Optimizing Network Bandwidth in Data Broadcasting

Abstract

The focus of this research is delivering data, such as stock prices, news, and multimedia to wireless users, while consuming the least bandwidth. Currently, the pricing mechanism for wireless data services is based solely on the number of bits a client receives or sends. However, the *actual cost* to the service provider depends on the usage of network resources, particularly the wired and wireless bandwidth. Since there are potentially many clients for the same data items, broadcasting the data will save bandwidth. Research has focused on scheduling the broadcasts such that client-side metrics are optimized. However, the supporting network infrastructure, i.e., provider-side cost, is usually not taken into account in broadcast scheduling. In this paper, we consider a wired network architecture such as the Internet where the broadcasts to the access points are sent using the popular IP Multicasting approach. Each access point receives the multicasts and wirelessly disseminates them to the users in their cells. We provide an Integer Programming formulation that minimizes the *total* bandwidth while at the same time allocating access points into multicast groups. We propose a heuristic to avoid setting up and solving this optimization problem frequently when clients are highly mobile thus groups change often.

1 Introduction

Wireless phones and text messaging (such as SMS) are currently the most widely used wireless services. Wireless carriers (e.g., Cingular, T-mobile) also offer E-mail, Internet access, and information services at a premium. Although the pricing mechanism for these services is based solely on the usage, the actual cost to the service provider (i.e., the carrier) depends on the usage of the network resources, particularly the bandwidth, for delivering the data. Therefore, the provider is motivated to employ mechanisms that consume the least bandwidth.

Wireless data include stock prices, news, and multimedia. Since there are potentially many clients for the same data items, broadcasting the data will save bandwidth. Many research articles have been published on the effectiveness of data broadcasting to wireless devices (including [1], [2], [3]). In data broadasting, data items are bunched together, and sent to the wireless channel where clients can download. Although the infrastructure is in place, data broadcasting has not yet materialized. One can attribute this to a lack of conviction on the part of the data providers that this service would indeed be profitable. Yet, a survey of existing research reveals that the cost of data broadcasting to the service provider has almost never been considered. This work will be novel in the sense that it will focus on the provider's cost in making the service available. A desired outcome of this research is to help providers come at the realization that data broadcasting will save costs.

In this paper, we look beyond the wireless infrastructure to derive the costs of data broadcasting. We consider a network architecture such as the Internet where the data broadcasts are sent to access points (or base stations) using the popular IP Multicasting approach[1]. In IP Multicasting, a client joins a multicast group or channel that delivers items of interest to the clients. Therefore, each data broadcast is treated as a multicast. Since each multicast packet is sent only once, and is replicated as needed down in the network, multicasting saves wired bandwidth. The protocol constructs a shortest path tree to each client, but the *contents* of the channels are usually independent from this multicast delivery tree. Also, a client may need to scan more than one channel if the client's items of interest are not all in one channel. We argue that the data channels can be designed more efficiently if the clients' items of interest and their positions on the delivery tree are known.

Therefore, the problem we are trying to solve is determining the content of the multicasts, given *a)* the set of data items requested by the clients, and, *b)* the multicast tree structure, while minimizing the network cost.

We solve the above problem using Integer Programming techniques. We also show that while it is easy to set up and solve the problem, due to the mobility of the clients, the problem may need to be solved very frequently, thus may not be very efficient in practice. Therefore, we propose the heuristic APPEND that avoids solving the problem until a threshold cost is reached.

The rest of the paper is organized as follows: we review existing research and techniques in Section 2, and the wireless and wired architectures and background on data broadcasting and IP multicasting in Section 3. We present the optimization model in Sect i on4. We outline the heuristic APPEND in Section 5, and conclude in Section6.

