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

John Chuang
chuang@sims.berkeley.edu

NetStore'99
October 14 1999

Outline

- Introduction: what is *stor-serv*?
- Resource allocation: model & example

Introduction

- Benefits of distributed network storage
  - Bandwidth savings
  - Latency reductions
  - Data availability/redundancy
  - Load balancing
- Different approaches
  - Caching: network-centric (max hit rate)
  - Replication: publisher-centric (max publisher value)
  - Others: push, pre-fetch, differential caching, etc.

Caching

- Pros:
  - Simple, adaptive (traffic-driven; best effort)
  - Object level granularity
  - Transparent to both publisher and consumer
  - Static cache hierarchy makes resource discovery easy
- Cons:
  - Publisher has no control over placement or replacement (traffic-driven; best effort)
  - Cache misses possible
  - Publisher cannot collect access statistics
  - Static cache hierarchy difficult to change

Replication

- Pros:
  - Publisher controls object placement & replacement
  - Advanced reservation and placement
  - Guaranteed data availability (no cache miss)
  - Easier to arrange for collection of access statistics
- Cons:
  - High setup cost (entire sites, not individual objects)
  - Not as adaptive to changes
  - Not necessarily transparent to consumers (resource discovery non-trivial outside of cache hierarchy)

What is *stor-serv*?

- Unified network storage service framework
- Inspired by intserv/diffserv
- Publisher can choose QoS level
  - Best-effort caching
  - Differential caching
  - Push/pre-fetch
  - Guaranteed service object replication

