Data Dissemination to a Large Mobile Network: Simulation of Broadcast Clouds

Aslihan Celik, JoAnne Holliday, Zachary Hurst
Santa Clara University
acelik@scu.edu, jholliday@scu.edu, zackhurst@gmail.com

Abstract

Research in data broadcasting in a wireless network has addressed the problems of what to broadcast and how to schedule the broadcasts. However, it is still not clear whether a single broadcast for the entire network, or an individual broadcast for each wireless cell is should be prepared. Recently, we proposed the Broadcast Clouds (BC) technique that bunches together wireless cells based on a cost saving principle, and prepares a common broadcast for these. This would presumably reduce the bandwidth cost and increase the service provider's profit. In this paper, we describe our simulation software, BC-SIM, that we use to evaluate the performance of the BC approach, and present the results.

1. Introduction

The widespread adoption of wireless data services is hampered by the relatively limited and costly wireless bandwidth. To serve more clients with less bandwidth, broadcasting techniques were proposed. These techniques prepare a *data broadcast* by appending individual information items together, and send it to a common channel for the mobile clients to download. Each mobile client filters the downloaded broadcast for the information that they are interested in. This solution is efficient since it eliminates the need to send the same item multiple times when each item is used by many clients.

Surprisingly, data broadcasting has not been readily adopted by service providers. We have found the following shortcomings in existing broadcasting approaches that could explain this lack of interest:

1. *Existing approaches [1, 2, 5, 6, 7, 9, 10] do not distinguish between individual cells and are very unspecific about how they work in a multi-cell environment.* Some of them disseminate the same broadcast to the entire wireless network. We call this approach ***one-for-all (1FA)***. Some protocols may be designed to suit the needs of a single cell, thus preparing and disseminating an individual broadcast for each cell. We call such approaches ***one-for-each (1FE)***.

2. *In a one-for-all broadcast, it is highly likely that a client is interested in only a small subset of the items in the broadcast*. This effect is compounded when certain data items have "locality", i.e., they are more likely to be requested at or around specific geographic locations. Hence, with a broadcast protocol that does not consider locality of data items, many clients will have to filter through numerous data items.

3. *Existing approaches [1, 2, 5, 6, 7, 9, 10] are limited in scope to the client-side of the network.* Even though the client-side quality of service metrics, such as access time and tuning time, are important, the network provider will also prefer the data broadcasting technique that minimizes the use of the network resources. Such a resource is the wireless bandwidth, thus a bandwidth minimizing protocol is desired.

4. *One-for-each techniques do not fully consider the mobility needs of a client when preparing the broadcasts.* One approach [4] frees up the broadcast space by excluding the items for the mobile hosts that have left the cell. Normally, these mobile units restart the process of requesting their items at the new cells, and have to wait until their items are included in the following broadcast intended for that cell only. A preemptive technique that would include these items in the new cell *before* the client requests them would reduce the need to resubscribe and save wireless bandwidth.

Therefore, in order for service providers to adopt data broadcasting, it is important to develop methods that reduce the wireless bandwidth cost in the face of client mobility and data item locality. The *Broadcast Clouds (BC)* approach [3] was proposed with the goal of minimizing the bandwidth cost. *BC* is a hybrid of one-for-each and one-for-all. It groups a set of neighboring cells into clusters based on the locality of the data items, and sends the cells in a cluster the same broadcast. Note that many wireless cells are very small and may not differentiate much in terms of data locality. This is particularly true for adjacent cells. BC groups these adjacent cells and prepares a common broadcast for the group. Such a group of cells is called a *cloud*. The number of cells in a cloud is a function of the probability distribution of the demand for the data items, the number of clients in each cell, and the movement pattern in and out of each cell. This way, the clients avoid resubscribing to the broadcast every time they switch cells, and the size of the broadcasts would reduce since the broadcast will be specifically prepared for that cluster. BC uses a heuristic to form these clusters based on a cost saving principle. BC is an off-line scheduling approach combined with push-based broadcasting, and also uses point-to-point delivery. However, although the BC approach is promising, and intuitively legitimate; it has not yet been validated under realistic simulations. In this paper we bridge this gap, and simulate BC with a goal of establishing the conditions where BC outperforms 1FA and 1FE.