2 Related Work

Much work has been done on scheduling data broadcast in wireless networks. Acharya et al. introduced the Broadcast Disks approach for pushing data items to the clients [1]. Essentially, the broadcast is scheduled by "spinning" disks that hold data items. Then these disks are multiplexed on the broadcast channel. Therefore, depending on the size and the rotational speed of the disks, the data items may be repeated in the broadcast. Jiang and Vaidya suggest an optimality condition for a broadcast schedule that minimizes mean waiting time of the users and maximizes the percentage of requests served in a pure push scenario [3]. Su et al. formulate the broadcast schedule as a deterministic dynamic optimization problem that minimizes the average response time [6]. Aksoy and Franklin propose an algorithm to prioritize data items based on the number of requests and how long they are missing on the air [2]. A similar measure, called the *Ignore Factor* is also proposed in [4].

On the IP multicasting side, Celik et al. implemented a secure data broadcasting technique that delivers data items to subscribers [5]. Yet, all these techniques focus on client side metrics such as time to access data and the energy consumption.

---

1. In the IP literature, the term "broadcast" refers to flooding the network. We continue using the term "broadcast" to refer to the "structure" of the information that is being multicast.

3 Architecture and Model

We consider a wireless network connected to an Internet backbone as shown in Fi gure1.

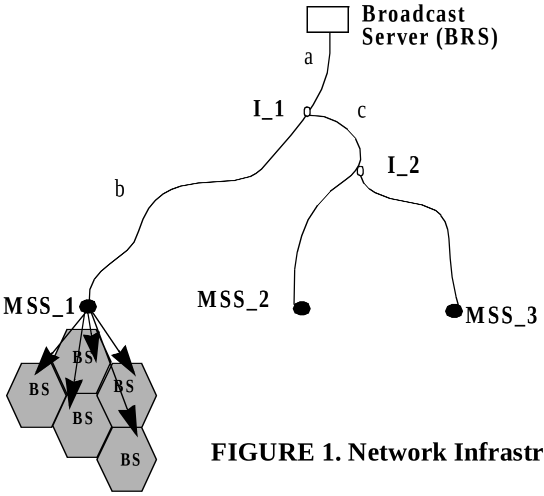

FIGURE 1. Network Infrastructure

Here, wireless clients are inside the wireless cells. Each cell is served by a base station (or access point) denoted as BS. MSS denotes a Mobile Support Station, which supports a group of BSs. Clients are subscribed to data items, thus they are members of multicast groups that include these data items. Each base station is connected to the Internet. The Broadcast Server (BRS) is in charge of preparing the data broadcast(s) and sending it to the multicast group(s).

In IP Multicasting, a (shortest path) tree is constructed from the source to the sinks based on the network characteristics and is used to send the multicast packets. In Figure 1, the lines connecting the Broadcast Server to the MSSs represent the routes prepared by the multicasting protocol. For example, the packets sent to MSS_2 travel from the BRS, to a router at I_1, pass the router at I_2, then arrive at MSS_2. The path between the BRS, I_1, I_2 and the MSSs could be actually composed of multiple routers and network switches. However, as we shall discuss shortly, only the network points where the routes intersect play a role in our model, therefore only these two intersection points in the figure, I_1, and I_2 are represented.

MSSs could belong to the same or different multicast groups. A multicast group is a set of nodes that receive the same multicast. For example, let, Group 1 = {MSS_1}, and Group 2 = {MSS_2, MSS_3}. Then, MSS_2 and MSS_3 will listen to the same multicast address for the data broadcast, but MSS_1 will have a different multicast address, thus will download a different data broadcast.

Let us now consider the cost of bandwidth and the effect of the multicast tree on the cost. A multicast packet travels along the tree, and is replicated when it has to cross an intersection point to reach all the members of a multicast group. Continuing with our example, the packets destined for Group 1 will follow the path BRS - I_1 - MSS_1, and those for Group 2 will follow BRS - I_1 - I_2, and will be duplicated at I_2 and follow I_2 - MSS_2 and I_2 - MSS_3. Clearly, the advantage of multicasting is that the packets destined for Group 2 are replicated only at the end, thus saving network bandwidth. Had the multicast tree been more sparse (e.g., if there were no intersections between the paths), the network bandwidth use under multicast would have been almost the same as the point-to-point packet delivery. Therefore, the fact that the multicast paths intersect incurs savings in bandwidth.