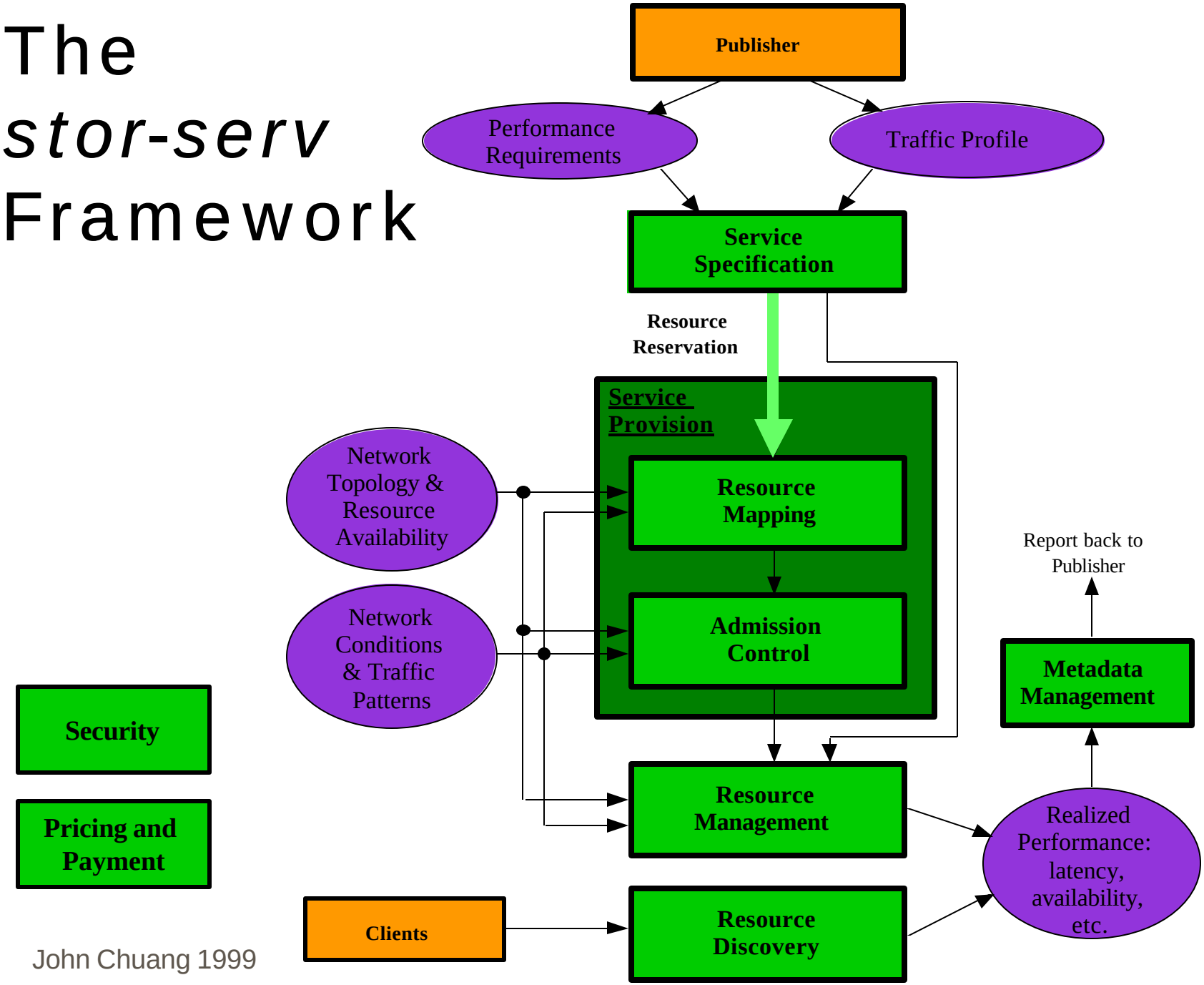
QoS Dimensions: Placement & Replacement

Service Class	Custom Placement	Custom Replacement
Simple Caching		
Differential Caching		✓
Push/pre-fetch	✓	
Replication	✓	✓

stor-serv Efficiencies

- Standardized service semantics + automated resource allocation
  - $\Rightarrow$ low setup cost
  - $\Rightarrow$ adaptability
- Statistical multiplexing
  - Resource reserved for object replication, etc.
  - Left-over capacity for best-effort caching

The *stor-serv* Framework



Service Specification

- Traffic Profile
  - storage capacity
  - time and duration (advance reservation)
  - data access pattern (if known)
- Performance Requirements
  - delay, distance, jitter, availability, etc.
  - deterministic vs. statistical guarantee

Example Replication Services

Deterministic	100kB storage capacity for 1 hour; maximum distance = 4 hops
Average	1GB storage capacity for 1 day; 500ms average network latency
Stochastic	1MB storage capacity for 1 hour; Probability[hops > 4] < ε
Advance Reservation	1GB storage capacity for 1 hour; starting at 11:59pm, Dec 31, 1999; average distance = 2 hops

Resource Allocation

- Resource Mapping
  - service specification → physical resource requirements
  - map into storage & transmission resources
  - facilities-location problem
- Admission Control
  - accept/reject service requests based on utilization level

Network Model

- network $G(V,E)$
- demand points $V=\{v_1,v_2,\ldots,v_i,\ldots\}$
- supply points $S=\{s_1,s_2,\ldots,s_j,\ldots\}$; $S \subseteq V$
- storage cost $c_S(j)$
- incremental transmission cost $c_T(i,j)$

Traffic Profile

- object collection $Q=\{q_1,q_2,\ldots,q_k,\ldots\}$
- object size = $b(k)$
- collection size, $B_{\text{corpus}} = \sum_{q_k \in Q} b(k)$

- reservation start time T_s and duration T_d
- data request rate λ
- data request distribution $g(i,k)$
  - conditional probability that object q_k is requested by some user at node v_i given that there is an object request

Performance Requirements

- worst-case delay: $D_{max} \leq \tau_{max}$
- average delay: $D_{avg} \leq \tau_{avg}$
- stochastic guarantee: $P[D > \tau_{threshold}] \leq \varepsilon$

Resource Mapping

- Find optimal replication set X_h:

$$\min \sum_{x \in X_h} B(x) \cdot c_S(x)$$ (Total Storage Cost)

where $B(x) = \sum_{q_k \in Q_x} b(k)$ (Storage requirement at node x)

s.t. performance requirement(s)

Admission Control

For each node $x \in X_h$, $T_s \leqslant t \leqslant (T_s + T_d)$ test:

