

Joint work with Sohail Hameed

Funded in part by Texas Advanced Technology Program

## **Data Broadcast in Asymmetric Wireless Environments** <sup>1</sup>

Nitin H. Vaidya

Sohail Hameed

Department of Computer Science

Texas A&M University

College Station, TX 77843-3112

E-mail: {vaidya,shameed}@cs.tamu.edu

Web: <http://www.cs.tamu.edu/faculty/vaidya/>

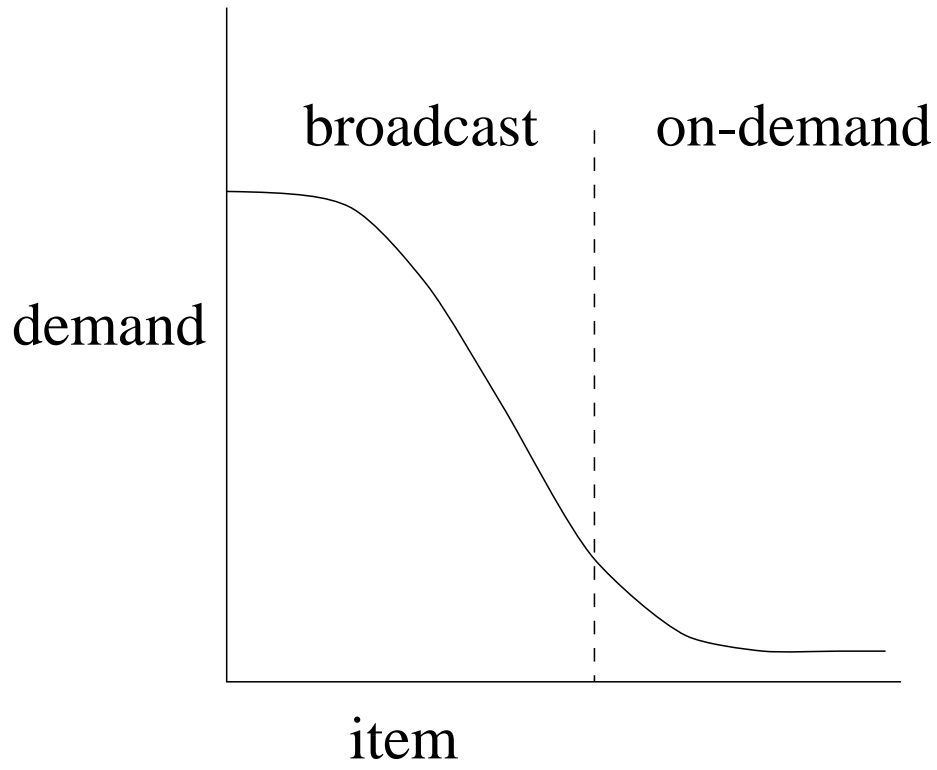
# Data Broadcast

---

- Information server needs to communicate information items to clients.
- Asymmetric environment:
  - client  $\rightarrow$  server bandwidth low
  - server  $\rightarrow$  client bandwidth high
- Broadcasting the data is effective when the environment is asymmetric.
- Problem statement: Determine when each item should be broadcast such that the “wait time” experienced by the clients is minimized. Items are broadcast periodically, frequency of each item may be dependent on parameters such as *demand* for the item.
- Limitation of this work: Caching not considered (on-going work). Caching schemes can be added to the proposed broadcast schemes relatively easily.

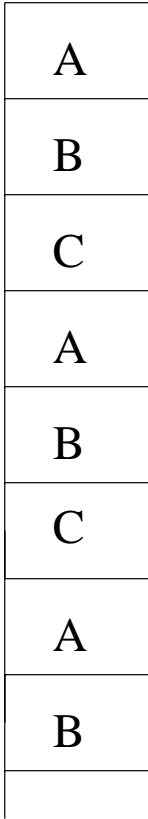
# When Broadcast?