We now turn our attention to what is actually being multicast, that is, the contents of the multicast packets. Recall that the multicast is the network application to send data broadcasts to mobile users. Therefore, multicast packets contain the data that is being broadcast by the Broadcast Server (BRS). An MSS collects the data item requirements from its mobile clients and presents them to the BRS. These requirements in turn are used to determine the contents of the broadcasts. If the item requirements are homogeneous across the MSSs, the BRS could choose to prepare a single broadcast and construct a single multicast group that includes all the MSSs. If the data needs are diverse enough, the BRS could construct multiple broadcasts, and assign the MSSs into multicast groups. We propose to use a cost function to help determine the multicast groups and content.

The Network Cost (C): The cost is defined as the summation of the packet size (in bits) weighted by the path length (in network edges). The objective is to minimize this cost:

$$\text{Minimize } C = \sum_{k \in N} \text{len}(l_k)(p_k) \text{, where } p_k = \sum_{m=1}^{M} \text{sizeof}(p_{k,m}) \qquad \text{(EQ 1)}$$

$$|l_k| = \text{len}(l_k) = \text{length of edge } l_k \text{ and } |s_i| = \text{sizeof}(s_i) = \text{size of a multicast } s_i \text{ in bits}$$

Therefore p_k is the sum of all packets passing over the edge k in bits. These packets are sent as part of a multicast m.

To shed some intuition on the problem, let the length of the path from BRS to I_1 be a, and that of the path from I_1 to MSS_1, and I_1 - I_2 - MSS_2 be b and c, respectively (these are shown on Figure1). Let us assume that the data items requested by MSS_1 and MSS_2 are somewhat similar, but not exactly the same. Then, the BRS has a choice of *a)* sending the same data broadcast to both MSSs, or *b)* composing two different broadcasts. Let us assume that the size of the broadcasts in bits in *a)* is X, and those in *b)* are Y and Z. Assume that $X>Y$, $X>Z$ and $X<Y+Z$. The cost of carrying one broadcast equals $(a+b+c)X$ as opposed to two broadcasts $(a+b)X+(a+c)Y$. Sending two broadcasts is more advantageous if $\frac{X}{Y} > 1 + \frac{a}{c}$. Let us now consider the following two scenarios:

1. $a<<c$: This means that the location of I_1 is very close to BRS. In this case, sending two broadcasts is justified since only a small portion of the path (BRS to I_1) will carry additional packets.

2. $a>>c$: This will require sending a single broadcast to both MSSs since the cost is cheaper.

Therefore, the problem we are trying to solve is to determine the content of the multicasts given the set of data items requested by the MSSs and the multicast tree structure while optimizing the network cost.

4 Optimization Model

In this section we present the optimization problem. Let each MSS have a set s_i of data items that are requested by the mobile users in the MSSs service area. Let us define S as the set of all possible multicasts based on the requirements of the MSSs. For example, if MSS_1, MSS_2 and MSS_3 request s_1, s_2, and s_3, respectively, then $S = \{s_1, s_2, s_3, s_{12}, s_{13}, s_{23}, s_{123}\}$. Here, $s_{ij} = s_i \cup s_j$, and similarly $s_{ijk} = s_i \cup s_j \cup s_k$. For ease of notation, we numerate the members of S and refer to them with the index i as $n_1, n_2, ..., n_K$.

We formulate the cost minimization problem as follows:

$$\text{Min Cost} = C = \sum_{k=1}^{L}\left[|l_k| \times \left(\sum_{s_i \in S} |n_i| \times d_{ki}\right)\right] \qquad \text{(EQ 2)}$$

Here, d_{ki} are 0,1 decision variables, and $d_{ki} = 1$ if a multicast n_i passes via an edge l_k, and is zero otherwise. L is the total number of edges on the multicast tree.

Demand constraints: The demand on the edges leading to MSSs must be satisfied by the flow on these edges. Let N be the number of MSSs. Therefore,

$$\sum_{n_i \subseteq s_i} d_{ji} = 1 \qquad \text{(EQ 3)}$$

for each link j such that l_j is the last link to an MSS. i is the index that corresponds to n_i.