$$B(x) + B_0(x,t) \leqslant TSC\ (x,t)$$

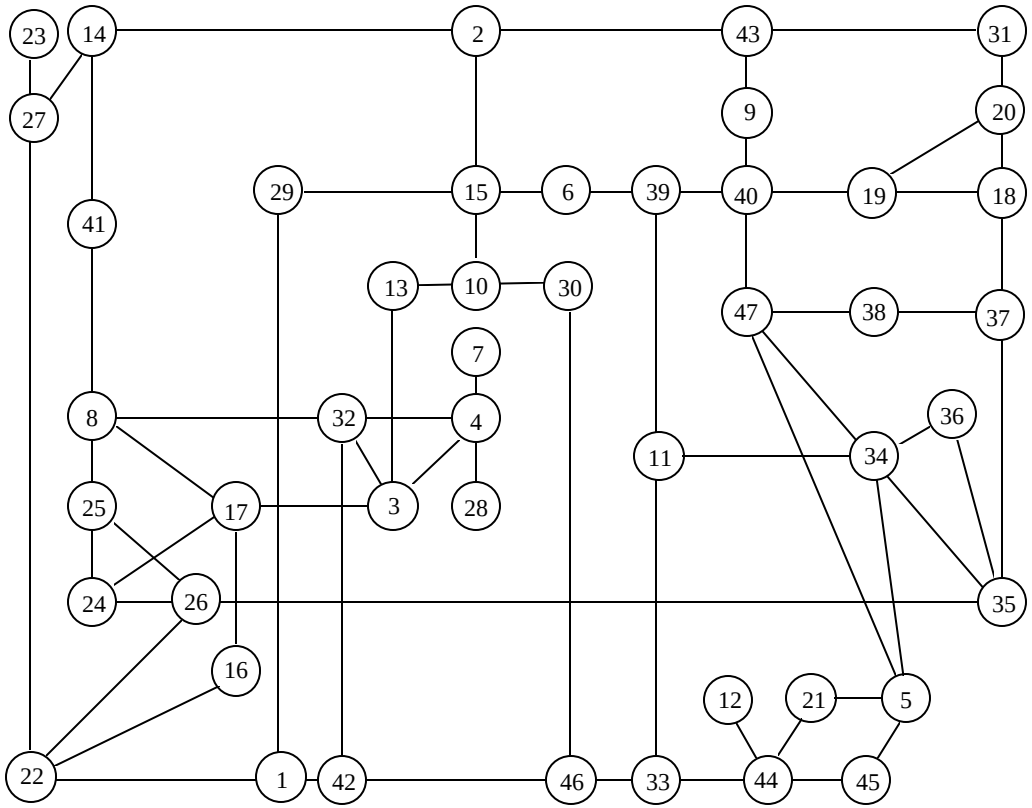
where

$B(x)$ is requested storage capacity

$B_0(x,t)$ is committed storage capacity at time t

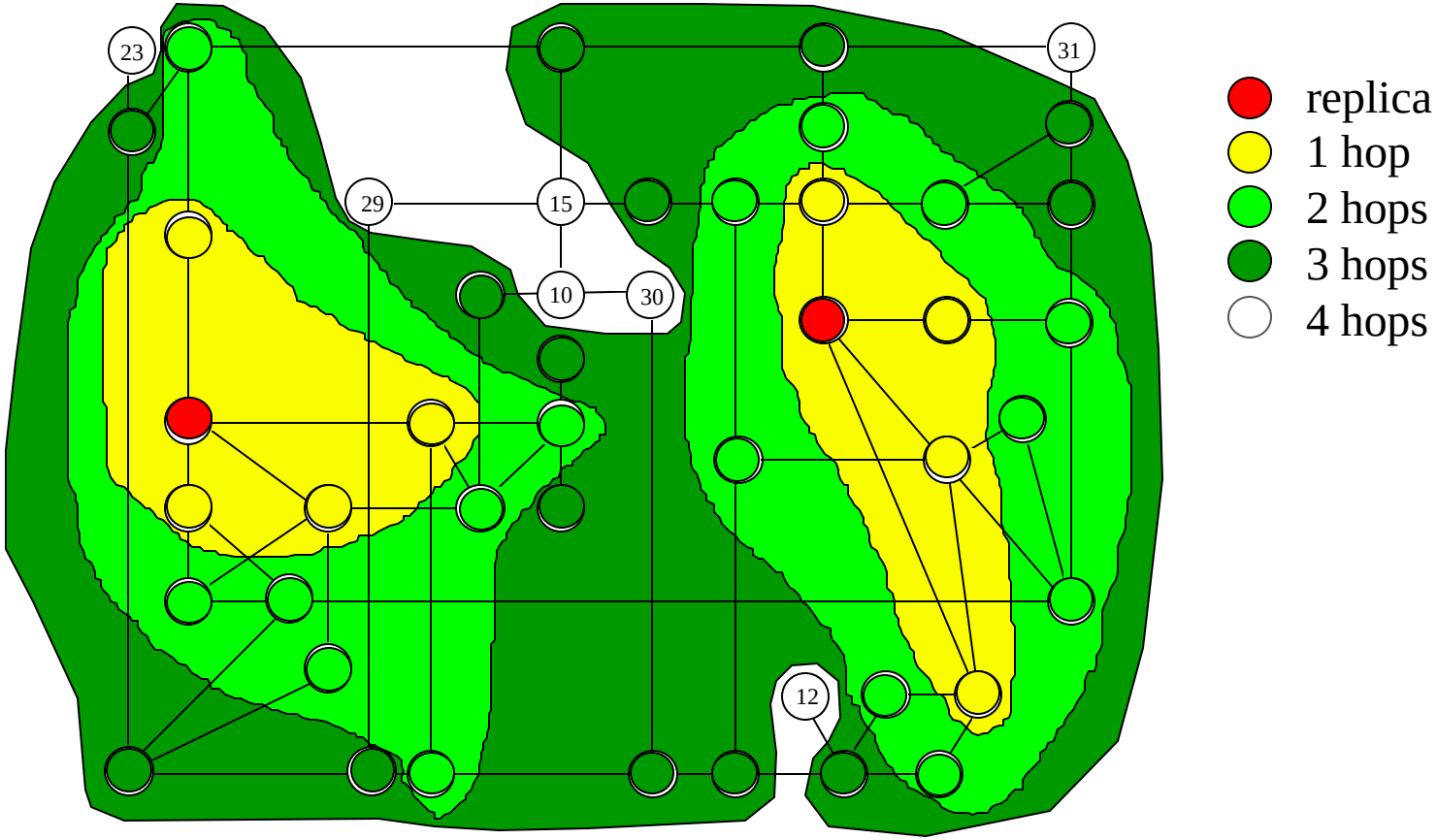
