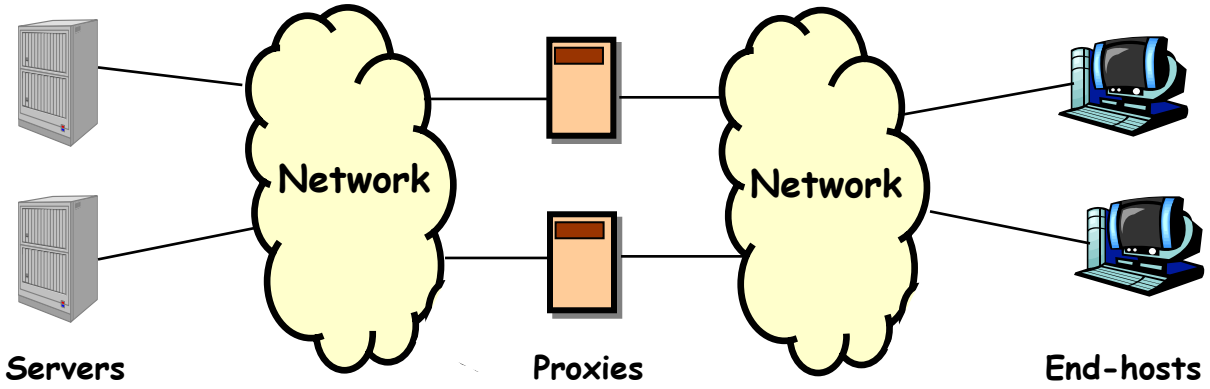
Cooperative Leases: Scalable Consistency Maintenance in CDNs

A. Ninan, P. Kulkarni, P. Shenoy,
K. Ramamritham and R. Tewari

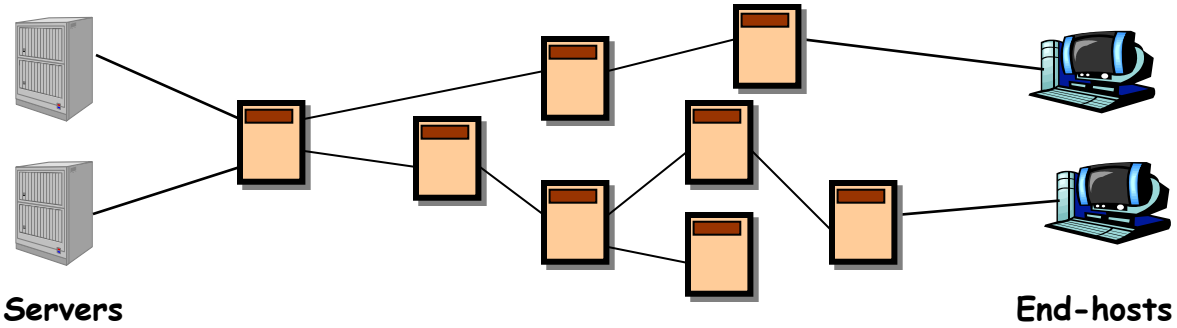
University of Massachusetts and IBM Research

Motivation

- Dramatic growth in world wide web traffic
- Web accesses are non-uniform in nature
  - Create hot-spots of server and network load, increase latency
- *Solution:* employ web proxy caches
  - *Reduces user response times, server load, network load*



Content Distribution Network



- Content distribution network (CDN)
  - Collection of proxies that act as intermediaries between servers and clients
  - Service a client request from "closest" proxy with the object
  - Similar benefits as single proxy environments, but larger scale
- Caching in CDN => must maintain cache consistency
  - Single proxy consistency mechanisms don't scale to CDNs
    - Example: TTL values
- *Goal:* scalable consistency mechanisms for CDNs

Talk Outline

- Motivation
- Cooperative Leases: Design and Implementation
- Experimental Evaluation
- Related Work
- Concluding Remarks

Key Idea: Cooperative Consistency

- Key Idea: CDN proxies cooperate to maintain consistency
  - Cooperation reduces burden on servers
  - Cooperation potentially reduces burden on individual proxies

- Cooperative consistency *orthogonal* to cooperative caching
  - *Coop. caching:* cooperate to service user requests
  - *Coop. consistency:* cooperate to maintain consistency