Topology and flow constraints: Simply, the packets leaving an intersection point must have arrived there. Therefore, a multicast leaving an intersection point I, must have arrived in the first place. These constraints also reflect the topology of the multicast tree.

$$d_{ki} \le d_{li} \qquad \text{(EQ 4)}$$

for each out edge (link) k and in edge l for an intersection point I as shown in Figure 2. Here, the index i corresponds to n_i. One flow constraint is included for each outgoing edge.

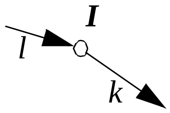

FIGURE 2. Flow constraint at I

4.1 Example problem

We give the objective function and the constraints for the problem in Figure 1 redrawn in Figure 3. s_1, s_2, and s_3 represent the demand set for the MSSs. l_1, l_2, l_3, l_4 and l_5 are the links on the tree. I_1 and I_2 are the intersection points. S = {s_1, s_2, s_3, s_{12}, s_{13}, s_{23}, s_{123}} and N = {n_1, n_2, n_3, n_4, n_5, n_6, n_7}, where n_i corresponds to the i^{th} ordinal element in S.

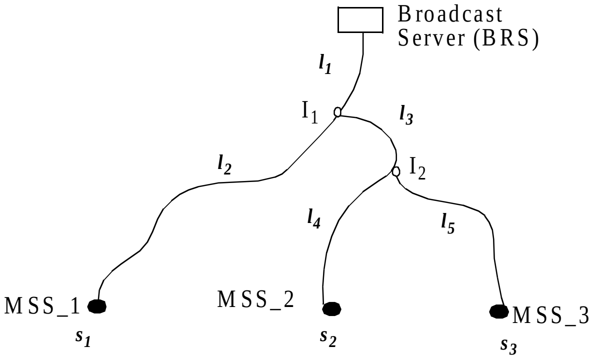

FIGURE 3. Example problem

The LP objective function is thus:

$$\begin{aligned}\min C \;=\; & |l_1| \times (|n_1| \times d_{11} + |n_2| \times d_{12} + |n_3| \times d_{13} + \\ & |n_4| \times d_{14} + |n_5| \times d_{15} + |n_6| \times d_{16} + |n_7| \times d_{17} \\ &) \\ & + |l_2| \times (|n_1| \times d_{21} + |n_2| \times d_{22} + |n_3| \times d_{23} + \\ & |n_4| \times d_{24} + |n_5| \times d_{25} + |n_6| \times d_{26} + |n_7| \times d_{27} \\ &) \\ & + \ldots \\ & + |l_5| \times (|n_1| \times d_{51} + |n_2| \times d_{52} + |n_3| \times d_{53} + \\ & |n_4| \times d_{54} + |n_5| \times d_{55} + |n_6| \times d_{56} + |n_7| \times d_{57} \\ &)\end{aligned} \tag{EQ 5}$$

Demand constraints are:

$$\begin{aligned}\text{demand at}\quad & l_2 : d_{21} + d_{24} + d_{25} + d_{27} = 1 \\ \text{demand at}\quad & l_4 : d_{42} + d_{44} + d_{46} + d_{47} = 1 \\ \text{demand at}\quad & l_5 : d_{53} + d_{55} + d_{56} + d_{57} = 1\end{aligned} \tag{EQ 6}$$

$$d_{ij}=0,1 \tag{EQ 7}$$

Topology and flow constraints are as noted in Table 1 :

at I_1	at I_2
$d_{21} \le d_{11}$	$d_{41} \le d_{31}$
$d_{31} \le d_{11}$	$d_{51} \le d_{31}$
$d_{22} \le d_{12}$	$d_{42} \le d_{32}$
$d_{32} \le d_{12}$	$d_{52} \le d_{32}$
$d_{23} \le d_{13}$	$d_{43} \le d_{33}$
$d_{33} \le d_{13}$	$d_{53} \le d_{33}$
…	…
$d_{27} \le d_{17}$	$d_{47} \le d_{37}$
$d_{37} \le d_{17}$	$d_{57} \le d_{37}$

TABLE 1. Topology and flow constraints at intersection points.

Note that all of the MSSs in the example problem are leaf nodes for ease of representation. It is possible to formulate the problem similarly when an MSS is an internal node.

