Distributed Servers Architecture for Networked Video Services

S.-H. Gary Chan and Fouad Tobagi

Presented by Todd Flanagan

A Little About the Authors

- Academic Pedigree with business backgrounds
- Not publishing for a university
- Tobagi – background in project management and co-founded a multimedia networking company

Overview

- Motivation
- Simplifying Assumptions
- Probability and Queuing Review
- Overview
- Previous Work
- Schemes
- Analysis
- Results and Comparisons
- Conclusions

Motivation

- What does this have to do with differentiated services?
- Local interest - EMC, Compaq, SUN, Storage Networks, Akami, and others
- Applications paper
- Not published through a university effort

Simplifying Assumptions

- A movie or video is any file with long streaming duration (> 30 min)
- Local network transmission cost is almost free
- The network is properly sized and channels are available on demand
- Latency of the central repository is low
- Network is stable, fault-recovery is part of the network and implied, and service-interruptions aren’t an issue
- Network channel and storage cost is linear

Nomenclature

TABLE II
NOMENCLATURE USED IN THIS PAPER

Symbol		Meaning
T_h	:	Movie length (minutes)
b_0	:	Streaming rate of the movies (MB/min)
α	:	Storage cost (\$/(min·MB))
β	:	Network channel cost (\$/(min·channel))
γ	:	$\triangleq b_0\alpha/\beta$, storage cost w.r.t. channel cost (channel/min, or simply, /min)
N_s	:	The number of local servers in the system
λ_i	:	Request rate for a specific movie in the local server i (req/min)
λ	:	Total request rate for a specific movie in the system (req/min)
Λ	:	Total request rate for all the movies in the system (req/min)
$\overline{B}_i$	:	Average buffer used for a movie in the local server i (minutes)
$\overline{S}$	:	Average number of repository channels used for a movie
C	:	The total cost of the distributed architecture (\$/min)
$\hat{C}$	:	Normalized total cost of the distributed architecture w.r.t. β

Probability and Queuing

- Stochastic processes
- Poisson process properties
  - Arrival rate = λ
  - Expected arrivals in time T = λT
  - Interarrival time = $1/\lambda$
  - Interarrival time obeys exponential distribution
- Little's Law
  - $q = \lambda T_q$

Overview

- On demand video system
  - Servers and near storage
  - Tertiary tape libraries and juke boxes
  - Limited by the streaming capacity of the system
- Need more streaming access in the form of more servers
- Traditional local clustered server model bound by the same high network cost
- Distributed servers architecture
  - Take advantage of locality of demand
  - Assumes much lower transmission costs to local users
  - More scalable

Overview (2)

- Storage can be leased on demand
- γ = ratio of storage cost to network
  - small γ -> relatively cheap storage
- Tradeoff network cost versus storage cost
- Movies have notion of skewness
  - High demand movies should be cached locally
  - Low demand serviced directly
  - Intermediate class should be partially cached
- Cost decision should be made continuously over time

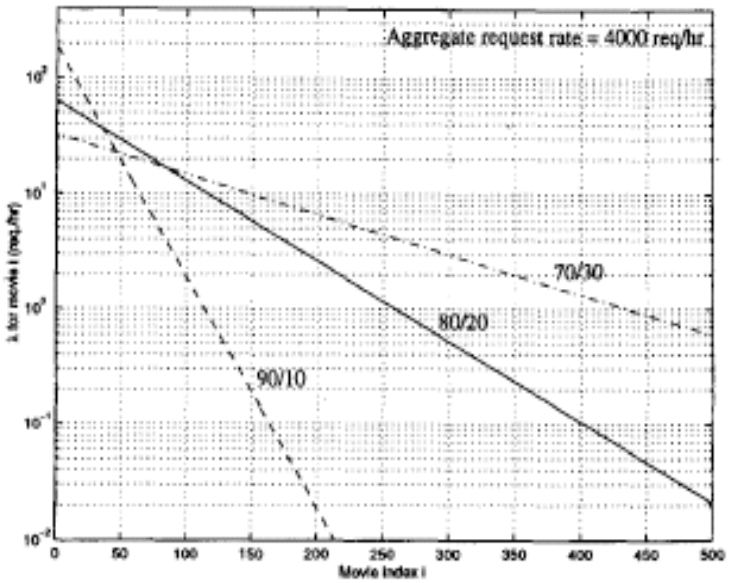


Fig. 1. λ for movie i in a video system with 500 movies and $\Lambda = 4000$ req/h (geometric video popularity).

