers are determined by a network capacity planning process, which we shall address in Section 4. Presently, they are treated as exogenous parameters, and each storage node s_j S has a variable storage cost of $c_S(j)$ per unit of storage per unit time.

Given the lengths of the links, it is possible to compute the shortest distance between a demand point

$v_i \in V$ and a supply point $s_j \in S$ as $d(i,j)$. This distance may be measured in terms of hop count, geographic distance or other distance metrics (e.g., one that incorporates hop count and link bandwidth [11]). Alternatively, $d(i,j)$ can represent the shortest network delay between nodes v_i and s_j. If we consider the effects of network congestion, i.e., link delay may vary according to changing traffic load conditions, then $d(i,j)$ becomes a random variable. In this case an expected value of network delay may more appropriate. We assume that the choice of replica sites does not affect the aggregate traffic flow pattern, and therefore has no impact on the link delays, i.e., network storage providers are “delay-takers” rather than “delay-makers”.

The content owner requests for storage service by providing a service specification. The specification is made up of two components: *traffic profile* and *performance requirements*.

2.1 Traffic Profile

The simplest traffic profile consists of only two fields: size of the corpus (B_{corpus}) and duration of service (T_d). However, the service requester may provide a richer profile to help improve the mapping efficiency (and thereby reduce storage requirement and service cost).

First of all, the profile may specify a start time T_s in addition to the duration T_d, facilitating advanced reservation of resources.

Next, instead of treating the corpus as a single monolithic object, the requester may choose to specify it as a collection of objects, which we denote as Q. This finer level of granularity may be exploited for performing partial replication of the collection, if desired. Each object $q_k \in Q$ is of size $b(k)$ octets, such that the total size of the corpus is

$$B_{corpus} = \sum_{q_k \in Q} b(k) . \tag{1}$$

We define $g(i,k)$ as the conditional probability that object q_k is requested by some content consumer at vertex v_i given that there is an object request. The marginal probability distribution functions (p.d.f.’s) across data objects in the collection and across network nodes are

$$g_q(k) = \sum_{v_i \in V} g(i,k) \tag{2}$$

and

$$g_v(i) = \sum_{q_k \in Q} g(i,k) \tag{3}$$

respectively. The joint and marginal p.d.f.’s are such that

$$\sum_{v_i \in V} \sum_{q_k \in Q} g(i,k) = \sum_{q_k \in Q} g_q(k) = \sum_{v_i \in V} g_v(i) = 1 . \tag{4}$$

If no *a-priori* information is available for the demand distribution, then it should be assumed that

$$g(i,k) = \frac{1}{\|V\| \, \|Q\|} \quad \forall i,k . \tag{5}$$

Finally, the requester may also furnish access frequency information in the profile. Let λ be the total number of requests for objects in collection Q in the network G per unit time. Then the expected number of requests for object q_k at node v_i within a specific time interval T_d is equal to the product of $g(i,k)$, λ and T_d.

2.2 Performance Requirements

A storage requester may specify performance requirements based on distance, delay, availability or other performance measures. In this paper we focus on delay as measured by network distance, ignoring effects of heterogeneous transmission capacity, server processing capacity, queuing delay and changing traffic conditions. The requester also has a choice over the degree of firmness of the guarantees:

- maximum (worst case) delay bound: $D_{\max} \le \tau_{\max}$
- average delay bound: $D_{avg} \le \tau_{avg}$
- average and worst case delay bounds: $D_{avg} \le \tau_{avg}$ and $D_{\max} \le \tau_{\max}$
- stochastic guarantee: Probability$[d > \tau_{threshold}] \le \varepsilon$

We will formulate the resource mapping problem for each service-class.

2.3 Resource Mapping

The resource mapper determines the minimal set of storage servers $X_h \subseteq S$ that satisfies the traffic profile and performance requirements of the request.

2.3.1 Service with Maximum Delay Bound

For this service, the set of storage servers X_h must be chosen such that the delay bound for data access is met for requests from any demand point $v_i \in V$ for any object $q_k \in Q$. This implies that all storage nodes in

X_h must maintain a full replication of the collection Q. Therefore, the amount of storage capacity to be reserved at each node $x \in X_h$ is $B(x) = B_{corpus}$.

If $d(i,x)$ is the shortest-path distance between vertex v_i and storage server x, $x \in X_h$, then the distance from v_i to the nearest storage server is

$$d(i, X_h) = \min_{x \in X_h} d(i,x) \cdot \tag{6}$$

It follows that the worst-case distance between any demand point in the network and its closest server is

$$D_{\max}(X_h) = \max_{i \in V} d(i, X_h) \cdot \tag{7}$$

Then the resource mapping problem can be expressed as the inverse of the k-center problem [12,13]:

$$\min \sum_{x \in X_h} B(x)\ c_S(x) \tag{8}$$

subject to

$$D_{\max}(X_h) \le \tau_{\max} \ ; \tag{8a}$$
$$X_h \subseteq S. \tag{8b}$$

If all storage nodes have identical variable costs c_S, then the problem can be simplified to the minimization of the required storage capacity:

$$RSC = \sum_{x \in X_h} B(x) \cdot \tag{9}$$

But since $B(x)$ is equal to B_{corpus} for $x \in X_h$, we can further simplify the problem to minimizing the number of replicas needed:

$$h_\tau = \min\{\ h\text{: } X_h \subseteq S;\ D_{\max}\ (X_h) \le \tau_{\max};\ \ h \ge 0 \text{ and integer}\}. \tag{10}$$

Kariv and Hakimi showed that the k-center problem is NP-hard, even for simple networks [14]. Algorithms to tackle the problem are presented in [15,16].