---

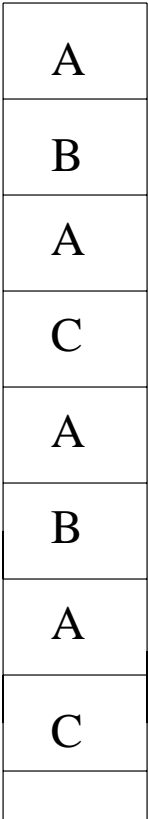


# Broadcast Scheduling

---



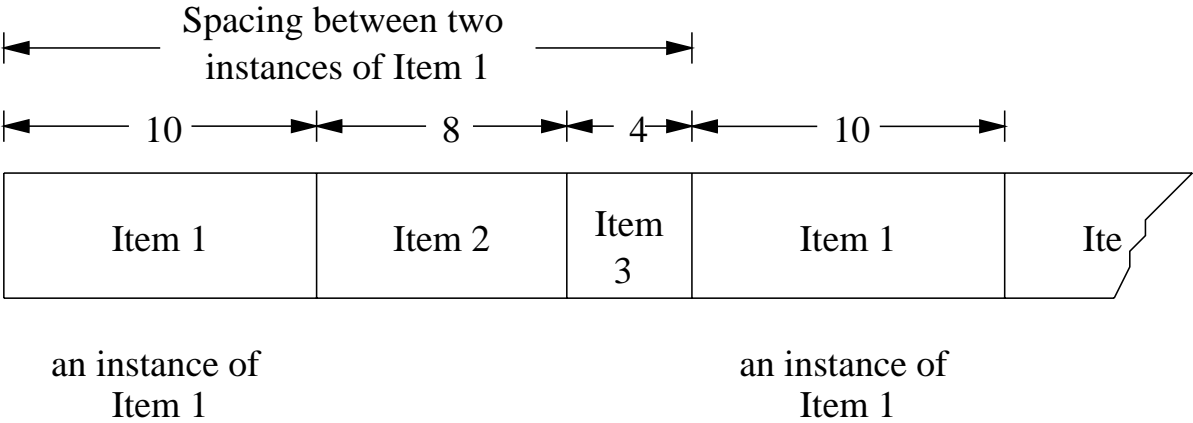
flat schedule



item A requested more often ==> transmit more often

# Data Broadcast

---



# Definitions

---

$p_i$  = demand probability for item  $i$  ( $M$  items in total)

$l_i$  = length of item  $i$

$f_i$  = frequency of item  $i$

*Item Mean Access Time*  $t_i$  for item  $i$  =  
Average wait encountered by a client waiting for item  $i$ .

If all instances of item  $i$  are spaced  $s_i$  apart, then,

$$t_i = \frac{s_i}{2}$$

*Overall Mean Access Time*  $t$  = average wait encountered by a client (averaged over all items). For equally-spaced items:

$$t = \sum_{i=1}^M \frac{s_i}{2} p_i$$

# Square-root rule

---

*Overall mean access time* is minimized when

$$f_i \propto \sqrt{\frac{p_i}{l_i}}$$

In other words:

$$s_i \propto \sqrt{\frac{l_i}{p_i}}$$

Two ways to interpret the above result:

—  $s_i^2 p_i / l_i = \text{constant}, \forall i$

—  $l_i / s_i = \phi_i$       $\phi_i$  is function of  $l_j$ 's and  $p_j$ 's

$l_i / s_i$  is bandwidth fraction allocated to item  $i$

This leads to two algorithms for broadcast scheduling.

Attempt to achieve the equality

$$\frac{s_i^2 p_i}{l_i} = \text{constant}, \quad \text{for all items } i$$

$Q$  = Current Time

$R(j)$  = Last time when item  $j$  was broadcast

$G(j)$  =  $(Q - R(j))^2 p_j / l_j$

## Algorithm A: On-line algorithm:

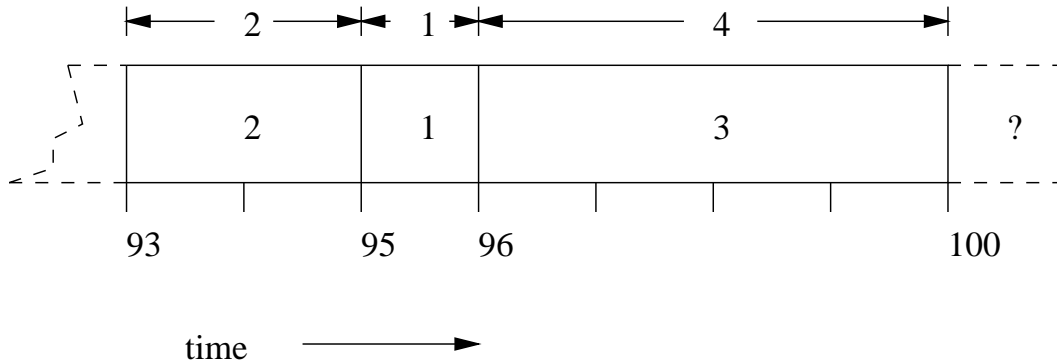
Step 1: Determine maximum value of  $G(j)$  over all items  $j$ .