Overview (3)

- Three models of distributed servers archetecture
  - Uncooperative – cable tv
  - Cooperative multicast – shared streaming channel
  - Cooperative exchange – campus or metropolitan network
- This paper studies a number of caching schemes, all employing circular buffers and partial caching
- All requests arriving during the cache window duration are served from the cache
- Claim that using partial caching on temporary storage can lower the system cost by an order of magnitude

Previous Work

- Most previous work studied some aspect of a VOD system, such as setup cost, delivering bursty traffic or scheduling with *a priori* knowledge
- Other work done with client buffering
- This study deals with multicasting and server caching and analyze the tradeoff between storage and network channels

Schemes

- Unicast
- Multicast
  - Two flavors
- Communicating servers

Scheme - Unicast

- Fixed buffer for each movie
- T_h minutes to stream the movie to the server
- W minute buffer at the server
- Think Tivo - buffers for commercials
- Arrivals within W form a cache group
- Buffer can be reduced by "trimming the buffer", but cost reduction is negligible

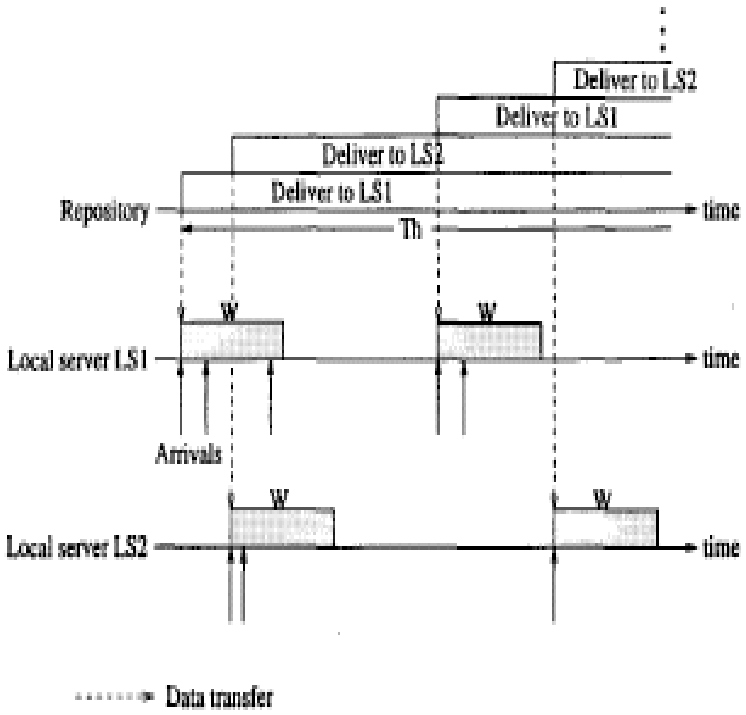


Fig. 3. Scheme for unicast delivery.

Scheme - Multicast with Prestoring

- Local server stores a leader of size W
- Periodic multicast schedule with slot interval W
- If no requests during W, next slot multicast cancelled
- Single multicast stream is used to serve multiple requests demanded at different times, only one multicast stream cost
- W=0 is a true VOD system