$TSC\ (x,t)$ is total storage capacity

Resource Mapping Example



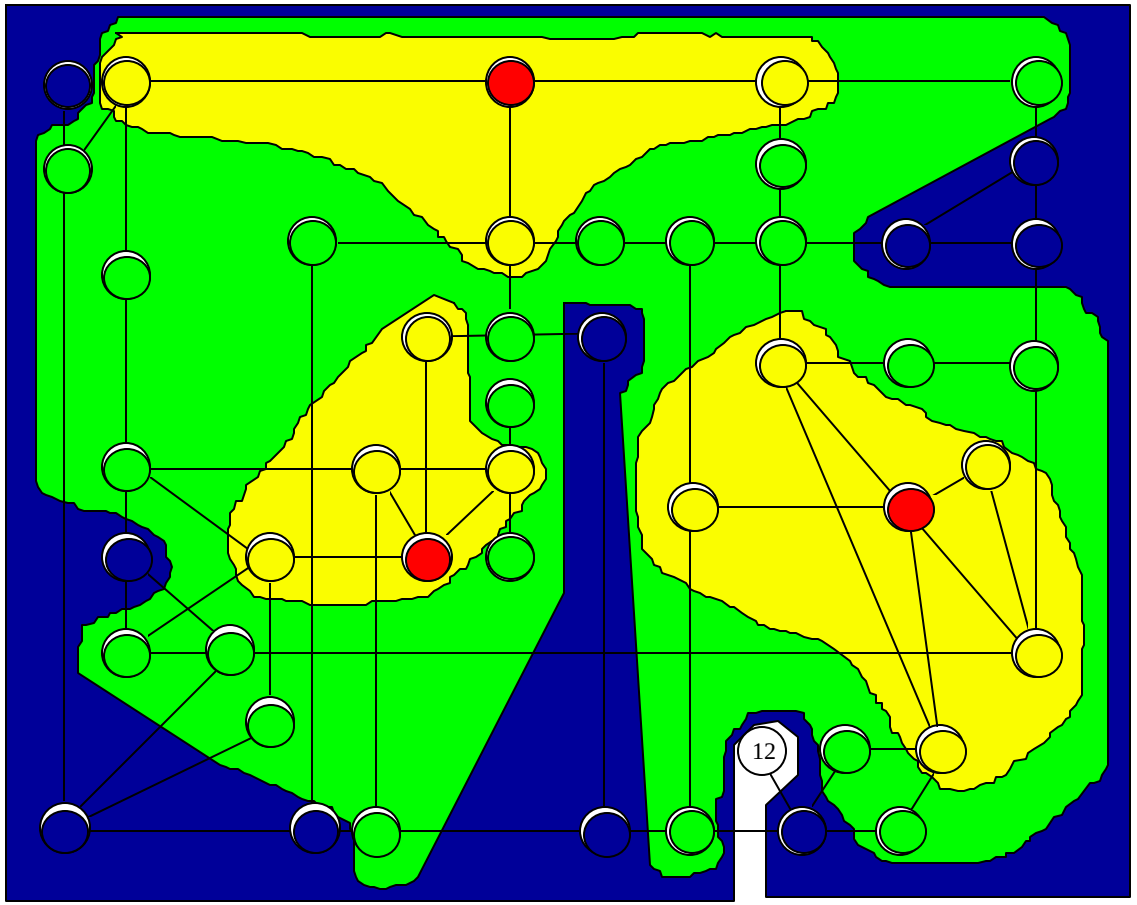
ARPANET
Number of nodes = 47
Number of links = 68
Average node degree = 2.89
Network diameter (hops) = 9