Let  $G_{max}$  denote the maximum value of  $G(j)$ .

Step 2: Choose item  $i$  such that  $G(i) = G_{max}$ . If this equality holds for more than one item, choose any one of them arbitrarily.

Step 3: Broadcast item  $i$  at time  $Q$ .

Step 4:  $R(i) = Q$ .



$$p_1 = 0.5, p_2 = 3/8, p_3 = 1/8$$

$$\begin{aligned} G(1) &= (Q - R(1))^2 p_1/l_1 \\ &= (100 - 95)^2 p_1/l_1 \\ &= 12.5 \end{aligned}$$

$$\begin{aligned} G(2) &= (Q - R(2))^2 p_2/l_2 \\ &= (100 - 93)^2 p_2/l_2 \\ &= 147/16 \end{aligned}$$

$$G(3) = 0.5$$

Item 1 is transmitted at time  $Q = 100$  as  $G(1)$  is largest.

# Algorithm B

---

*Time complexity of Algo. A is linear in number of items.*

- Reduce the time complexity by partitioning items into “buckets”.
- Items in each bucket are maintained in a cyclic queue.
- Modified algorithm (algo. B) similar to algorithm A,  
BUT decision rule in step 1 applied *only* to items at front of each bucket.
- Time complexity is linear in number of buckets.

## Algorithm B

---

$m_j$  = number of items in bucket  $B_j$

$q_j$  = average demand probability the items in bucket  $B_j$

$d_j$  = average length of the items in bucket  $B_j$

Let item  $i$  be in bucket  $j$ . For optimality:

$$\text{spacing } s_i \propto \sqrt{d_j/q_j}$$

$$\text{In other words, } \frac{s_i^2 q_j}{d_j} = \text{constant, } \forall j, 1 \leq j \leq k \text{ and } i \in B_j$$

Expression for optimal *overall mean access time* :

$$\frac{1}{2} \left( \sum_{j=1}^k m_j \sqrt{q_j d_j} \right)^2$$

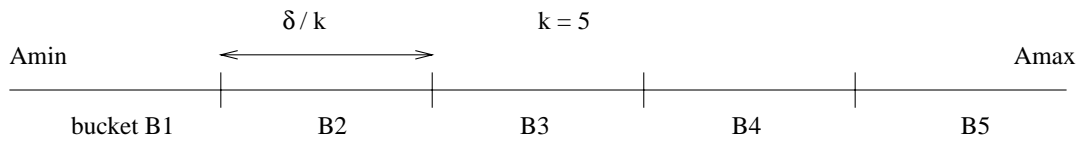
This is also a *lower bound*.

# Bucket Assignment

---

Assignment of items to buckets affects performance of the bucketing scheme.

Our heuristic:



The range of  $\sqrt{p_i/l_i}$  values for the items is divided into  $k$  equal parts.

Items in the  $i$ -th part go into bucket  $B_i$ .

# Transmission Errors

---

1	2	ERRORS 3	4	5	ERRORS 3	2	No errors 3	5
---	---	-------------	---	---	-------------	---	----------------	---



A client cannot use an item containing uncorrectable errors.

Larger items tend to contain more errors.

How to schedule in presence of errors ?

# Transmission Errors

---

$E(l)$  = probability that uncorrectable errors occur in an item of length  $l$

Optimal *Overall mean access time* is :

$$t = \frac{1}{2} \sum_{i=1}^M s_i p_i \left( \frac{1 + E(l_i)}{1 - E(l_i)} \right)$$

*Overall mean access time* is minimized when

$$f_i \propto \sqrt{\frac{p_i}{l_i}} \left( \frac{1 + E(l_i)}{1 - E(l_i)} \right)^{1/2}$$

and

$$s_i \propto \sqrt{\frac{l_i}{p_i}} \left( \frac{1 - E(l_i)}{1 + E(l_i)} \right)^{1/2}$$