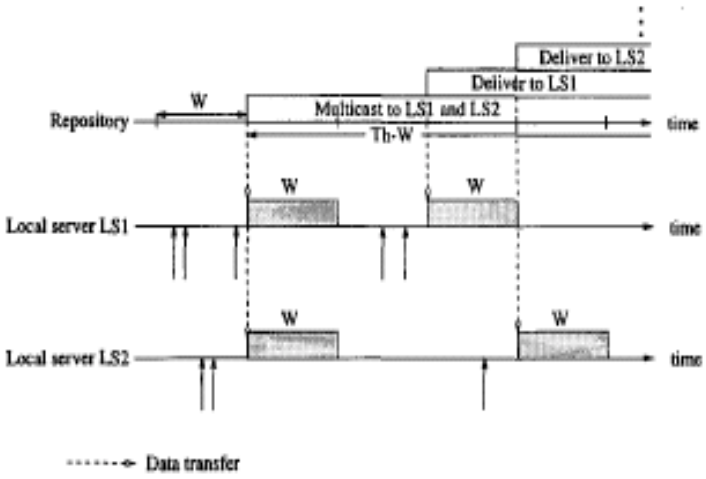


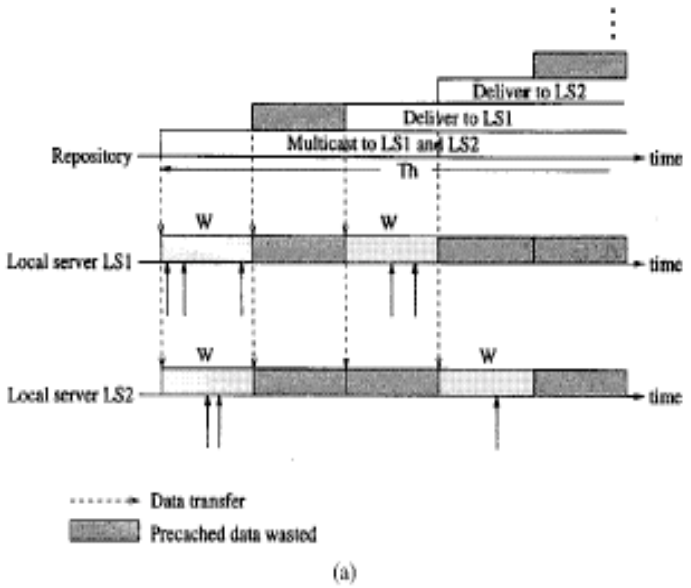
Fig. 4. Prestoring scheme for multicast delivery.

Scheme - Multicast with Precaching (1)

- No permanent storage in local servers
- Decision to cache made in advance
- If no requests, cached data is wasted
- If not cached, incoming request is VOD

Scheme - Multicast with Precaching (2)

- Periodic multicasting with precaching
- Movie multicast on interval of W min
- If request arrives, stream held for T_h mir
- Otherwise, stream terminated



Scheme - Multicast with Precaching (3)

- Request driven precaching
- Same as above, excep that multicast is initiated on receipt of first request (for all servers)
- All servers cache window of length W

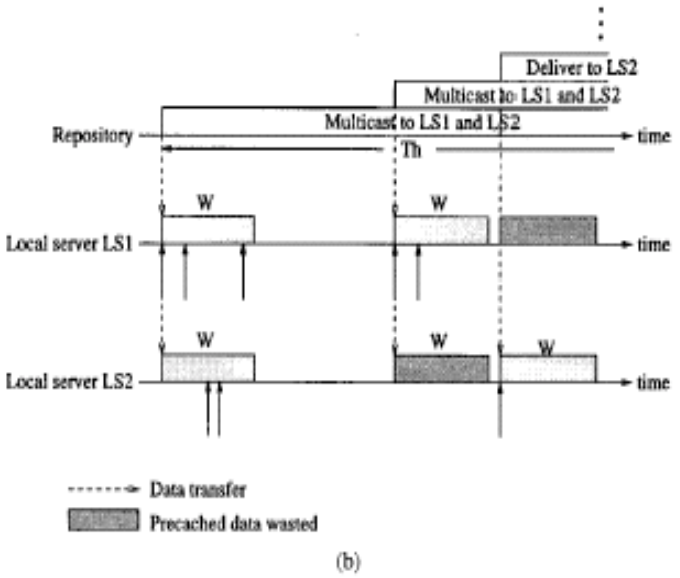


Fig. 5. Precaching schemes for multicast delivery. (a) Periodic multicasting with precaching. (b) Request-driven precaching.

Scheme - Communicating Servers

- Movie unicast to one server
- Additional local requests served from within group forming a chain
- Chain is broken when two buffer allocations are separated by more than W minutes