Cooperative Consistency using Leases

- *Lease:* fixed duration contract between server and proxy
  - Server agrees to notify proxy of all updates to an object over duration *d*
  - "*d*" is the lease duration
  - Lease may be renewed upon expiry
- Limitations of leases for CDNs
  - Server needs to notify each proxy caching the object of an update
    - Excessive burden for popular objects
  - Leases requires a server to maintain state
    - Overhead can be excessive for large CDNs
  - Leases provide strong consistency
    - Overkill for many cached web objects (weak consistency suffices)
- *Problem:* leases don't scale to CDNs

Scaling Leases to CDNS

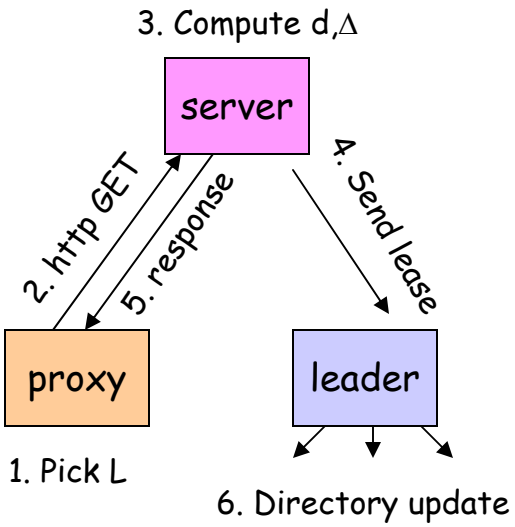
- Problem: excessive notification burden at server
  - Solution: send notification to subset of proxies, proxies forward to others
- Problem: excessive state space overhead
  - Solution: server only maintains state info for leader
    - Leaders maintain information about other proxies caching the object
- Problem: not all objects need strong consistency
  - Solution: associate a rate parameter Δ each lease
    - Send notification no more than once every Δ time units
- Resulting scheme: "Cooperative Leases"

Cooperative Leases: Basics

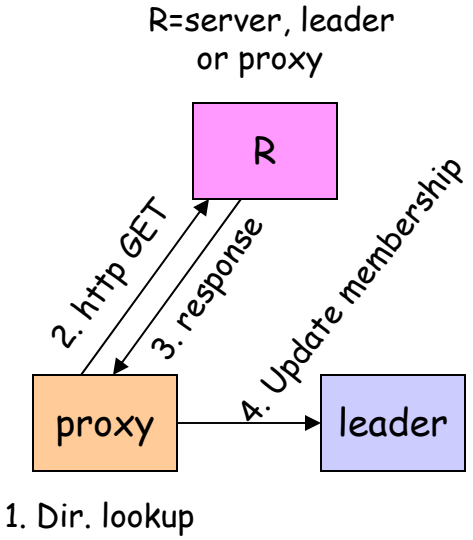
- Use one proxy to represent a group *(leader proxy)*
- Server grants a single lease to the entire group
- Update => send notification only to leader
  - Leader forwards to other proxies in the group
    - Only those proxies caching the object are notified
- Leader renews lease on behalf of entire group

- Different proxies can be leaders for different objects
  - Distribute leader responsibilities across proxies in group

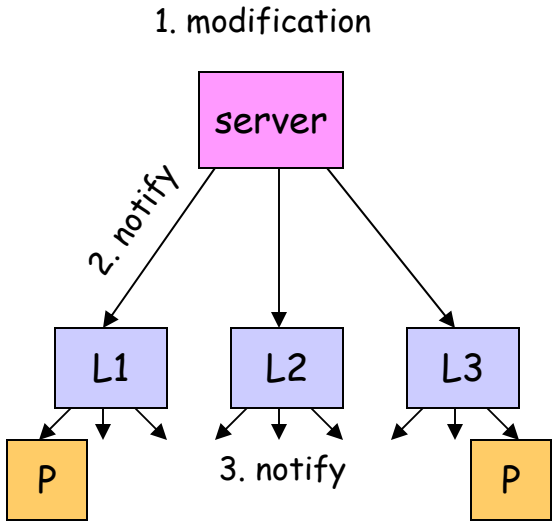
Cooperative Leases: Operations



First-time Requests



Subsequent Requests



Object update

Design Considerations

- How to choose a leader?
  - First proxy is leader
    - Potential imbalance but less communication overheads
  - Use a hashing function: *leader = hash(URL)*
    - Better load balancing, potentially more communication