Max Delay Bound = 4 hops



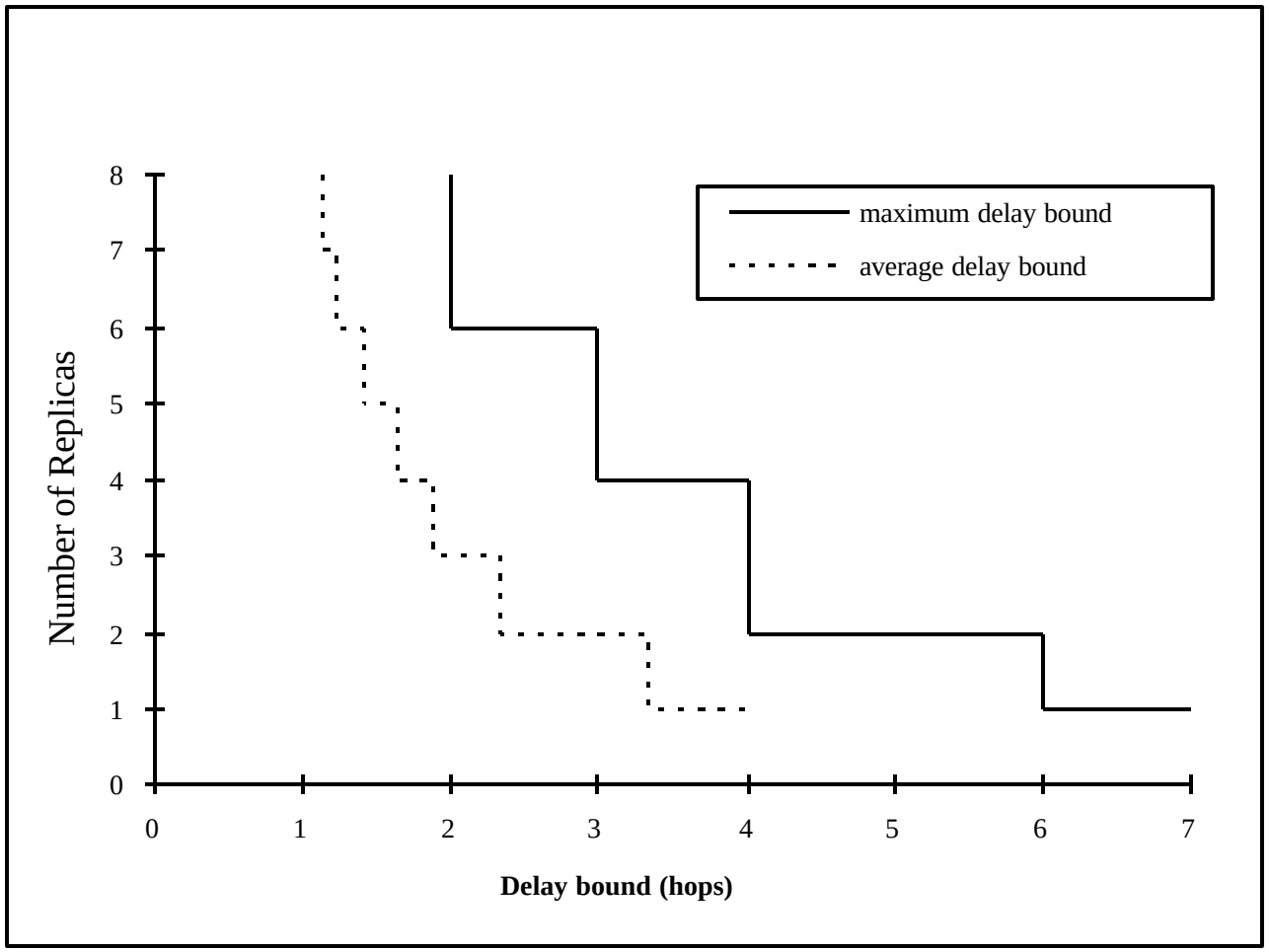
2 replicas: average delay = 2.34 hops; maximum delay = 4 hops