2.3.2 Service with Average Delay Bound

Consider a service which specifies that the average delay of data accesses not exceed τ_{avg}. In this case, based on the demand distribution $g(i,k)$, the resource mapper has to determine the optimal set of storage nodes X_h, and the optimal subset of objects Q_x to be replicated at each node $x \in X_h$, such that the delay requirement is satisfied.

Let $X_k \subseteq X_h$ be the set of storage servers that will keep a copy of object q_k. We can specify the distance from node v_i to the nearest storage server $x \in X_k$ as $d(i,X_k)$. Then the average delay can be computed as:

$$D_{avg}(X_h) = \sum_{v_i \in V} \sum_{q_k \in Q} g(i,k)\ \ d(i, X_k) \cdot \tag{11}$$

At each storage server $x \in X_h$, it will have a subset Q_x of the collection Q. The amount of storage capacity required at node x can be computed as

$$B(x) = \sum_{q_k \in Q_x} b(k) \cdot \tag{12}$$

The mapping problem can be expressed as a variant to the inverse k-median problem:

$$\min \sum_{x \in X_h} B(x)\ \ c_S(x) \tag{13}$$

subject to

$$D_{avg}(X_h) \le \tau_{avg} \ ; \tag{13a}$$
$$X_h \subseteq S. \tag{13b}$$

Again, if all storage nodes have identical variable costs c_S, then the problem can be simplified to the minimization of the required storage capacity as stated in Equation (9).

Intuitively, we expect that the replicas are placed closer to the nodes with the largest numbers of data requests. Furthermore, we may also expect to find that the most popular objects in the collection are most widely replicated. However, these intuitions are not always true. If the demand distribution $g(i,k)$ is not independent in i and k, and there is high correlation between popularity and nodes, then the demand for a highly popular object may originate from a limited geographic area. In this case, a few local copies (with sufficient server processing capacity) may suffice in servicing most of the data requests.

For those applications where partial replication of the collection is not possible, we need to impose the additional constraint: $X_k = X_h$ for all k, or equivalently, $Q_x = Q$ for all $x \in X_h$. This implies that there will be equal number of copies of each of the individual objects, regardless of their difference in access frequency.

2.3.3 Service with Average and Maximum Delay Bounds

A service may specify that the average delay of data accesses not exceed τ_{avg}, and further stipulate a maximum delay bound of $\tau_{\max}$. The mapping problem can be stated as

$$\min \sum_{x \quad X_h} B(x) \quad c_S(x) \qquad (14)$$

subject to

$$D_{\mathrm{avg}}(X_h) \leq \tau_{\mathrm{avg}} \; ; \qquad (14a)$$
$$D_{\max}(X_h) \leq \tau_{\max} \; ; \qquad (14b)$$
$$X_h \quad S. \qquad (14c)$$

2.3.4 Service with Stochastic Guarantees

Finally, the mapping problem for a service with stochastic guarantees on delay bounds may be expressed as

$$\min \sum_{x \quad X_h} B(x) \quad c_S(x) \qquad (15)$$

subject to

$$\sum_{v_i \quad V} \sum_{q_k \quad Q} g(i,k) \Big|_{d(i,X_k) > \tau_{threshold}} \leq \varepsilon \; ; \qquad (15a)$$
$$X_h \quad S. \qquad (15b)$$

2.4 Admission Control

Each storage node $s_j \quad S$ has total storage capacity $TSC(j,t)$ and committed storage capacity $B_0(j,t)$ at time t. For each $x \quad X_h$, a local admission control decision is made to accept storage request if, for $T_s \leq t \leq (T_s + T_d)$,

$$B(x) + B_0(x,t) \leq TSC(x,t). \qquad (16)$$

In the event that one or more of the storage nodes in X_h return a rejection, the resource mapping process may be repeated. These nodes, however, will have to be excluded from the candidate pool for future iterations of the mapping process for the same request.

3. Resource Mapping for ARPANET

To demonstrate the application of the model, we will consider the resource mapping problem for a network derived from the real world, the early ARPANET (Table 1 and Figure 1). Given the modest size of the network, we can apply the simple enumeration (exhaustive search) technique to the mapping problem. This allows us to explore the various different facets and dimensions of resource mapping.

Table 1. ARPANET Statistics.

Network	ARPANET
Number of nodes	47
Number of links	68
Average node degree	2.89
Network diameter (hops)	9

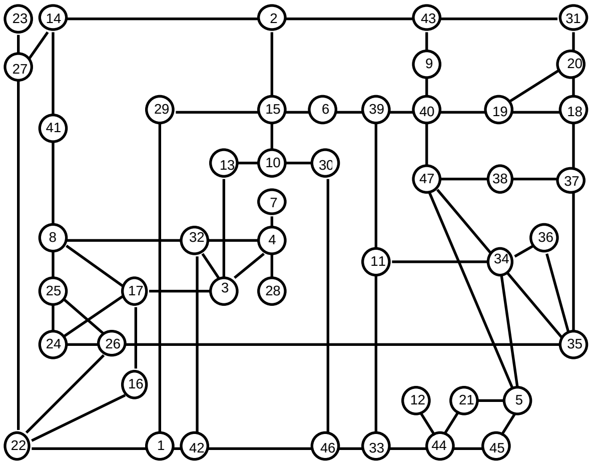


Figure 1. Network topology of early ARPANET.