- When should a leader renew a lease?
  - Eager renewals: renew while there are interested proxies
    - Use terminate message to indicate lack of interest
    - Suitable for popular objects
  - Lazy renewals: renew only if a proxy makes subsequent requests after expiry
    - Suitable for less popular objects

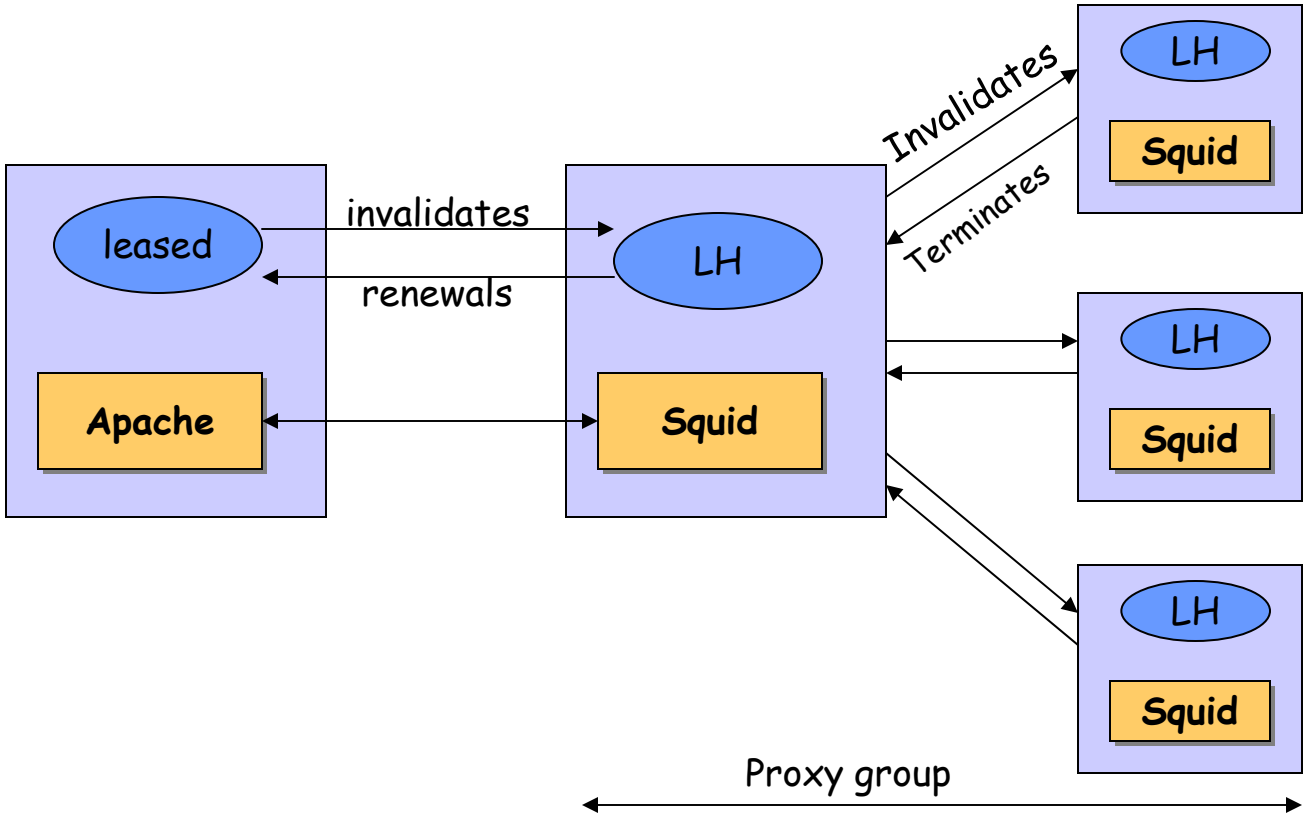
Design Considerations

- Propagating updates versus invalidates
  - Updates: more overhead, especially if no subsequent access
  - Invalidate: extra overhead upon subsequent access
  - Choose based on object characteristics
    - Send updates for popular objects, invalidates for others