Avg. Delay Bound = 2 hops

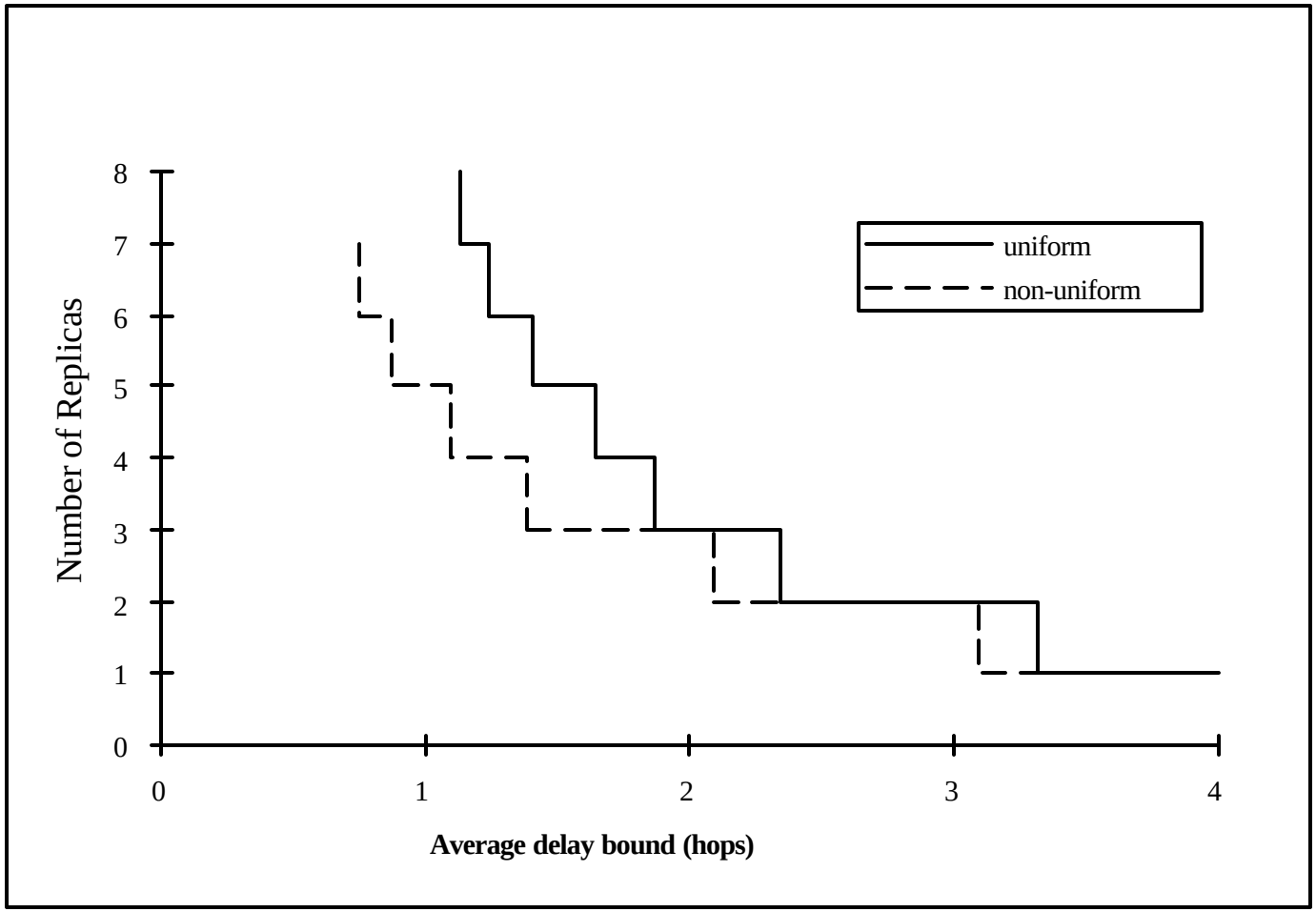


Average Distance = 1.87 hops; Maximum Distance = 4 hops

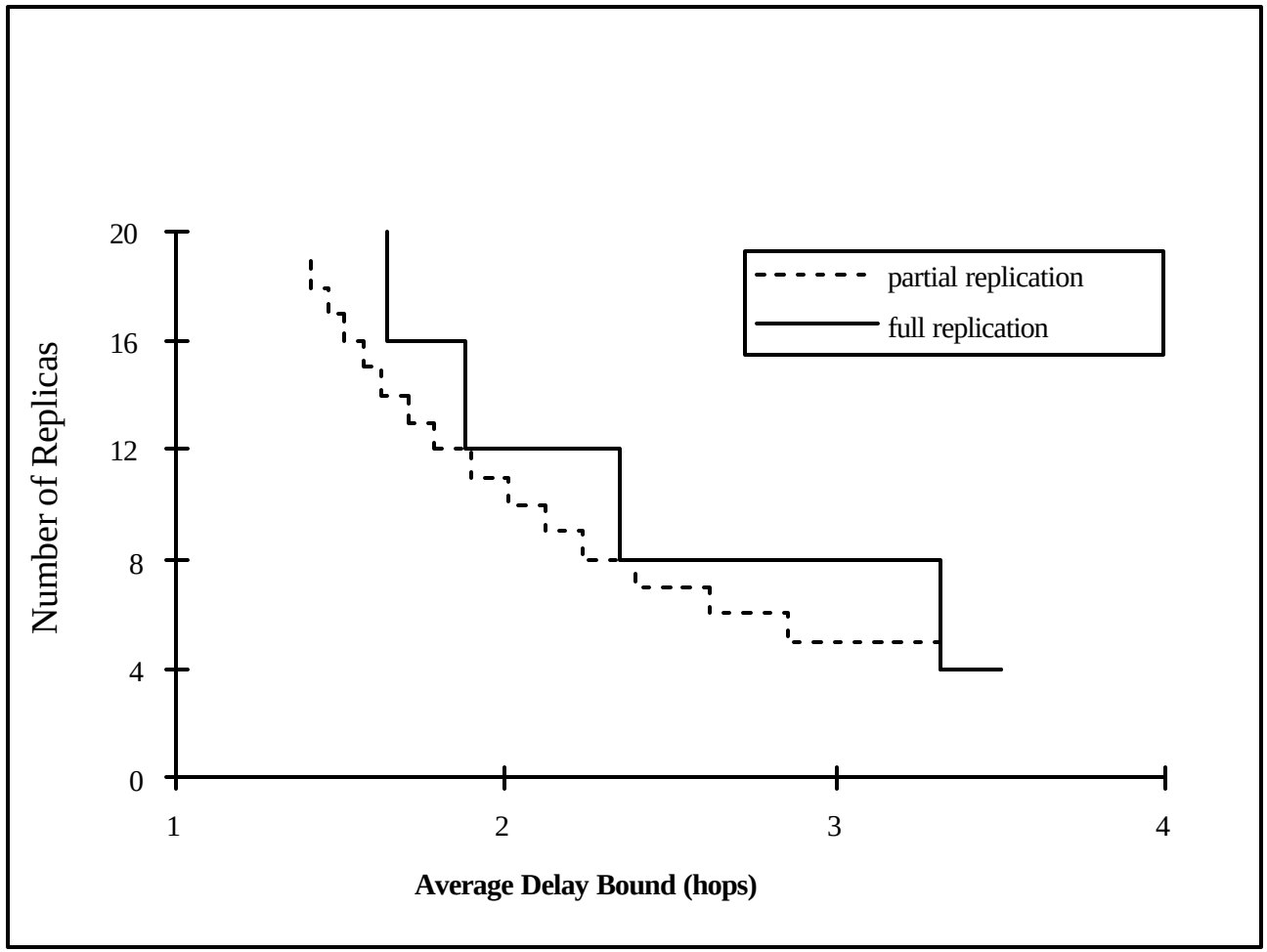
Resource Mapping for Services with Max and Avg Delay Bounds