3.1 Base Case: Uniform Demand Distribution, Unconstrained Replica Locations

The simplest case is to consider the resource mapping of a collection Q with uniform demand distribution across space and objects, i.e., $g(i,k)$ is described by Equation (5). Additionally, there is no constraint on the geographic placement of the replicas, i.e., $S = V$ and replicas may be placed at any node v_i in the network. Finally, all nodes are assumed to have identical per-unit storage costs. The objective is to solve the cost-minimization problems as stated in Equations (8) and (13) for mapping services with maximum and average delay bounds, respectively.

Figure 2 shows the results for both the maximum and average delay bound problems. The discontinuities are a direct result of the fact that only integer numbers of replicas are possible. As expected, a service with a smaller delay bound will require a larger number of replicas. For example, a service with τ_{max} = 4 hops requires two replicas, whereas a service with τ_{max} = 2 hops will require six replicas. From the plot we also observe that given the number of replicas, the achieved average delay bound is always lower than the achieved maximum delay bound.

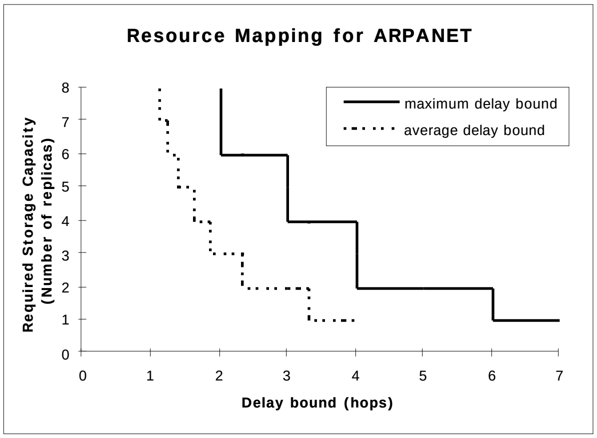


Figure 2. Resource Mapping for ARPANET.

Table 2 shows additional results for the average delay bound mapping problem. Specifically, it shows the delay reductions (in hops) achieved by each additional replica, and the optimal replica locations. It is important to point out that moving from *h* replicas to *h*+1 replicas does not involve the mere addition of a new replica site. Instead, the optimal locations of the *h*+1 replicas can be completely different from those of the *h* replicas. For example, the replica at node 26 is not retained, but replaced by those at nodes 8 and 47, when we move from an *h*=1 to an *h*=2 solution. On the other hand, we see that only 15 out of the 47 nodes in the network are potential replica sites for solutions with up to seven replicas.

Table 2. Resource Mapping for Service with Average Delay Bound.

# of replicas (h)	Avg. Delay (D_{avg})	Delay Reduction (ΔD_{avg})	Replica Locations (X_h)
1	3.32		26
2	2.34	0.98	08,47
3	1.87	0.47	02,03,34
4	1.64	0.23	02,03,33,35
5	1.40	0.23	15,26,32,40,44
6	1.23	0.17	15,19,22,32,34,44
7	1.13	0.11	04,08,15,19,22,34,44

3.2 Non-Uniform Spatial Demand Distribution

As noted in Section 2, any *a-priori* knowledge of the spatial distribution of object accesses may be leveraged to improve the utilization of storage resources for services with average delay bounds. An example of a non-uniform spatial distribution may be:

$$g_k(i) = \begin{cases} 10\ C_1; & i = 1,..,6 \\ C_1; & i = 7,..,47 \end{cases} \qquad (17)$$

where C_1 is a constant such that condition (4) is satisfied. This means that nodes 1 through 6 each experience ten times more requests than nodes 7 through 47. Figure 3 shows the improvements in average delay for this distribution over a uniform spatial distribution. Specifically, a service with τ_{avg} = 1.5 hops will only require three replicas, rather than five replicas in the uniform spatial distribution case.

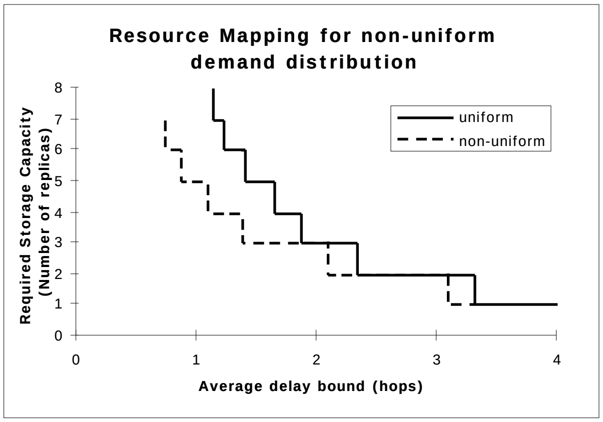


Figure 3. Resource mapping for non-uniform spatial distribution.

3.3 Partial Replication of Multi-Object Collection

A multi-object collection may have a non-uniform demand distribution $g_q(k)$ across its individual objects, i.e., some objects in the collection are more frequently accessed than others. If the collection owner can convey this distribution function to the resource mapper, the latter can leverage this information to improve the resource utilization for services with average delay bounds. Specifically, by allowing partial replication of the collection, the resource mapper can independently determine the optimal number of copies of each individual object. In particular, if $g(i,k)$ is independent across *i* and *k*, then there will be more copies of the more frequently accessed objects.

Consider a multi-object collection *Q*, where all objects $q_k \in Q$ are of identical size. The collection has a uniform spatial demand distribution, i.e.,

$$g_v(i) = \frac{1}{\|V\|} \quad \forall v_i \in V. \qquad (18)$$

However, the collection has a non-uniform demand distribution across its objects. Specifically, the object access pattern obeys Zipf's distribution [17]:

$$g_q(k) = \frac{C_2}{k} \qquad (19)$$

where C_2 is a constant such that condition (4) is satisfied. Figure 4 shows, for a multi-object collection, the efficiency gains of a mapping solution based on partial replication over one based on full replication. The full replication solution is constrained in two ways: (i) the addition of each new full replica necessarily means the addition of $||Q||$ object-copies, (ii) there has to be equal number of copies of each object.