In Section 2, we outline the cost derivation for the BC approach. In Section 3, we highlight a heuristic to form the clusters. In Section 4, we describe the BC-SIM simulation software and show preliminary results on the efficacy of BC. Section 5 concludes the paper.

2. Cost of Broadcast Clouds (BC)

Let $C(I)$ denote the cost of bandwidth for operating a cloud I in the broadcast clouds model. $C(I)$ is measured in monetary terms. Therefore,

$C(I) = (B_I + A_I + H_I)$ where

B_I: cost of fixed broadcasts,

A_I: cost of sending additional items not included in the broadcast, i.e., unicast cost,

H_I: cost of handling subscriptions that result from clients that change cells in the cloud *i*.

Cost of sending broadcasts, B_I, is straightforward: A cloud uses network bandwidth equal to the summation of broadcast sizes in the wireless cells. The broadcast size across the cloud is constant and the same. Therefore, the total cost of sending broadcasts is simply broadcast size times unit cost of wireless bandwidth times number of cells in the cloud. Note that the broadcast cost in a wireless network is *always* the same regardless of any groupings of the cells into clouds. A base station is in charge of sending the broadcast to its cell whether or not that cell is part of a cloud. For example, if a cloud consists of five cells, there will be five, albeit the same, broadcasts.

The cost of handling subscriptions, H_i, is a function of the number of clients in the cloud and the frequency of moving in and out of the clouds. This includes hand-off cost for clients moving from cell to cell. This message traffic must be secure and reliable, i.e. a set of acknowledgement messages must also be sent.

A_I, the cost of unicasting at a cost C_{uni} per Byte is incurred for the items that do not fit in the broadcast. The item is unicast for each of its subscriber.

3. Constructing the Clouds

Recall that a cloud is a collection of adjacent wireless cells that receive the same broadcast. The operation *Merge* $\widehat{M}(I \bullet J) \rightarrow \widehat{IJ}$ merges two individual cells or clouds I and J into a single cloud $\widehat{IJ}$. The cells in this new cloud receive the same broadcast. The following *Merging Rule* states that two clouds should be merged if it is more expensive to operate them individually than in the merged mode.

Merging Rule. Given two neighboring clouds, I and J, merge them if $C(I) + C(J) > C(\widehat{IJ})$ where $C(\widehat{IJ})$ is the cost for the merged clouds.

The cost of sending the broadcasts is the same before and after merging the clouds since two broadcasts must reach the two clouds in exactly the same way, and the size of the broadcast is fixed. The other cost components $A_{\widehat{IJ}}$ and $H_{\widehat{IJ}}$ vary according to the network parameters. $c_{\widehat{IJ}}$ is the number of cells in the new cloud.

Let $NumSubs_k(I)$ denote the number of subscribers for item a in cloud I, and F denote the fixed broadcast size. N is the number of items available (size of the database).

Since movements within a cloud are cost-free, the cost components are as follows:

$$A_{\widehat{IJ}} = C_{uni} \times \sum_{k = F+1}^{N} NumSubs_k(\widehat{IJ}) \qquad (1)$$

$$H_{\widehat{IJ}} = H_I + H_J - (H_{I,J} + H_{J,I}) \qquad (2)$$

Lemma 1 . For two neighboring clouds, I and J, merge the clouds if $H_{I,J} + H_{J,I} > A_{\widehat{I,J}} - (A_I + A_J)$

Proof. See [3]

The lemma states that two clouds should be merged if handling hand-offs and subscriptions is more expensive than operating the additional broadcasting modes.