- How to choose lease duration and notification rate Δ?
  - Analytical models for choosing lease duration [Infocom00]
  - Choose Δ proportional to server load
    - Stronger guarantees to low/moderate loads
    - Progressively weaker guarantees at high loads
  - Δ can also be chosen based on the object type/user preferences
    - Example: choose Δ based on object size

Prototype Implementation

- Implemented Cooperative Leases in Apache and Squid



Talk Outline

- Motivation
- Cooperative Leases: Design and Implementation
- Experimental Evaluation
- Related Work
- Concluding Remarks

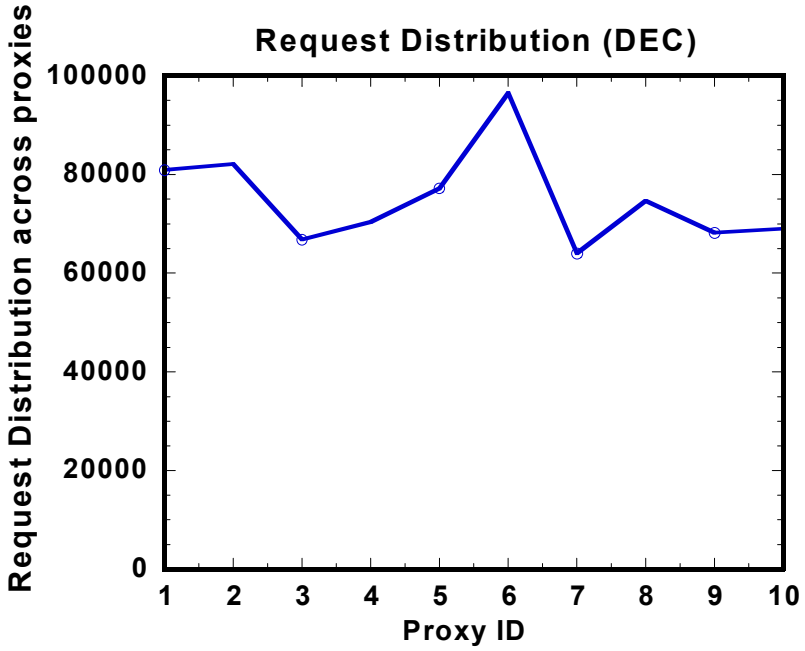
Methodology

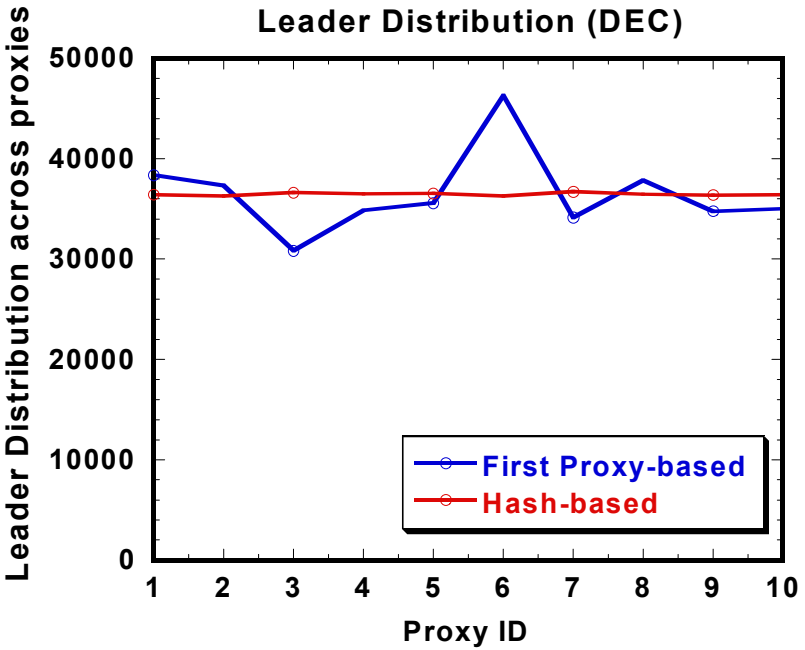
- Combination of simulation and prototype evaluation
  - Use simulations to explore design space
  - Use prototype to measure implementation overheads