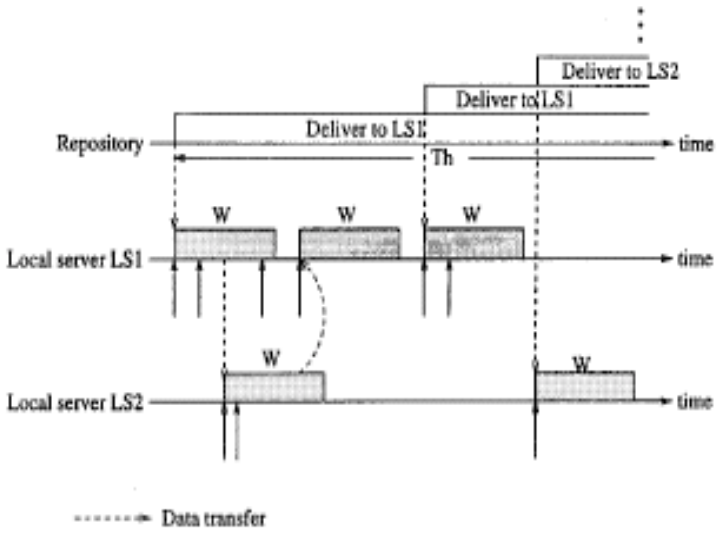


Fig. 6. Scheme for communicating servers.

Scheme Analysis

- Movie length T_h min
- Streaming rate b_0 MB/min
- Request process is Poisson
- Interested in
  - Ave number of network channels, $\overline{S}$
  - Ave buffer size, $\overline{B}$
  - Total system cost: $\hat{C} = \gamma \sum \overline{B}_i + \overline{S}$

Analysis - Unicast

- Interarrival time = W + $1/\lambda$
- By Little's Law: $\bar{S} = \frac{T_h}{W + 1/\lambda}$
- Average number of buffers allocated = $(1/(W+1/\lambda))T_h$ which yields $\bar{B}$
- Eventually: $\hat{C} = (\gamma - \lambda)\bar{B} + \lambda T_h$
- To minimize $\hat{C}$, either cache or don't
  - $\lambda < \gamma$ B = W = 0
  - $\lambda > \gamma$ B = T_h

Analysis - Multicast Delivery

- Note that Poisson arrival process drives all results
  - Determines the probability of an arrival, thus the probability that a cache action is wasted
- Big scary equations all boil down to capturing cost from storage, channel due to caching, channel cost due to non-caching
- Average buffer size falls out of probability that a buffer is wasted or not

Analysis - Communicating Servers

- Assumes that there are many local servers so that requests come to different servers
  - Allows effective chaining
- From Poisson, average concurrent requests is λT_h so average buffer size is $\lambda T_h W$
- Interarrival time based on breaking the chain
  - Good chaining means long interarrival times

Results - Unicast

- For unicast, tradeoff between S and B give λ is linear with slope ($-\lambda$)
- Optimal caching strategy is all or nothing
- Determining factors for caching a movie
  - Skewness
  - Cheapness of storage

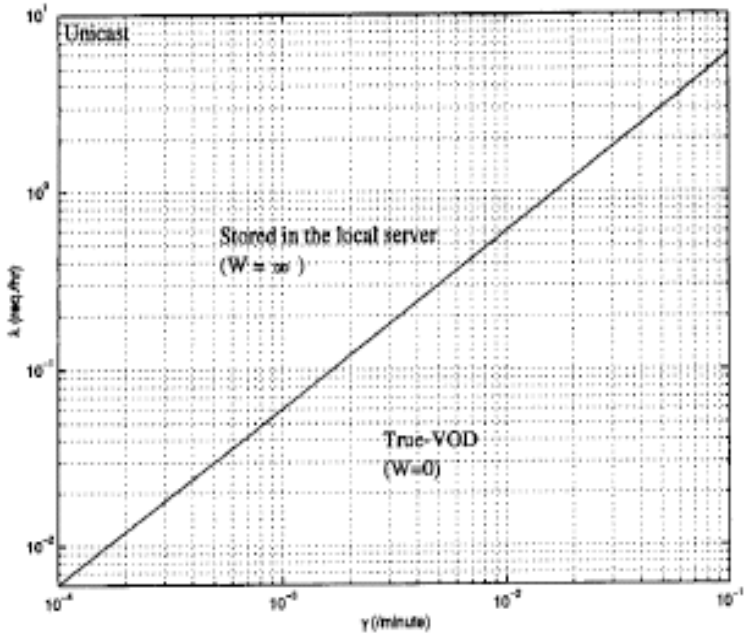


Fig. 7. Relationship between λ and γ to minimize the system cost for unicast delivery.