We can now utilize the merging rule to decide whether two clouds I and J should be merged. The MERGE-CELL heuristic [3] starts at the center of the network, and moves in a spiral fashion. At each iteration, it considers a new cell, and decides whether to add that cell to an existing cloud or form a new cloud. The decision is based on Lemma 1.

4. Performance Analysis with BC-SIM

To analyze the performance of the Merge-Cell heuristic, we simulated the one-for-each (1FE), the one-for-all (1FA), and the Broadcast Clouds (BC) techniques in a Wireless network. 1FE is the extreme where there is a cloud for each single node, and 1FA is the other extreme with one cloud for the entire network.

The simulation code is written in Visual C++, and it simulates a scenario with numerous wireless cells and subscribers who move in and out of cells. We assume a steady-state distribution of clients, but due to mobility, the number of clients in each cell may be different.

A client's move frequency determines how often it will receive a move event. Each client has a resubscription delay that determines how long a client will wait after entering a new cell before resubscribing to a new set of data items.

Parameter	**Description**	**Baseline**
NumClients/cell	Number of clients per cell	400-2000
Fast Clients	Clients with high move frequency (1 move/min), as percentage of total clients	50%,
Slow Clients	Low move frequency (0.2 move/min), as percentage of total clients	50%
Resubscribe Delay	When a client resubscribes to items after entering a new cell	0.3 min for Fast; 0.1 min for Slow
HM LM	Network mobility (High or Low Mobility): Determined by move probability of clients	90% (HM: High mobility)
NumCells	Number of cells	400
NumItems	Number of available items	10,000
F	Fixed broadcast size (items)	50
SL	Network with some local items. Denotes the number of items with locality	100
NL	Network with no local items	(not in baseline)
NL_NP Item	Number of not local, not popular items (0.1% probability of subscription)	8910
NL_P Item	Not local, popular items (1% probability of subscription)	990
C_B	Cost of broadcasting in the regular broadcast channel	\$C_B/bit
C_S	Cost of wireless channel used for subscription	\$2C_B/bit
C_uni	Cost of unicasting	\$0.5C_B/bit
Subscription coefficient	Size of subscription packets per-item	16B
z M P	Zipf exponent (M: medium, P: peaked): determines how many cells around an item center have subscribers for that item	1 (Medium)
Bcast_rate	Broadcast rate	19.2KB/s
itemsize	Size of a data item	1KB
sim duration	Duration simulation is run	60 minutes

TABLE 1. Baseline Parameters

A data item may have a different probability of being subscribed to in each cell. This is determined by whether or not an item has locality. If an item does not have locality, then its subscription probability is the same in every cell. An item with locality has a center cell where the probability of this item being subscribed to is the highest. This probability decreases the farther one moves away from the center. The probability P_i of a client in cell i subscribing to an item with locality is governed by the following formula:

$$P_i = \frac{P_c}{(d+1)^z} \tag{3}$$

where P_c is the probability in the item's center cell, d is the distance from cell c to cell i, and z is an integer called the Zipf exponent. The larger z is, the faster the probability will decrease as the distance increases. For each data item available in the client's current cell, the client determines whether it will subscribe to the item based on the item's subscription probability in that cell.

When a client enters a new cell, after a delay, it changes its subscription set. Once the client's new set of data items has been chosen, the new set is compared with the old set. Any new items that the client did not subscribe to before require a subscription packet to be sent to the network. This uplink packet sent to the network is sent on the subscription channel. The network must then send a packet to the client containing the encryption key for that data item. This downlink packet sent from the network is sent on the unicast channel.