Optimal *overall mean access time* in the presence of errors :

$$t_{opt\_error} = \frac{1}{2} \left( \sum_{i=1}^M \sqrt{p_i l_i} \left( \frac{1 + E(l_i)}{1 - E(l_i)} \right)^{1/2} \right)^2$$

Also,

$$\frac{s_i^2 p_i}{l_i} \left( \frac{1 + E(l_i)}{1 - E(l_i)} \right) = \text{constant}, \quad 1 \leq i \leq M$$

# Multiple Channels

---

channel	
1	2
A	A
B	B
C	C
A	A
B	B
C	C
A	A
B	B

channel	
1	2
A	
B	A
A	B
C	A
A	C
B	A
A	B
C	A
	C

# Multiple Channels

---

Server broadcasts simultaneously on  $c$  channels.

Different clients listen to different number of channels.

$H = \{1, 2, \dots, c\}$  = the set of all broadcast channels

A client may listen to any non-empty subset  $S$  of the set  $H$ .

$\Pi_S$  : the probability that a client listens to set  $S$ .

*Overall mean access time* should now be averaged over all clients.

On-line scheduling algorithm to minimize this access time is presented in a recent technical report (Nov. 1996).

# Multiple Channels

---

Define

$$G_h(j) = \frac{p_j}{l_j} \left( \sum_{S \subseteq H, h \in S} \Pi_S (Q - R^S(j))^2 \right)$$

**On-line algorithm for channel  $h$ ,  $1 \leq h \leq c$  :**

Step 1:  $R^S(j) = \max_{h \in S} R_h(j)$ ,  $\forall S, \forall j, S \subseteq H, 1 \leq j \leq M$ .

Step 2: Determine maximum  $G_h(j)$  over all items  $j, 1 \leq j \leq M$ .

Let  $G_{max}$  denote the maximum value of  $G_h(j)$  over all  $j$ .

Step 3: Choose item  $i$  such that  $G_h(i) = G_{max}$ . If this equality holds for more than one item, choose any one of them arbitrarily.

Step 4: Broadcast item  $i$  on channel  $h$  at time  $Q$ .

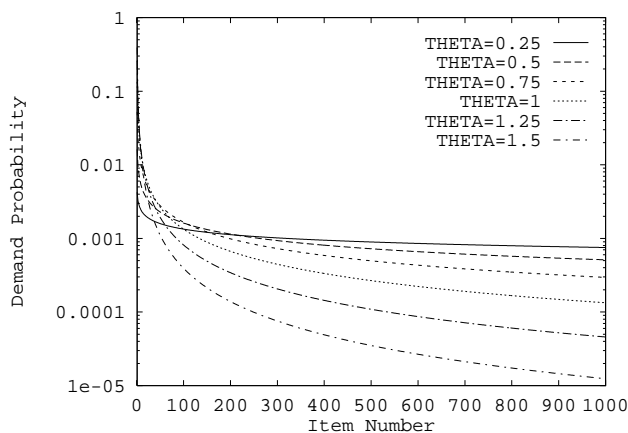
Step 5:  $R_h(i) = Q$ .

**Demand Probability Distribution:** Zipf Distribution :

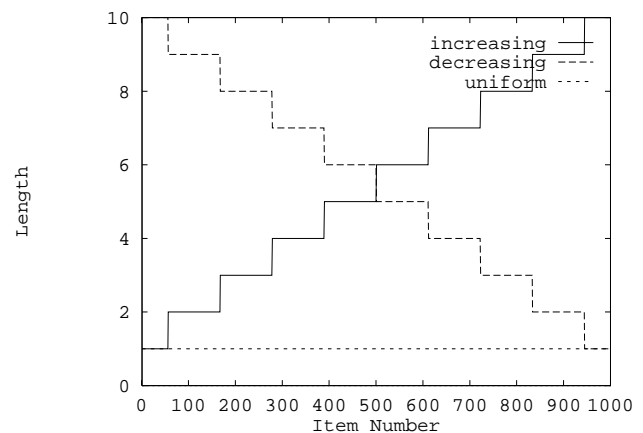
$$p_i = \frac{(1/i)^\theta}{\sum_{i=1}^M (1/i)^\theta} \quad 1 \leq i \leq M$$

**Length Distributions:**

- *Uniform Length Distribution:*  $l_i = 1$ .
- *Increasing Length Distribution:*  $1 \leq l_i \leq 10$ .
- *Decreasing Length Distribution:*  $1 \leq l_i \leq 10$ .
- *Random Length Distribution:* Lengths randomly distributed from 1 to 10 with uniform probability.