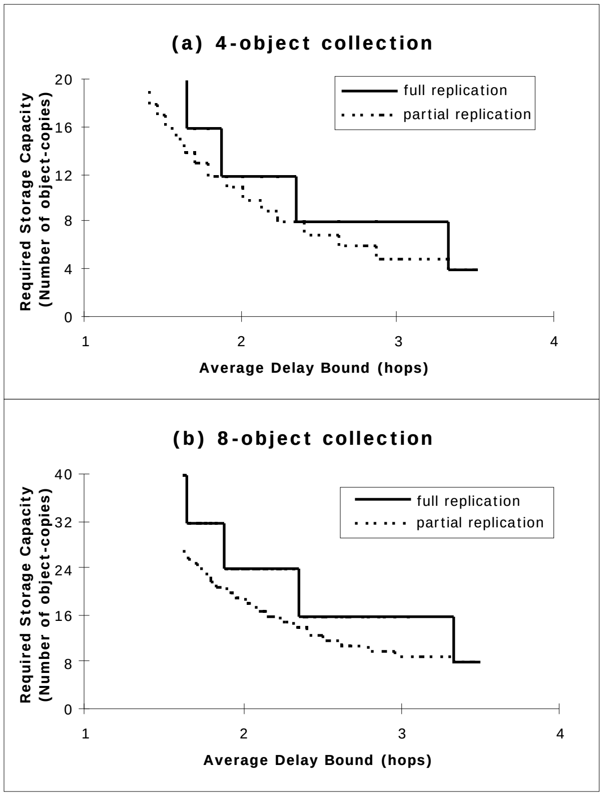


Figure 4. Resource mapping for multi-object collections with non-uniform object distribution using full or partial replication.

Let us consider a service for a four-object collection with τ_{avg} of 2.30 hops. The mapping solution based on full replication will require three full replicas (of four objects each, resulting in a total of twelve object copies) at nodes 2, 3, and 34 (see Table 2). On the other hand, the mapping solution based on partial replication will only require eight object copies (see appendix for solution method and full results). The eight object copies include: three copies of q_1, at nodes 2, 3, and 34; two copies each of q_2 and q_3, at nodes 8 and 47; and one copy of q_4 at node 26. For purposes of admission control, this translates into $B(2) = B(3) = B(34) = b(q_1)$, and $B(8) = B(47) = b(q_2) + b(q_3)$, $B(26) = b(q_4)$, and zero otherwise.

Figure 5 shows, for different sizes of collection Q, the resource mapping solution based on partial replication.

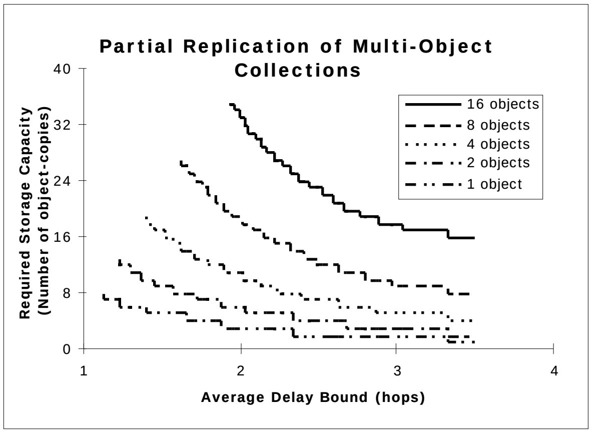


Figure 5. Resource mapping for multi-object collections with non-uniform object distribution using partial replication.

4. Network Storage Capacity Planning Problem

So far we have concentrated on the resource mapping problem (as opposed to the long-term capacity planning problem), treating the fixed cost of setting up a replication server as sunk, and focusing on the marginal cost of storage. This implies that there are no economies of scale in network storage costs, and the cost of reserving 1GB of storage at each of ten nodes is equal to the cost of reserving 10GB of storage at a single node.

In the long run, however, the network storage service provider does have to take the fixed cost into consideration. If this cost were negligible, the network storage service provider may choose to install replication servers at every network node, i.e., $S = V$, and the resource mapper will have the freedom to place replicas at any location within the network. On the other hand, if the fixed cost is substantial, the network storage service provider may want to install replication servers at only a subset of all nodes in the network, i.e., $S \quad V$.

The long term network storage capacity planning problem is really an extension to the short term resource mapping problem. Whereas we were interested in determining the optimal number and placement of the object replicas, i.e., h and X_h, in the resource mapping problem, we are now concerned with the optimal number, placement and capacity of the replication servers, i.e., $||S||$, S and $TSC(s_j) \quad s_j \quad S$. Furthermore, the cost function to be minimized now includes a fixed cost c_0 that is incurred each time a new replication server is installed. For a network storage provider who wishes to achieve an average delay bound τ_{avg} for its aggregate traffic $Q_{aggregate}$, the network storage capacity planning problem can be stated as:

$$\min \sum_{sj \quad S} c_0(s_j) + TSC(s_j) \;\; c_S(s_j) \tag{20}$$

subject to

$$D_{avg}(S) \leq \tau_{avg} \, ; \tag{20a}$$

$$S \quad V. \tag{20b}$$

As a starting point, it appears reasonable for the provider to determine the optimal replication server set based upon the demand patterns aggregated across all collections in its target market. If the number of collections is large enough, and the traffic patterns across different collections are not strongly correlated, then the rate of change of the aggregate $g(i,k)$ should not necessitate frequent and rapid changes in S.