Unicast channel is cheaper than the broadcast channel. *C_B, C_uni* and *C_S* denote the cost of the broadcast, unicast and subscription channels for delivering one Byte of data, respectively. The unicast and subscription channels are assumed to have infinite bandwidth. Therefore, all unicast items can be sent in parallel

4.1.Baseline Results

The simulation parameters and their baseline values are shown in Table 1. The simulations are run with the same random data item distributions for all three techniques. The results of the simulation with the baseline parameters are presented in Figure 1. Note that there are

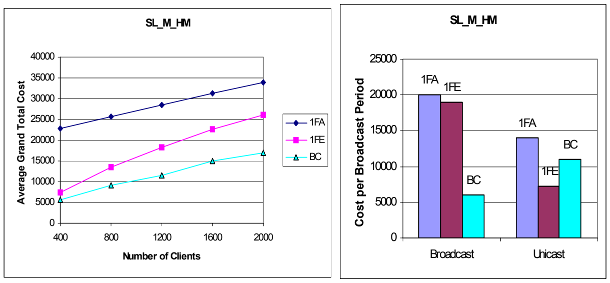

FIGURE 1. Baseline Simulation Results

10,000 items, but only 100 of the items are local and dispersed in the network. The rest of the items do not exhibit locality. The subscription process is governed by popularity: 100 of these items are popular (1% subscription probability), the remaining 8910 are not popular (0.1% subscription probability). Although the probabilities may seem low, this makes about 5-20 clients for popular, and 10-40 clients for unpopular items, respectively. In 60 minutes, this network produces 1382 broadcasts on average. The results for 1FA are much higher than the rest since it sends the same broadcast to *all* cells, regardless of whether there are clients in it or not. This is the only way to make sure that clients find an item in the broadcast when they enter a new cell. At 400 clients, 1FE and BC have about the same cost, but as the number of clients is increased the performance of BC improves. Figure 1 also shows the breakdown of the cost components for simulation runs with 2000 clients. Here, we see an interesting trade-off between 1FE and BC: at the given cost values, BC is more efficient in broadcasting, but since it compensates for lack of space in broadcasts with unicasting, it spends more than 1FE for unicasts. If the unicast cost were much higher (or broadcast cost lower), then 1FE would be preferable to BC.

The baseline parameters are varied to see the sensitivity of the results to specific parameters. We changed the mobility rate, item distribution, and broadcast size to obtain the results discussed in the following sections. We also discuss the subscription cost breakdown at the end.

4.2.Network with Low Mobility.

We repeated the simulations with move probability of 33% (LM). The results, as shown in Figure 2, are almost identical to the High Mobility (HM) case. The only fluctuations are due to subscription costs. Note that we do expect that when move probability is zero, the BC constructs a single cloud with all the cells, thus overlapping with the 1FA approach. The figure also shows the cost breakdown with 2000 clients.

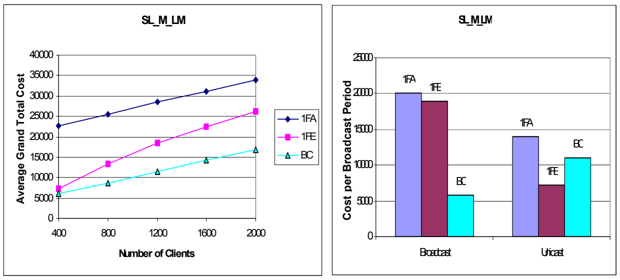

FIGURE 2. Network with Low Mobility

4.3.Network with other Data Distribution Models

We model two variations, one with high Zipf exponent (P), and one with no data item locality (NL). Recall that for an item with locality, it is likely that the interest for the item is highest at the center and gets less as the distance to the center increases. To model this, we use the Zipf distribution. The baseline case has a Zipf exponent, z=1; here we present the results with z=2. Figure 3, shows the cost of 1FE is from 40 to 25% lower than its baseline values, whereas the cost of BC is only 32 to 13% lower. 1FA is not sensitive to data item distribution, whereas 1FE and BC produced lower costs with the peaked distribution.