4.2 Solution of the example problem

The example problem in Section 4.1 is solved with the following parameter values shown in Table 2. Here, s_{12} = {1, 2, 3, 4} and its size, represented by $|n_4|$ = 4. Similarly, s_{13} = {1, 2, 3, 4, 5} and $|n_5|$ = 5; s_{23} = {2, 3, 4, 5} and $|n_6|$ = 4; s_{123} = {1, 2, 3, 4, 5} and $|n_7|$=5.

parameter	value
s_1	{1,2}, $\|n_1\|$=2
s_2	{2,3,4}, $\|n_2\|$=3
s_3	{3,4,5}, $\|n_3\|$=3
l_1	5
l_2	3
l_3	2
l_4	2
l_5	2

TABLE 2. Parameters for the example problem

The objective function of the minimization problem is 60, and the decision variables d_{11}, d_{16}, d_{21}, d_{36}, d_{46} and d_{56} are set to 1. This means that MSS$_1$ will receive n_1={1, 2}, and MSS$_2$ and MSS$_3$ will receive n_6 = {2, 3, 4, 5}. Recall that $n_6 = s_2 \cup s_3 = \{2, 3, 4, 5\}$. This solution effectively forms two multicasts, the first one comprising of MSS$_1$ only, and the second one having MSS$_2$ and MSS$_3$

5 Changes in the problem

The optimization problem discussed above is rather straightforward to model: objective function and the constraints are well defined. However, the size of the problem makes solving it costly. In the following, we discuss cases where the problem needs to be reformulated and solved.

5.1 Adding a new MSS

The number of decision variables in the IP formulation depends on the number of the demand sets and the number of links on the multicast tree. The number of decision variables, *ndvar*, is defined as $ndvar = nlink \times nset$, where *nlink* is the number of links, and *nset* is the number of possible demand sets. We have $nset = 2^{nMSS} - 1$, where *nMSS* is the number of MSSs requesting a set of items. Thus adding one more MSS to the problem will increase the number of decision variables by at least 2^{nMSS}. The number of decision variables in each demand constraint will double, and a new demand constraint will be added for the new MSS. New flow constraints must be added based on the new links. The objective function will have to incorporate all the new decision variables.

5.2 Changes in the subscription sets

The subscription sets of the MSSs could vary depending on the clients' needs. Obviously, an item that is no longer requested by the clients does not need to be sent to the MSS any more. Similarly, when a client requests a new item, it should be included in the set. This would effectively change the demand constraints in the IP.

Assuming that a client's data items of interest are fixed, and the client wishes to continously access these items, there are two reasons that cause a change in the subscription set of an MSS: 1) all clients with a given data item move out of the MSS, and 2) at least one client with a data item not already in the subscription set of MSS move in.

5.2.1 How often does the problem change?

To find how often the problem changes due to changes in the subscription sets, we resort to queueing theory, and model the system as an $M/M/\infty$ queuing system. This is an infinite server system (a.k.a. delay sytem). We derive the average number of changes.

We first derive the solution for a single MSS, *j*. To simplify the analysis, we assume that each MSS consists of a single wireless cell. We start by assuming that there is only one item *i* (we will generalize this later), and that the clients for the item arrive and depart the MSS in a Poisson pattern, with rates λ and μ, respectively. The steady state probabilities for this system are given by:

$p_n = e^{-a} \frac{a^n}{n!}$, $n = 0, 1, 2, \ldots$ where $a = \frac{\lambda}{\mu}$ (see Allen [7], p. 284). p_n is defined to be the probability that the system is at state *n* (i.e., has *n* clients for the item) in the steady state.