We propose three heuristic solutions to the capacity planning problem, namely full replication, greedy partial replication and conservative partial replication. In the full replication strategy, the network storage provider installs $||S||$ replication servers, and each server has the same amount of storage capacity, i.e., $TSC(s_j) = ||Q_{aggregate}||$ for all s_j. Both of the partial replication strategies allow different total storage capacities to be installed at different servers, and $TSC(s_j) \leq ||Q_{aggregate}||$. In the greedy strategy, the goal is to minimize the global storage capacity (summation of $TSC(s_j)$ across all replication servers), regardless of the number of replication servers needed. In the conservative strategy, the primary goal is to minimize the number of replication servers, and the secondary goal is to minimize global storage capacity. Note that the different heuristics may yield different solutions of $||S||$ and global storage capacity for the same delay target.

These heuristics are evaluated using the ARPANET example. Assume that the aggregate $g(i,k)$ has a uniform spatial distribution across nodes v_i and a Zipfian distribution across objects q_k (i.e., similar to that in Section 3.3). Figure 6 shows, for different fixed costs c_0 (relative to c_S), the total storage cost incurred by the different strategies.

When the fixed cost of installing a new replication server is zero, the greedy partial replication strategy will realize the lowest cost solution among the three approaches (Figure 6(a)). In fact this solution is also the optimal solution to the planning problem when $c_0 = 0$. The similarity between Figure 6(a) and Figure 4 reminds us that this solution is basically that of the short-term resource mapping problem, where fixed cost was assumed sunk.

When the fixed cost of installing a replication server is non-zero, the conservative strategy becomes a more attractive heuristic, since it attempts to minimize the number of servers before minimizing global storage capacity. This strategy will yield a solution that is reasonably close to, if not equal to the optimal solution, especially in high fixed cost conditions. Finally, we note that at very high fixed costs, the variable cost of storage becomes negligible, and the full replication strategy will perform almost as well as the conservative partial replication strategy (Figure 6(d)).

4.1 Resource Mapping with Constrained Replication Server Sites

We have stated the network capacity planning problem and proposed solution strategies based upon the demand distribution of the traffic in its aggregate. However, each individual collection may have its collection-specific $g(i,k)$ similar to or different from the aggregate $g(i,k)$. How will the choice of the constrained replication server set S affect the mapping efficiency of specific collections?

Consider the scenario where $c_0/c_S = 100$, and the network storage provider chooses to maintain five replication servers to achieve an aggregate average delay bound of 1.50 hops (Figure 6(d)). The results from Table 2 (reproduced as the first three columns of Table 3(a)) indicate the optimal locations of the servers are at nodes 15, 26, 32, 40 and 44.

With this new constraint of $S = \{15,26,32,40,44\}$, it is possible to perform resource mapping for services with spatial demand distributions similar to or different from the aggregate distribution. The right three columns of Table 3(a) summarize the mapping result for a collection Q_1 that has a uniform spatial demand distribution as described by Equation (18), i.e., the collection-specific distribution is identical to the aggregate distribution. We observe that the h=1 and h=5 solutions are identical to the unconstrained scenario, and therefore incur no penalty in realized average delay. On the other hand, mapping solutions with two to four replicas will incur a delay penalty of between two and five percent.