Non-Uniform Demand Distribution Improves Mapping Efficiency



Mapping Efficiency through Partial Replication



Mapping into Storage & Transmission Resources

- Minimize total cost subject to meeting performance requirements

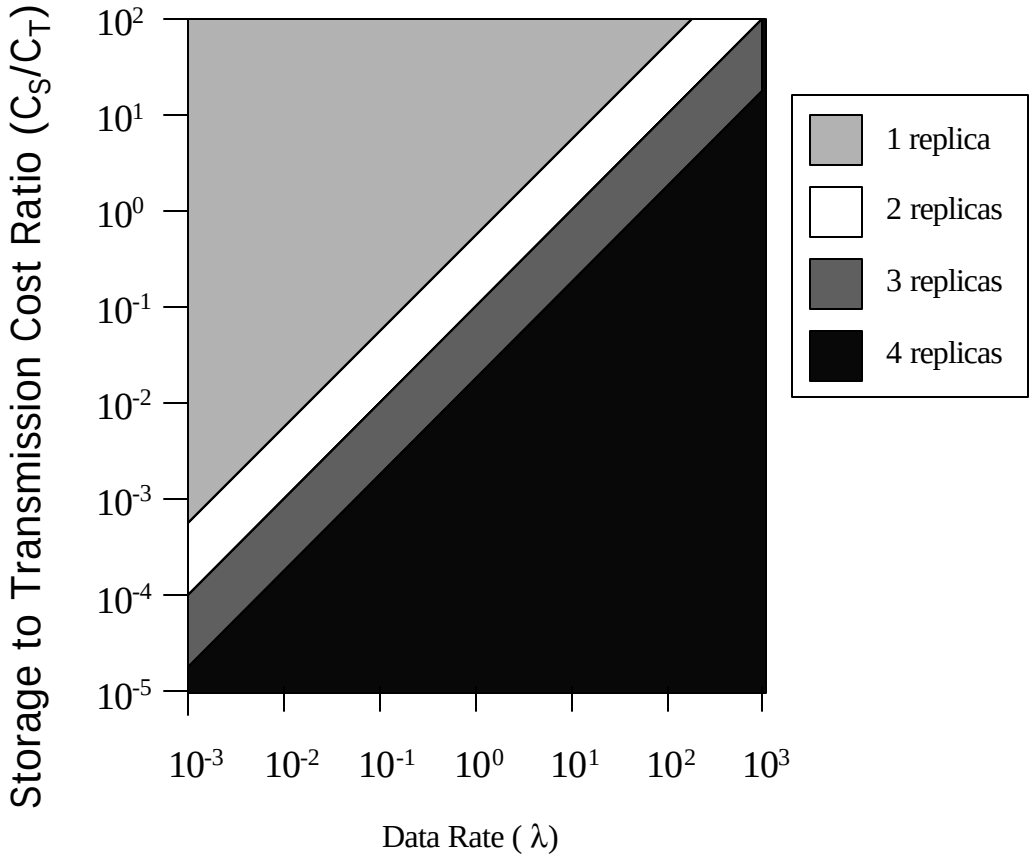
$$\min_{\substack{X_h \subseteq S \\ V_L \subseteq V}} \left[\int_{t=Ts}^{Ts+Td} \left\{ \sum_{x \in X_h} B(x) \cdot c_S(x) + \sum_{i \in V_L} \sum_{q_k \in Q} l \cdot g(i,k) \cdot b(k) \cdot c_T(i, X_k) \right\} \cdot dt \right]$$