Results - Multicast with Prestoring

- There is an optimal W to minimize cost
- The storage component of this curve becomes steeper as γ increases

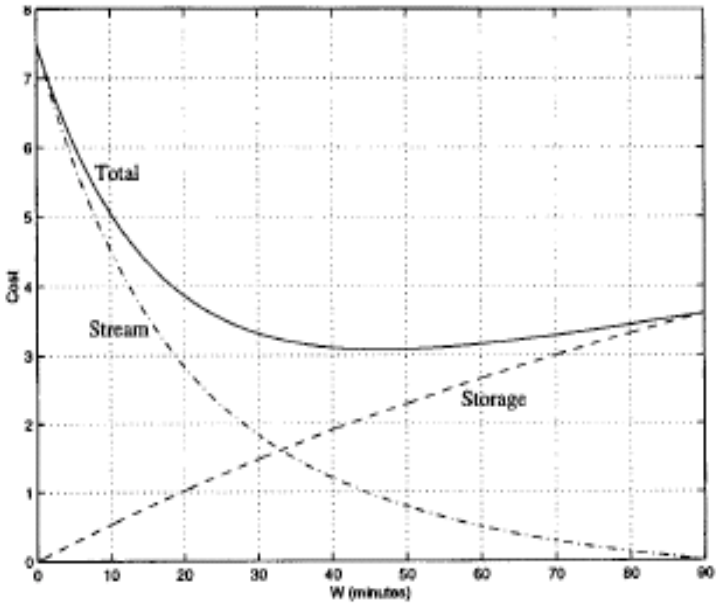


Fig. 8. $\hat{C}$, $\overline{S}$, and $\gamma\overline{B}$ versus W for prestoring ($\lambda = 5$ req/min, $N_s = 20$, $\gamma = 0.002$/min, and $T_\lambda = 90$ min).

Results - W* vs λ for Mulitcast with Prestoring

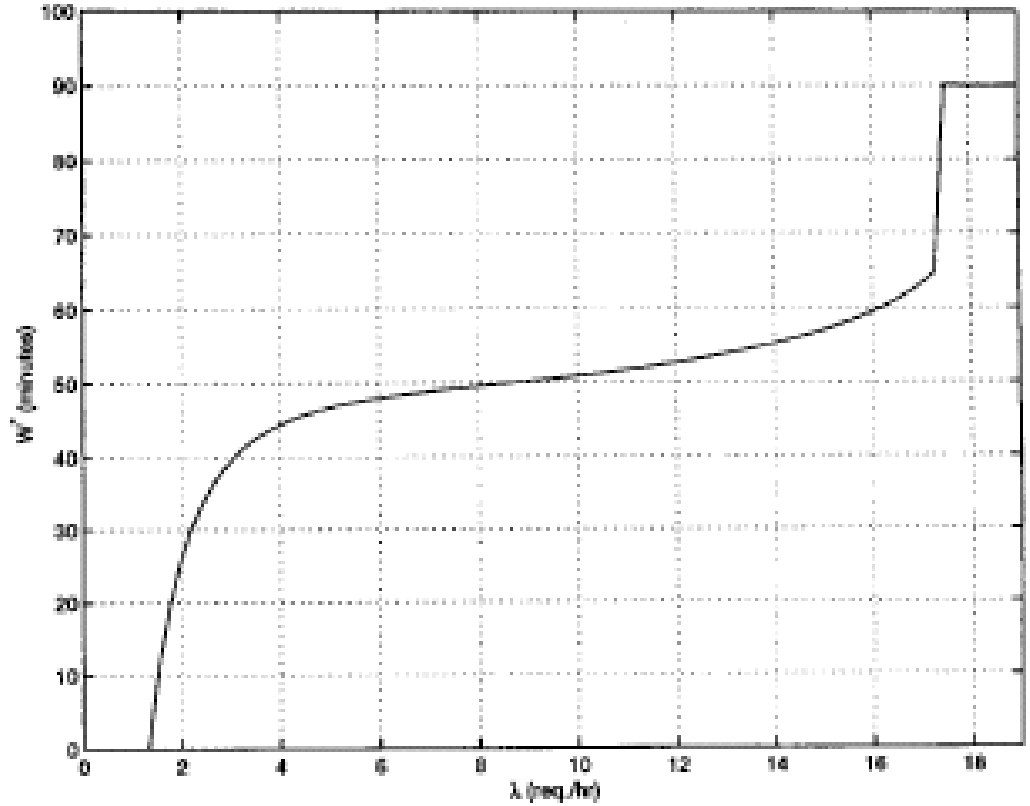


Fig. 9. W^* versus λ for prestoring ($N_s = 20$, $\gamma = 0.002$/min, and $T_\lambda = 90$ min).