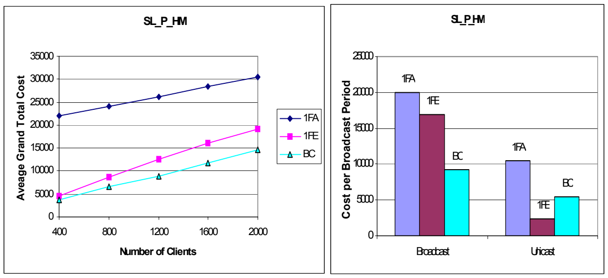

FIGURE 3. Network with Zipf = 2 (Peaked)

Figure 4 illustrates the results of a network with no item locality. 1FA and 1FE costs are significantly lower, and the advantages of clustering using clouds is less apparent. As BC capitalizes on the similarity of neighboring cells for deciding on the broadcast boundaries, as item locality disappears as does BC's advantage. Cost breakdown with 2000 clients is shown for both figures.

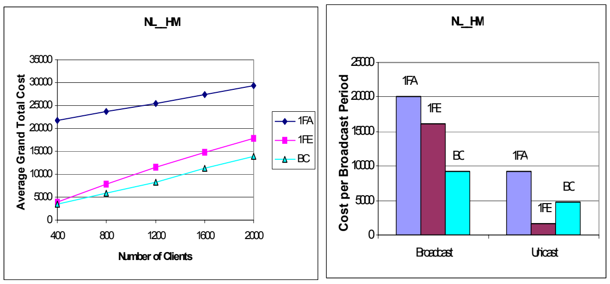

FIGURE 4. Network with No Locality (NL)

4.4.A Note on Subscription Costs

In this section, we report the total subscription cost under each protocol. Figure 5 shows the subscription costs for 2000 clients for each of the networks we simulated. Recall that subscription cost is a function of the move probability and subscription probability of the items; if a

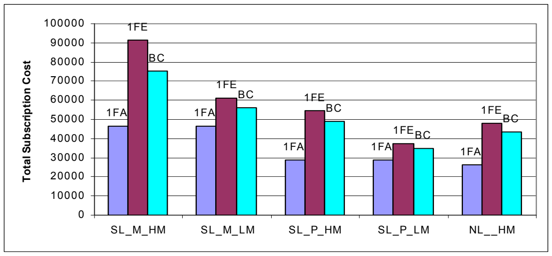

FIGURE 5. Subscription Cost

subscribed item is not in the broadcast, the client will have to request it. 1FA has the least subscription cost, and that cost is due to random changes in the clients' subscription set, and clients' location driven by item locality. It is a baseline by which to compare the other protocols. Any additional subscription cost incurred by 1FE and BC is due to the broadcast organization and boundaries.

4.5.Network with Higher Broadcast Size

Figure 6 illustrates the cost of the three broadcasting approaches for F=200, the fixed broadcast size parameter. With only 400 clients in the network, the cost of 1FE is 20% higher than that of BC. As the number of clients is increased, 1FE is 64% higher than BC. In the baseline case, this ratio varies between 30 to 53%. As F is increased, the unicast cost decreases since more items can fit in the broadcast. The increasing trend in 1FE is due to increased subscription activity: As F gets longer, more clients switch cells during the broadcast, thus increasing cost due to subscriptions. BC is affected similarly, but the use of the clouds dampens the effect of the subscription cost. Cost breakdown with 2000 clients is also shown.