- Simulations use traces from actual proxies

Trace	Requests	Objects	Writes
DEC	750K	276914	17126
NLANR	750K	393853	14385

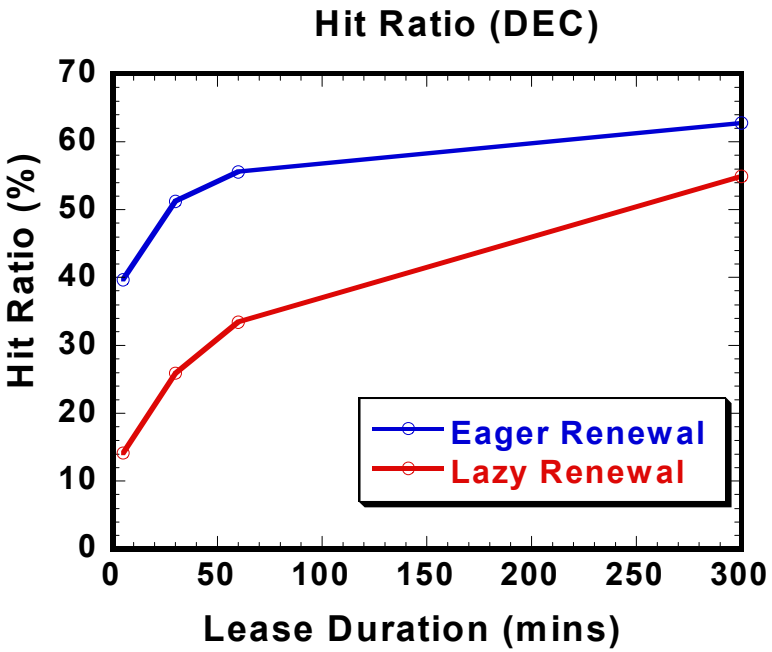
Impact of Leader Selection Policy

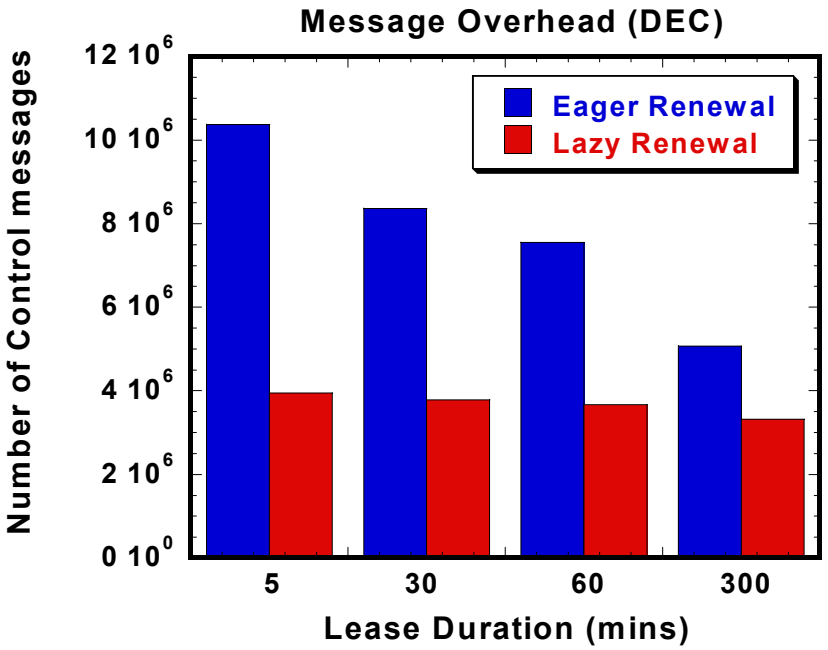




- Hash-based scheme yields better load balancing
  - Increase in communication overhead small (< 10%)
- Result: hash-based schemes preferable for leader selection