Results - W^* vs λ for Multicast with Precaching

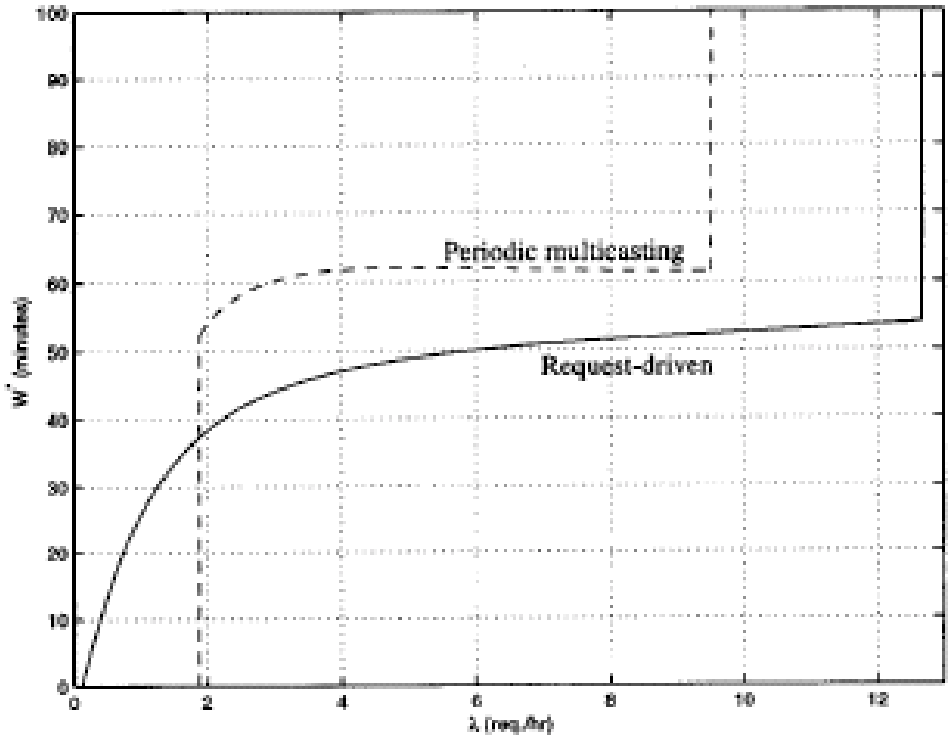


Fig. 10. W^* versus λ for precaching ($\gamma = 0.002$/min, $N_s = 20$, and $T_l = 90$ min).

Results - W^* vs λ for Chaining

- The higher the request rate, the easier it is to chain
- For simplicity, unicast and multicast channel cost are considered equal
- Assumes zero cost for inter-server communication
- Even with this assumption, chaining shouldn't be higher cost than other systems unless local communication costs are very high

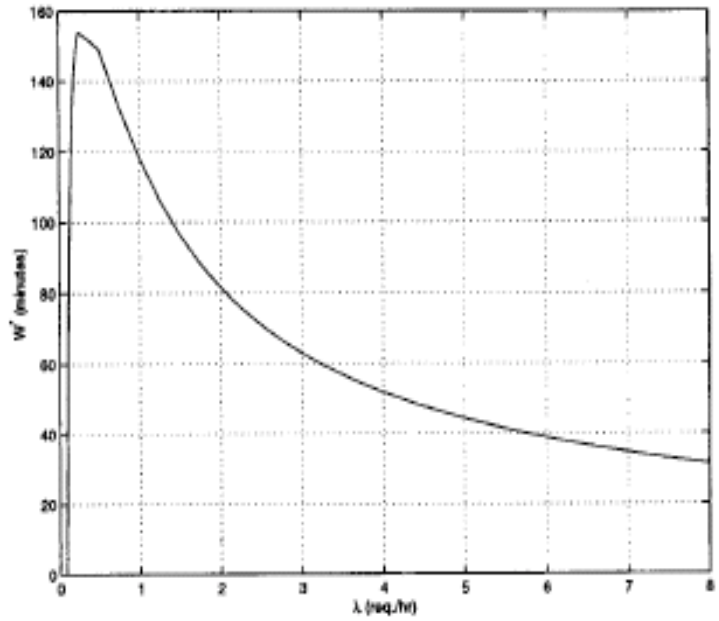


Fig. 11. W^* versus λ for chaining ($\gamma = 0.002$/min, and $T_h = 90$ min).