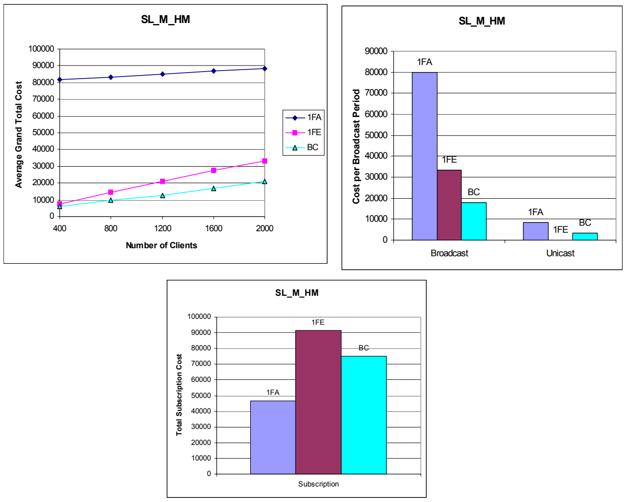

FIGURE 6. Network with Broadcast Size = 200

5. Conclusions

In this paper, we addressed a long neglected problem of optimally disseminating data broadcasts to a network of wireless cells. We first identified that broadcasts could be sent in two general formats: one for each cell (1FE) and one for the entire network (1FA). We noted that some data items could exhibit *locality*, i.e., a higher concentration of requests around certain geographic locations. We then highlighted the Broadcast Clouds (BC) approach that finds a better grouping between the extremes 1FA and 1FE, and a heuristic to form the *clouds* based on a cost-saving principle. We then compared the BC with the 1FA and the 1FE broadcasting approaches using a simple broadcast scheduling technique. Our results indicate that while each of the three approaches has strengths and weaknesses, the cost of BC is almost always bounded by that of either 1FA or 1FE. Therefore, the BC offers a nice alternative particularly when the network parameters are dynamic.

Our additional contribution is an understanding that the wireless bandwidth cost has an effect on the broadcast dissemination, which in turn affects all aspects of a data broadcasting application, including broadcast scheduling. Traditional performance metrics - the access and tuning times - though worthwhile, are not sufficient for optimizing the overall network cost because they consider only the clients' needs and not those of the service provider. Indeed, if the service provider can't make a profit, the service will not be offered. We hope that our work will motivate researchers to incorporate network costs in their analyses.

References

[1]Acharya, S., R. Alonso, M. Franklin and S. Zdonik. Broadcast Disks: Data management for asymmetric communication environments. Proceedings of ACM SIGMOD, San Jose, CA, May (1995).

[2]Aksoy, D. and M. Franklin. RXW: A scheduling approach for large-scale on-demand data broadcast. IEEE/ACM Transactions on Networking (1999) 7(6):846-860.

[3]Celik, A., J. Holliday and P. Ding. Will It Rain With Broadcast Clouds? Proceedings of the Tenth Americas Conference on Information Systems, New York, August 2004.

[4]Datta, A., D. VanderMeer, A. Celik and V. Kumar. Adaptive Broadcast Protocols to Support Efficient and Energy Conserving Retrieval from Databases in Mobile Computing Environments. *ACM Transactions on Database System*s (1999) 24(1):1-79.

[5]Erlebach, T. and A. Hall. NP-hardness of broadcast scheduling and inapproximability of single-source unsplittable min-cost flow. Proceedings of 13th Annual ACM-SIAM Symposium on Discrete Algorithms, San Francisco, CA, (2002), pp 194-202.

[6]Guo, Y., S. K. Das and C. M. Pinotti. A new hybrid broadcast scheduling algorithm for asymmetric communication systems: Push and Pull data based on optimal cut-off point. Proceedings of ACM MSWIM 2001, July 2001, Rome, Italy, pp 123-130.

[7]Hameed S. and N. H. Vaidya. Efficient algorithms for scheduling data broadcast. Wireless Networks 5 (1999) pp 183-193.

[8]Imielinski, T., S. Viswanath and B. R. Badrinath. Data on air: Organization and access. IEEE Transactions on Knowledge and Data Engineering, 9(3):353-372, May/June (1997).

[9]Kenyon, C. and N. Schabanel. The data broadcast problem with non-uniform transmission times. Proceedings of ACM-SIAM Symposium on Discrete Algorithms, January (1999), Baltimore, MD, pp 547-556.

[10] Su C.-J., L. Tassiulas and V. J. Tsotras. Broadcast scheduling for information distribution. Wireless Networks 5 (1999) pp 137-147.