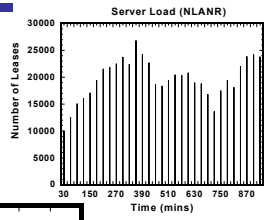
Impact of Lease Renewal

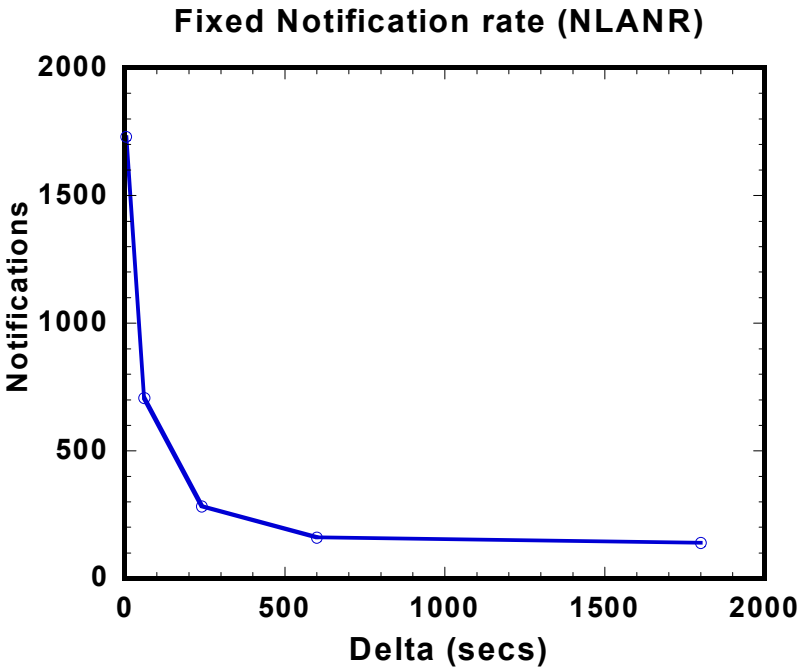


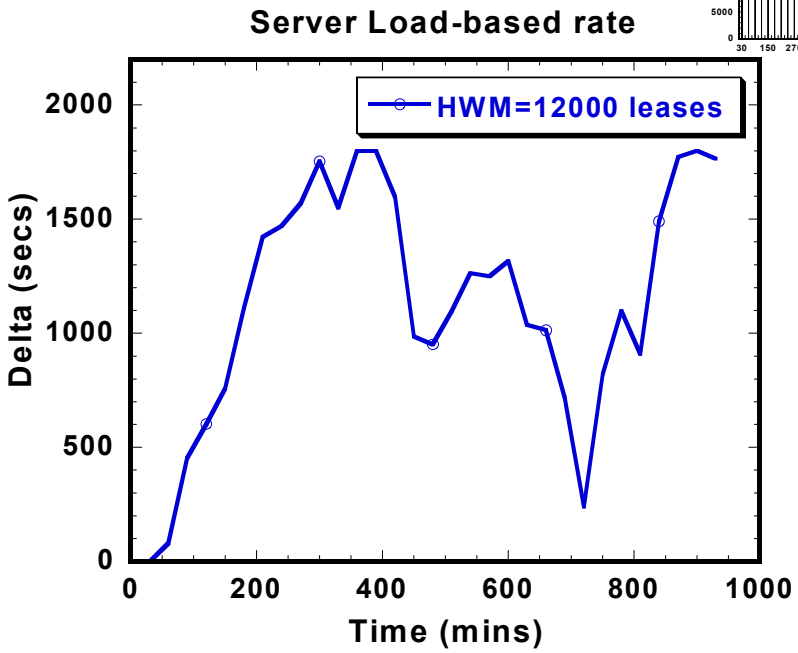


- Higher hit ratios for eager renewals
- 33-175% increase in message overhead (extra renew messages)
- Tradeoff: better hit rate/response time versus message overhead