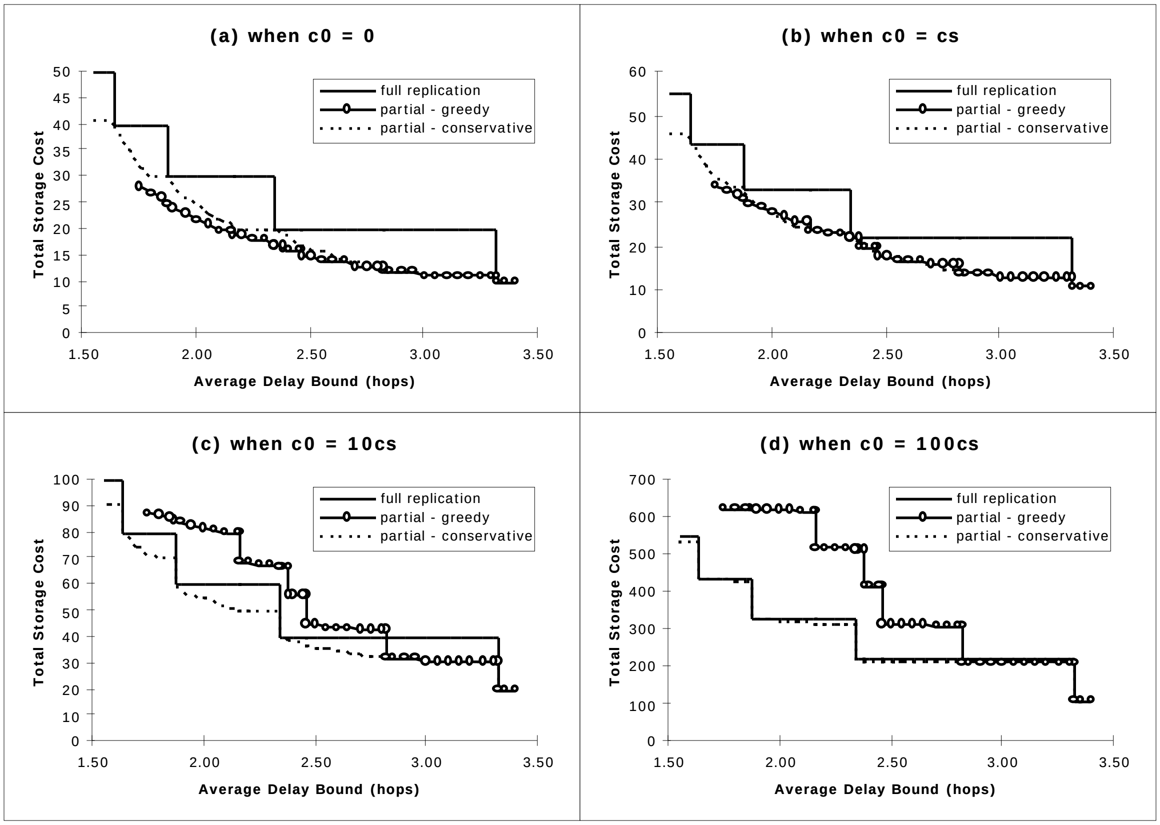


Figure 6. Network capacity planning problem - different replication strategies may realize the lowest cost solution depending on the relative magnitudes of fixed (c_0) and variable (c_S) costs.

Table 3. Comparing mapping efficiencies for constrained versus unconstrained replication server sites.

	Unconstrained: *S* = *V*		**Constrained: *S* = {15,26,32,40,44}**		
No. of replicas	Replica Locations (X_h)	Avg Delay	Replica Locations (X_h)	Avg Delay	% penalty in delay
1	26	3.32	26	3.32	0%
2	08,47	2.34	32,40	2.38	2%
3	02,03,34	1.87	26,32,40	1.96	5%
4	02,03,33,35	1.64	15,26,32,40	1.70	4%
5	15,26,32,40,44	1.40	15,26,32,40,44	1.40	0%

(a) collection Q_1 has uniform spatial demand distribution as described by Eqn. (18)

	Unconstrained: *S* = *V*		**Constrained: *S* = {15,26,32,40,44}**		
No. of replicas	Replica Locations (X_h)	Avg Delay	Replica Locations (X_h)	Avg Delay	% penalty in delay
1	15	3.09	15	3.09	0%
2	03,40	2.09	32,40	2.09	0%
3	03,05,15	1.38	15,32,40	1.68	22%
4	01,02,03,05	1.09	15,26,32,40	1.51	38%
5	01,02,03,05,06	0.87	15,26,32,40,44	1.37	57%

(b) collection Q_2 has non-uniform spatial demand distribution as described by Eqn. (17)

Now consider a second collection Q_2 whose spatial demand distribution is not uniform, but as described by Equation (17). The collection-specific demand distribution is now different from the aggregate, and this results in significant mapping inefficiencies as shown in Table 3(b) and Figure 7. For example, an h=5 solution with a constrained S will incur a 57% delay penalty over the unconstrained case. Specifically, a service with τ_{avg} = 1.50 hops will require three replicas (X_h = {3,5,15}) in the unconstrained case, five replicas (X_h = {15,26,32,40,44}) in the constrained case. This translates into a 67% increase in storage capacity requirement.

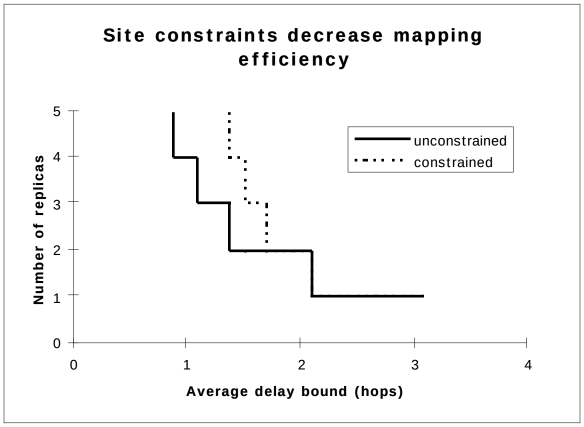


Figure 7. Resource mapping with constrained server sites result in decreased mapping efficiency.

The above example demonstrates that a constrained set of replication servers may result in significant loss of mapping efficiency, especially if the spatial demand distribution of a collection is different from that used for determining the optimal set in the first place. Service providers need to take this fact into account when performing capacity planning. For the content providers, this result should also serve as a reminder that co-location of multiple collections with very different spatial demand distributions may result in a solution that is far from optimal.

5. Mapping into Storage and Transmission Resources

The basic model presented in Section 3 performs mapping into storage resources only, assuming the presence of an underlying best-effort transmission service. In this section, the model is extended to handle mapping into an optimal combination of storage and transmission resources, where the transmission resource provides explicit performance improvements over best-effort service. These transmission resources may be physical transmission capacity (e.g., dedicated leased lines) or may be QoS services based on intserv [18], diffserv [19], or IP "overnet" services such as those offered by Digital Island [20].

Assume that some form of transmission-based QoS is available in the network, such that the delay between vertex v_i and storage server s_j can be reduced from $d(i,j)$ to $d_R(i,j)$ at an additional cost of $c_T(i,j)$ per octet transmitted. Then the reduced delay between vertex v_i and its nearest storage server is

$$d_R(i,X_h) = \min_{x \quad X_h} d_R(i,x) \qquad (21)$$

at a cost of $c_T(i,X_h)$.