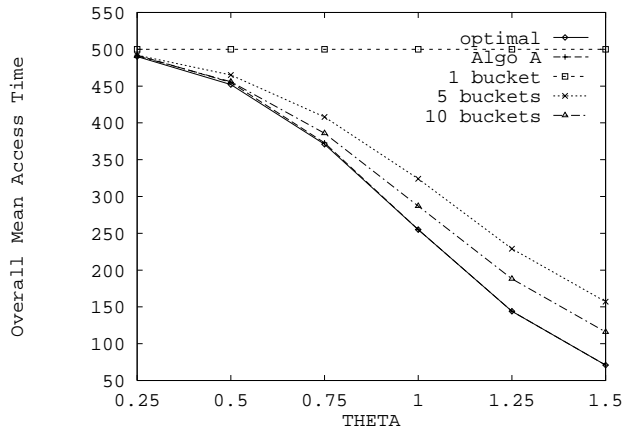


(a) Zipf distributions

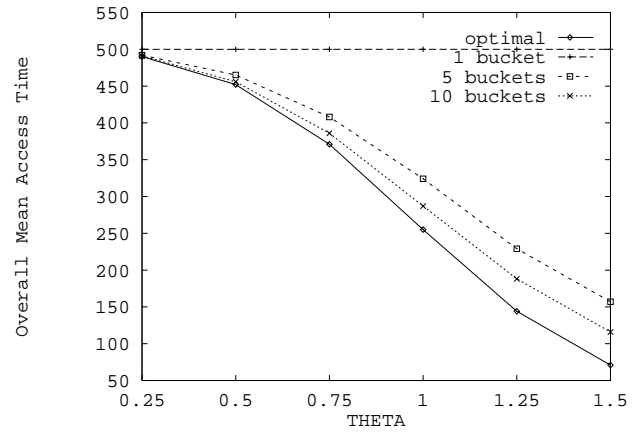


(b) Length distributions

## With Uniform Length Distribution

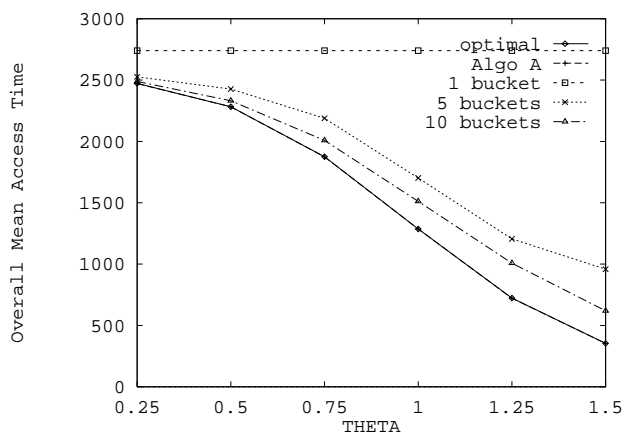


(a) Simulation results

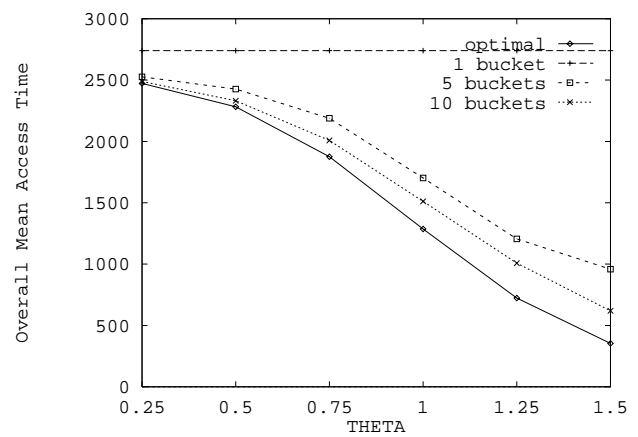


(b) Analytical lower bounds

## With Random Length Distribution



(a) Simulation results

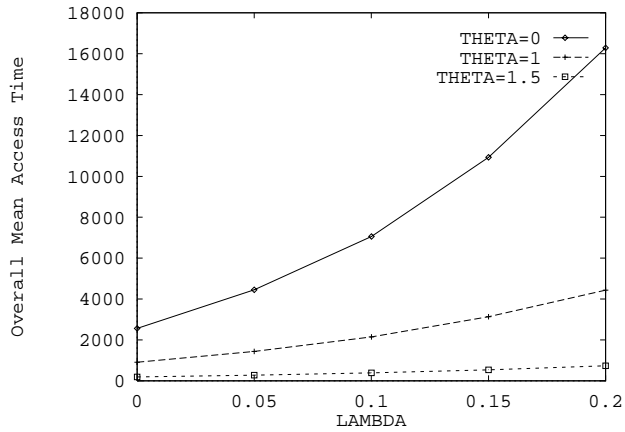


(b) Analytical lower bounds

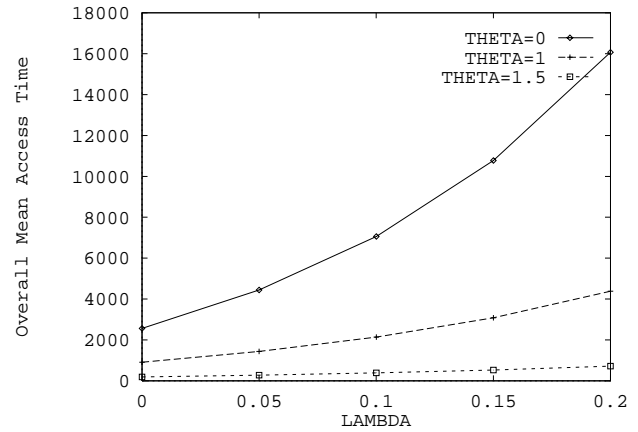
# Performance

With errors:  $E(l) = 1 - e^{-\lambda l}$

With Increasing Length Distribution