Impact of Notification Rate

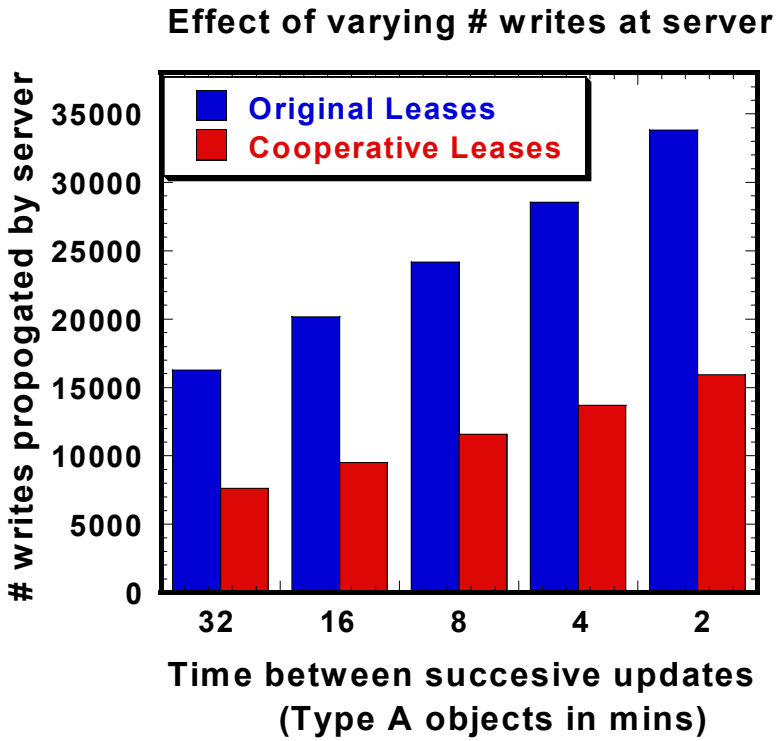






- Smaller delta: more notifications (overhead), stronger guarantees
- Choosing delta based on server load can help at heavy loads

Comparison with Original Leases



- Smaller server overhead as compared to original leases
  - Reduction in server msg overhead: 2.5 X, state space: 20-30%
  - But larger inter-proxy communication overheads (3.7 X)

Implementation Overheads

Server Overheads

Event	Time (ms)
Lease grant	0.64
Lease renew	0.28
Send invalidate	3.36

Proxy Overheads

Event	Time (ms)
Dir. broadcast	2.7
Lease renew	2.65
Send invalidate	0.565

- Implementation overheads seem reasonable

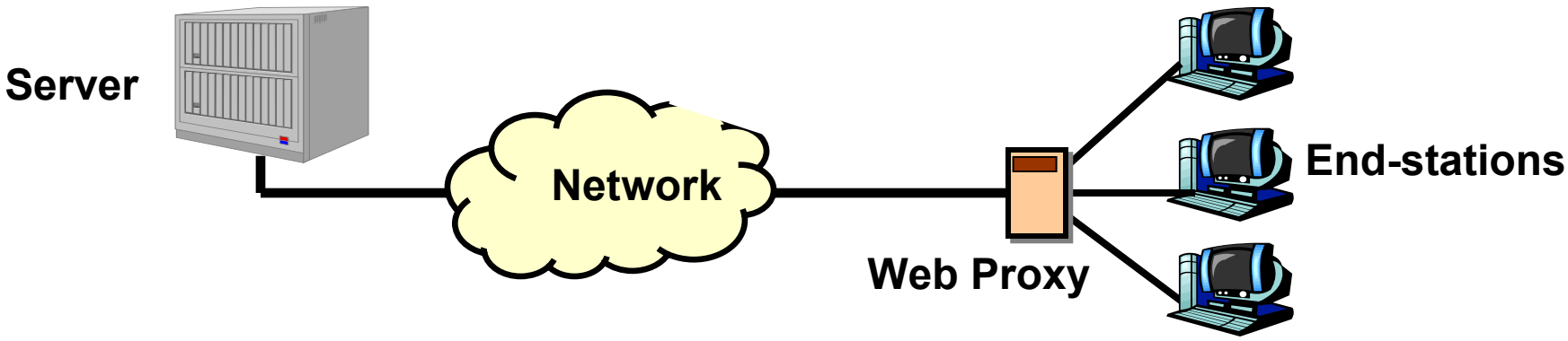
Related Work

- Volume leases: use a lease for a group of objects
- WCIP: protocol for propagating invalidates
- DOCP: distributed object consistency protocol
- Hierarchical WAN consistency [Yin99]
- Use of multicast for consistency in hierarchies [Yu99]

Concluding Remarks

- Single proxy consistency mechanisms don't scale to CDNs
- Cooperative leases: flexible, scalable consistency for CDNs
  - Use a single lease for a group of proxies
  - Application-level multicast of server notifications
  - Effectiveness demonstrated via an experimental evaluation
- More at http://lass.cs.umass.edu

Web Proxy Caching: Benefits



- Reduces end-user access latencies
  - By deploying proxies close to clients
- Reduces network bandwidth on access links
  - By caching near access links
- Reduces server load
  - By servicing requests using cached data