Comparison of C^* vs λ

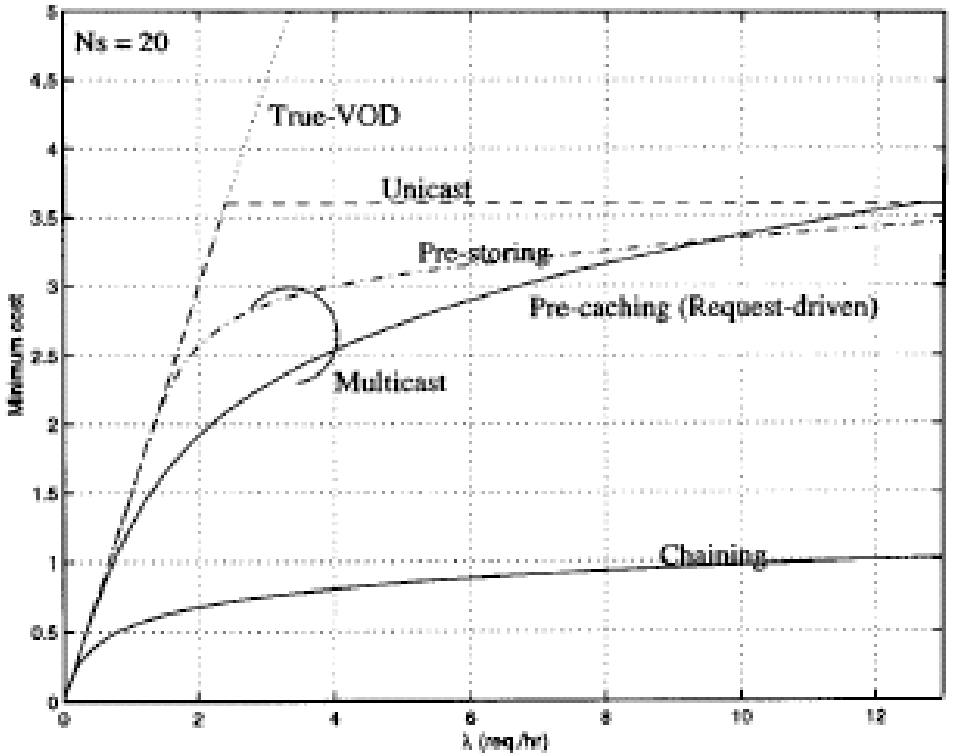


Fig. 12. Comparison of $\hat{C}^*$ versus λ for the proposed caching schemes ($\gamma = 0.002$/min, and $T_h = 90$ min).

Further Analysis - Batching and Multicasting (1)

- Assumes users will tolerate some delay
- Batching allows fewer multicast streams to be used, thus lowering the associated cost
- DS architecture can achieve lower system cost with zero delay

Further Analysis - Batching and Multicasting (2)