Consider a service mapping problem with worst-case delay guarantees. For a given X_h, we can determine the set $V_L \quad V$ such that

$$\max_{i \quad V_L} d_R(i,X_h) \le \tau_{\max} \qquad (22a)$$

and

$$\max_{i \quad V \backslash V_L} d(i,X_h) \le \tau_{\max} . \qquad (22b)$$

Our objective is then to minimize total cost[1]:

$$\min_{\substack{X_h \quad S \\ V_L \quad V}} \left[\int_{t=T_S}^{T_S+T_d} \left\{ \sum_{x \quad X_h} B(x) \; c_S(x) + \sum_{i \quad V_L} \sum_{q_k \quad Q} \lambda \;\; g(i,k) \;\; b(k) \;\; c_T(i,X_k) \right\} dt \right] \qquad (23)$$

From Equation (23) it is clear that there are two cost components associated with storage and transmission respectively. Storage cost is calculated as before: the amount of storage capacity at each node x X_h multiplied by the per-unit storage cost $c_S(x)$. Additional transmission cost is incurred for each node v_i V_L. The cost is calculated by multiplying the amount of requested data by that node within the time period $[T_s,T_s+T_d]$ and the per octet incremental transmission cost c_T.

Consider a resource mapping problem for the ARPANET with τ_{max} = 3 hops. Since we are concerned with worst case delay, we cannot take advantage of any non-uniformity in demand distribution. From Figure 2 we see that this service may be mapped into a solution with four replicas.

Alternatively, this service may be satisfied with a combination of storage and transmission resources, such that those data requests that originate from further than three hops away from the closest replica are

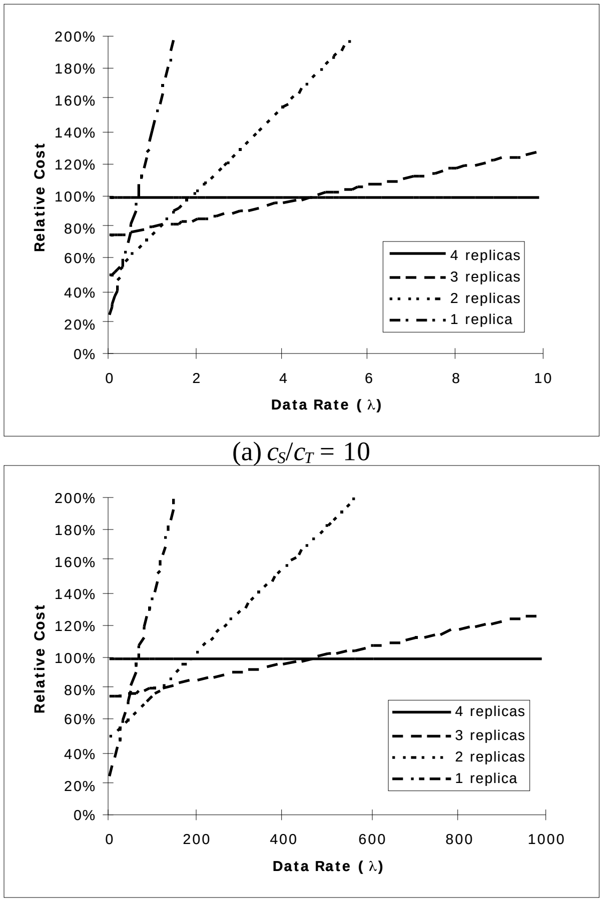
[1] We do not include the transmission cost for initial population and subsequent updates of the replicas here, though it may become significant if the data write-to-read ratio is high.

serviced with additional transmission resources, allowing them to experience service quality comparable to those requests that originate from within three hops of the closest replica.

Figure 8 shows, for two different storage to transmission cost ratios, the cost of various transmission-plus-storage alternatives relative to the storage-only (four replicas) solution. In both cases, we see that each of the four solutions is the optimal solution for a given range of data access rate λ. The storage-only solution is optimal for the frequently accessed objects, since all nodes can be served from less than three hops away and no additional transmission cost is incurred. However, as the frequency of access declines, it becomes more economical to have fewer replicas and pay transmission charges for each data access that originates from more than three hops away. Substantial savings (as much as 70% over storage-only solution in this example) may be realized. It is worthwhile to point out, however, that accurate information on data access rate is crucial when choosing a transmission-plus-storage solution. A higher than expected data access rate may quickly turn the optimized solution into a highly sub-optimal one.

Figure 9 shows the decision diagram for the ARPANET resource mapping problem with τ_{max} = 3 hops, across data rate λ and storage to transmission cost ratio c_S/c_T. Consistent with our expectations, a higher data rate leads to an optimal solution with more replicas, while a higher storage to transmission cost ratio leads to an optimal solution with fewer replicas.

Finally, Figure 10 is a three-dimensional representation of the cost of the various solutions relative to the storage-only solution. It is once again evident that significant cost savings may be achieved if the cost factors and the data rate is well understood and made known to the resource mapping entity.



(a) c_S/c_T = 10

(b) c_S/c_T = 0.1

Figure 8. Resource cost relative to storage-only solution (4 replicas) at different storage to transmission cost ratios.

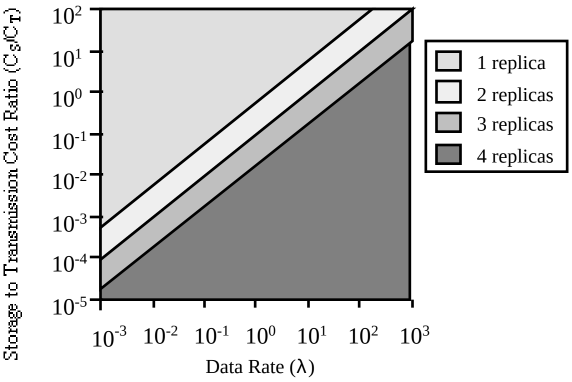


Figure 9. Optimal mapping decision for storage and transmission resources (arpanet with τ_{max} = 3)