storage cost

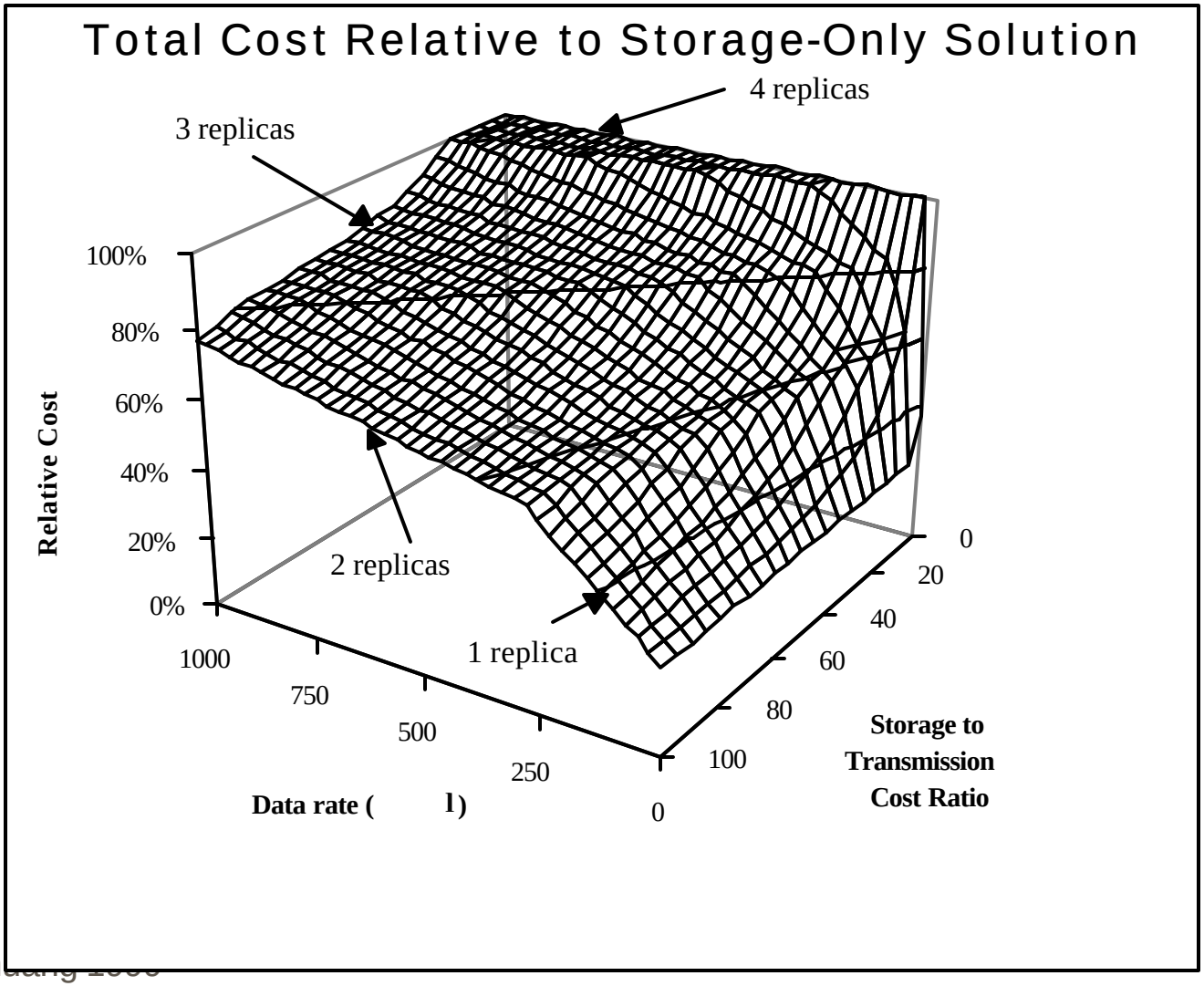
transmission cost

Optimal Solution Depends on

- Relative cost of storage v. transmission
- Object access frequency



Optimal Combination of Storage and Transmission



Conclusion

- There is no one-size-fits-all storage solution
  - *stor-serv* provides unified QoS framework (from caching to replication)
- Resource allocation should be able to support
  - Object level granularity
  - Short service durations
  - Placement (spatial) and replacement (temporal) control
  - Deterministic and stochastic guarantees
  - Mapping into storage and transmission resources