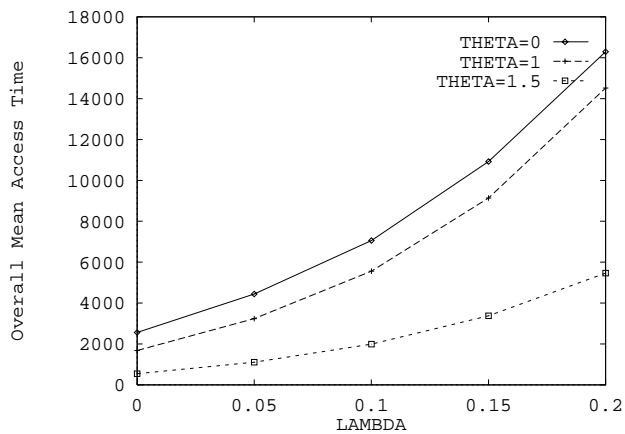


(a) Simulation results

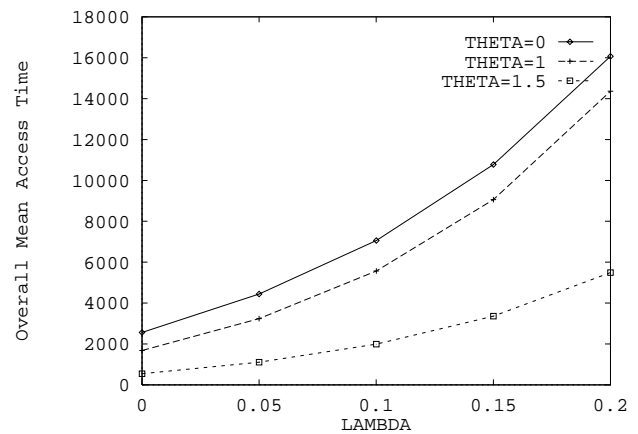


(b) Analytical lower bounds

With Decreasing Length Distribution



(a) Simulation results

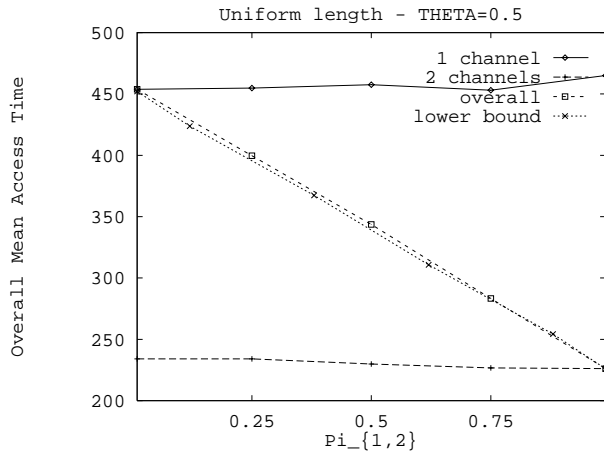


(b) Analytical lower bounds

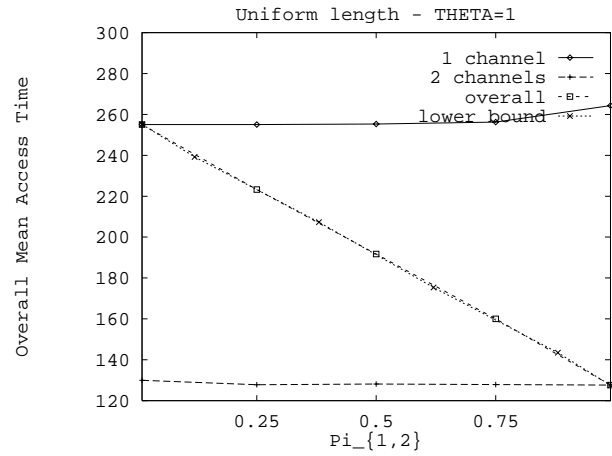
# Performance

$$2 \text{ channels: } \Pi_{\{1\}} = \Pi_{\{2\}} = \frac{1 - \Pi_{\{1,2\}}}{2}$$

With Uniform Length Distribution

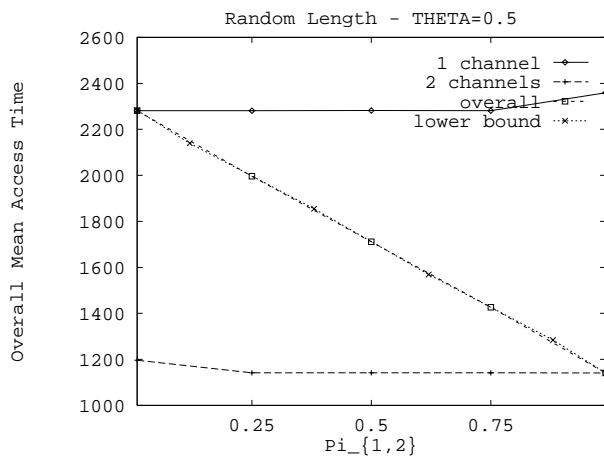


(a)  $\theta = 0.5$

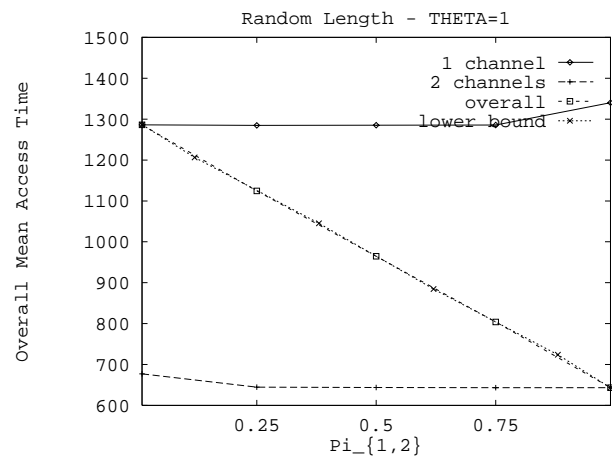


(b)  $\theta = 1$

With Random Length Distribution



(a)  $\theta = 0.5$



(b)  $\theta = 1$

## Summary

---

- On-line broadcast scheduling algorithms to minimize the *access time*.
- Bucketing scheme facilitates trade-off between time-complexity and performance.
- Impact of errors on optimal broadcast schedules.
- Algorithm for multichannel broadcast scheduling.

Future work: Simple/efficient caching algorithms, and theoretical bounds on performance with caching.

A revised version of this paper (including the new multichannel algorithm) is available as technical report 96-022 (Nov.1996) from

<http://www.cs.tamu.edu/faculty/vaidya/mobile.html>