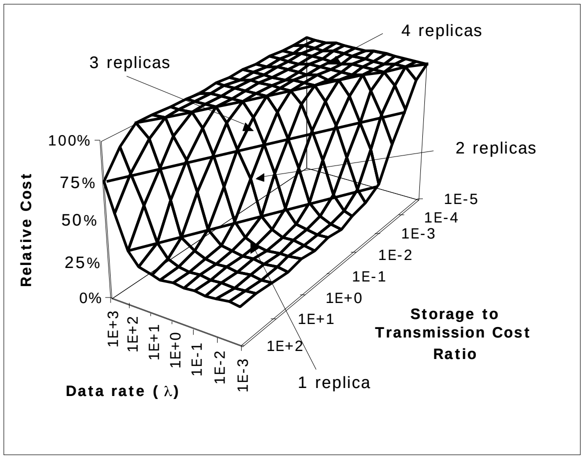


Figure 10. Resource cost relative to storage-only solution (4 replicas).

6. Conclusions and Future Work

This paper has developed a formal resource allocation model for supporting distributed network storage services. These services can provide Quality-of-Service guarantees, satisfying applications with very different traffic profiles and performance requirements. The model is validated with an example network and several observations are drawn: (i) services with deterministic guarantees require more network resources than those with statistical or stochastic guarantees; (ii) knowledge of non-uniformity in data access patterns may be exploited to achieve more efficient resources usage; (iii) partial replication of collections can improve mapping efficiency; (iv) the number and placement of storage servers in the network can significantly affect mapping efficiency for individual collections; and (v) depending on the relative costs of storage versus transmission resources, and the data access rates of the collection, a transmission-plus-storage solution may result in significant cost savings over a storage-only solution. Finally, the resource allocation model is also extended to solve network storage capacity planning problems.

The allocation model may be improved by incorporating other components of the end-to-end delay as perceived by a user. These components may include server queuing, server processing and router queuing delays. Heterogeneity in transmission capacity and server processing capacity will have to be taken into account, together with an appropriate traffic model. There is work on providing QoS support for priority-based request scheduling at the server level [21]. Strong complementaries clearly exist between these two works.

There is clearly a tightly-coupled relationship between admission control and resource mapping in the resource allocation process. When resource utilization level is high, the likelihood of a service request being rejected by the individual resource nodes also rises, and the resource mapping and admission control process may be iterated several times before a success is finally encountered. In this situation, a "greedy" algorithm or a quorum-based algorithm may become appropriate.

Finally, while resource allocation is performed prior to service establishment, resource management may be performed in real-time throughout the duration of the service. Further replication, de-replication and migration of data objects may be warranted in response to traffic shifts and network changes (e.g., new nodes and links, or congestion or failure of same). Consistency in the design of these two modules is important for optimizing overall resource utilization.

References

[1] Chuang, J.C.-I., and M.A. Sirbu. "Distributed network storage service with quality-of-service guarantees." Proceedings of Internet Society INET'99, San Jose CA, June 1999. <http://www.isoc.org/inet99/proceedings/4q/4q_3.htm>.

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

[3] Baentsch, M., L. Baum, G. Molter, S. Rothkugel, and P. Sturm. "Enhancing the web's infrastructure - from caching to replication." *IEEE Internet Computing* 1, no. 2 (1997): 18-27.

[4] Beck, M., and T. Moore. "The Internet2 distributed storage infrastructure project: an architecture for Internet content channels." Third International WWW Caching Workshop, Manchester England, June 1998.

[5] Bestavros, A. "WWW traffic reduction and load balancing through server-based caching." *IEEE Concurreny*, Special Issue on Parallel and Distributed Technology, Jan-Mar 1997, 56-67.

[6] Gwertzman, J., and M. Seltzer. "The case for geographical push-caching." 5th Annual Workshop on Hot Operating Systems, May 1995.

[7] Luotenen, A., and K. Altis. "World-wide web proxies." 1st International Conference on the WWW, May 1994.

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

[9] Obraczka, K. "Massively replicating services in wide-area internetworks." Ph.D. dissertation, University of Southern California, 1994.

[10] Chuang, J.C.-I. "Economies of scale in information dissemination over the Internet." Ph.D. dissertation, Carnegie Mellon University, November 1998.

[11] Ma, Q. "QoS routing in the integrated services network." Ph.D. dissertation, CMU-CS-98-138, Carnegie Mellon University, January 1998.

[12] Hakimi, S.L. "Optimum locations of switching centers and the absolute centers and medians of a graph." *Operations Research* 12 (1964): 450-459.

[13] Hakimi, S.L. "Optimum distribution of switching centers in a communication network and some related graph theoretic problems." *Operations Research* 13 (1965): 462-475.

[14] Kariv, O., and S.L. Hakimi. "An algorithmic approach to network location problems, I: the p-centers." *SIAM Journal of Applied Mathematics*

37 (1979): 513-538.

[15] Halpern, J., and O. Maimon. “Algorithms for the m-center problems: a survey.” *European Journal of Operation Research* 10 (1982): 90-99.

[16] Labbé, M., D. Peeters, and J.-F. Thisse. “Location on networks.” In *Network Routing*, ed. M.O. Ball et al., 8: Elservier Science B.V., 1995.

[17] Zipf, G.K. *Human behavior and the principle of least effort*. Cambridge MA: Addison-Wesley, 1949.

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

[19] Blake, S. et al. “An architecture for differentiated services.” RFC 2475, December 1998.

[20] Rendleman, J. “Reducing web latency -- Stanford University tries web hosting to boost net access.” *Communications Week*, June 30 1997, 9-.

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