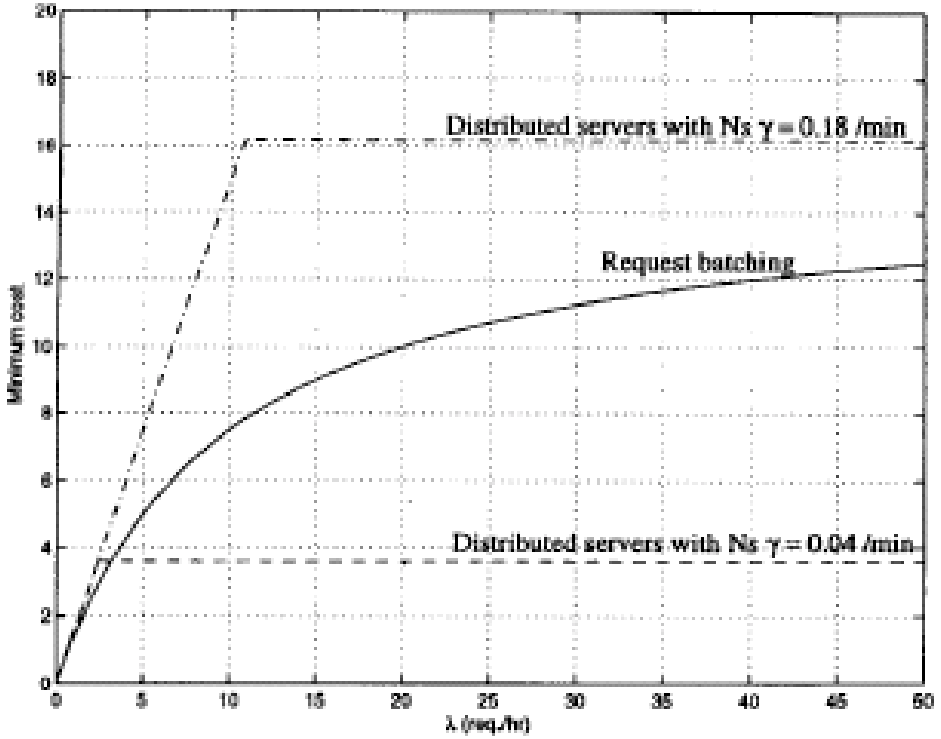


Fig. 13. Comparison of $\hat{C}^*$ versus λ between a system with request batching and a distributed servers architecture based on unicast delivery ($D_{\max} = 6$ min, and $T_h = 90$ min).

The Big Picture - Total Cost per Minute vs Λ

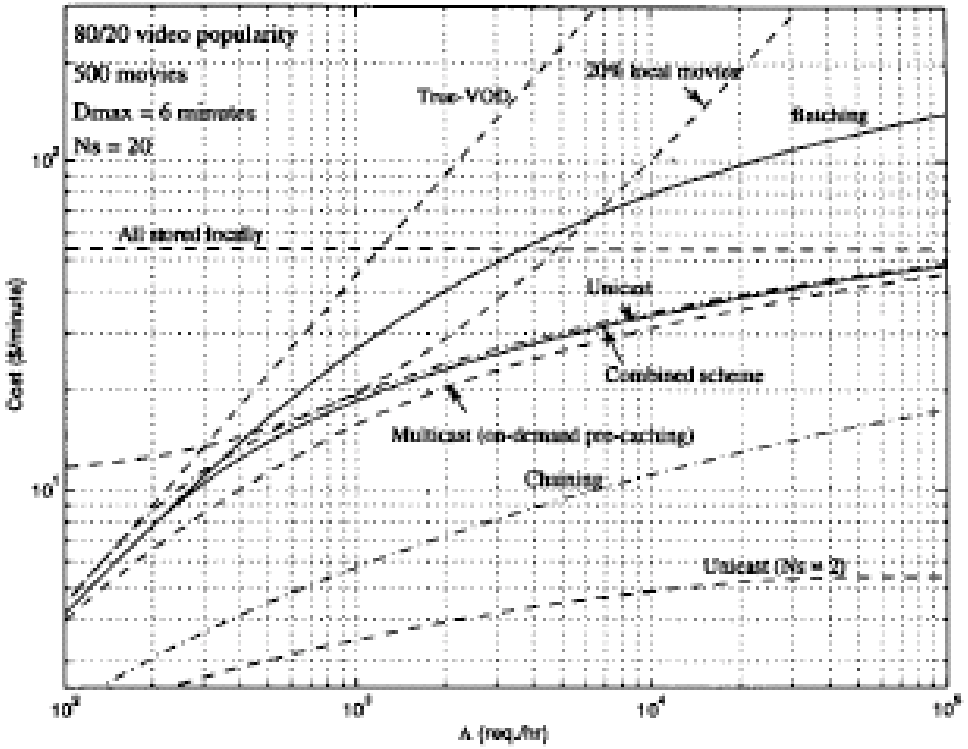


Fig. 14. Total cost per minute versus Λ for different systems ($\gamma = 0.002$/min, $\beta = \$0.03$/(min-channel), $N_s = 20$, number of movies = 500, 80/20 video popularity, $D_{\max} = 6$ min, and $T_\Lambda = 90$ min).

Conclusions

- Strengths
  - Flexible general model for analyzing cost tradeoffs
  - Solid analysis
- Weaknesses
  - Optimistic about skewness
  - Optimistic about Poisson arrival
  - Zero cost for local network