A change in the subscription set occurs when the MSS moves back and forth between *state 0* (with no clients for the item) and *state 1* (with only one client for the item). Let *M(i,j)* denote the event that a client of the item *i* moves in between states *0* and *1* in MSS *j*. We derive the expected value of the number of moves from state *0* to state *1* and vice versa during a given time period *T* as:

$$E[M(i,j)] = p_0(\lambda T) + p_1(\mu T) = e^{-\lambda/\mu}(\lambda T) + e^{-\lambda/\mu}(\lambda/\mu)\mu T = 2e^{-\lambda/\mu}\lambda T \quad \text{(EQ 8)}$$

$$\text{When } \lambda = \mu,\ E[M(i,j)] = \frac{2\lambda T}{e}. \quad \text{(EQ 9)}$$

We can generalize this derivation to include not just one item but all the items in the database by simply multiplying *E[J]* by *n* items. Similarly, the solution extends to multiple MSSs. Therefore, the expected number of *system-wide* changes, *E[M]* in the subscription set is:

$$E[M] = E[M(i,j)] \times n_{items} \times n_{MSS} = \frac{2\lambda T}{e} \times n_{items} \times n_{MSS}. \quad \text{(EQ 10)}$$

Therefore, for a given time interval, as *nMSS* or n_{items} increases, the frequency of solving the IP problem increases by λ-fold. To illustrate this, let $\lambda = 1/60$, ($\lambda = \mu$), T=60*s*, $n_{items} = 5$, *nMSS* = 3. *E[M]*= 10, meaning that, the IP must be updated and solved 10 times in just one minute!

5.3 Heuristic to avoid resolving the problem

We propose a heuristic to reduce the frequency of resetting and solving the optimization problem. The heuristic APPEND, listed in Table 3, exploits the fact that a modified subscription set may become equal to or contained in an existing subscription set. If that is the case, then the MSS of the modified subscription set is simply added to the existing group. If not, the modified MSS is made into a new group and sent a new broadcast. This solution will clearly be sub-optimal. However, solving the optimization problem will be avoided until the cost of the system exceeds the cost of the previous optimal solution by a given factor, *k*.

6 Conclusion

In this paper, our objective is to convince the wireless providers that data broadcasting conserves bandwidth, if implemented correctly. We argue that existing solutions have not justified their claims that data broadcasting saved costs. Therefore, we have designed an implementation of data broadcasting using the IP Multicasting approach, which we believe is the most practical method of data delivery to large number of users. In our method, we group the clients based on their location in the network, and assign each location to a multicast group. Each multicast group receives the same data broadcast. The objective is the minimization of the used bandwidth. Given a multicast tree, we showed that it is possible to optimally solve this problem. We have also proposed a heuristic to avoid resolving the optimization problem when the clients are highly mobile. As future

APPEND

```
begin
  set ThresholdCost = TotalCost*k (k>0)
  while TotalCost < ThresholdCost
  do
   if MSS$_i$'s subscription set is changed to $n_x$ then
     if $n_x \subseteq n_y$ and $n_y$ is already being sent to a group
        then add MSS$_i$ to that group, route $n_y$ to MSS$_i$
       else
       make MSS$_i$ a new group and route $n_x$ to MSS$_i$
     end if
    Update TotalCost
   end if
  end while
end
```

TABLE 3. Heuristic APPEND

work, we are planning to test our heuristic in a simulated network environment and obtain results to help us devise better techniques.

References

[1] S. Acharya, R. Alonso, M. F. Franklin and S. B. Zdonik. "Broadcast disks: Data management for asymmetric communication environments", Proceedings of the ACM Special Interest Group on Management of Data Conference *(SIGMOD 1995)*, 1995.

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

[3] S. Jiang and N. H. Vaidya. "Scheduling data broadcast to 'Impatient' Users", Proceedings of the ACM International Workshop on Data Engineering for Wireless and Mobile Access *(MobiDE)*, 1999, 52-59.

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

[5] A. Celik, J. Holliday and B. Ramavarjula. "I-DG and HI-DG: Scalable and Efficient Techniques for Subscription-Based Data Dissemination via IP Multicasting", E-Commerce Research Journal *(ECR)*, 3 (1-2), pp. 143-166, 2003.

[6] C.-J. Su, L. Tassiulas and V. J. Tsotras. "Broadcast Scheduling for Information Distribution", Wireless Networks 5 (1999) 137-147.

[7] A. O. Allen. "Probability, Statistics, and Queuing Theory with Computer Science Applications". Second Edition. Academic Press, Inc. 1990.