Reducing Multicast Traffic Load for Cellular Networks using Ad Hoc Networks

Li Lao[1], Jun-Hong Cui[2]
llao@cs.ucla.edu, jcui@cse.uconn.edu
[1] Computer Science Department, University of California, Los Angeles, CA 90095
[2] Computer Science & Engineering Department, University of Connecticut, Storrs, CT 06269

Abstract

Recently, there has been extensive research on integrating cellular networks and ad hoc networks to overcome the limitations of cellular networks. Although several schemes have been proposed to use such hybrid networks to improve the performance of individual multicast groups, they do not address the Quality of Service (QoS) issues when multiple groups are present in the networks. Our work, on the other hand, considers an interesting scenario of hybrid networks when an ad hoc network cannot commodate all the groups and a base station has to select a subset of groups to optimize its bandwidth savings and maximize the utilization of the ad hoc network, while providing QoS support for multicast users. In this work, we develop a network model for multicast admission control which takes wireless interference into account, formulate the group selection problem as a multidimensional knapsack problem, and propose an Integer Linear Programming (ILP) formulation and a polynomial-time dynamic algorithm. We also examine a distributed implementation of the dynamic algorithm in real systems. Simulation studies demonstrate that the dynamic algorithm is able to achieve very competitive performance under various conditions, in comparison with the optimal solution computed by the ILP approach.

1 Introduction

In recent years, wireless networks have been evolving rapidly with the proliferation of wireless devices and the widespread deployment of wireless infrastructures, such as 2.5G and 3G cellular networks. A cellular network usually consists of a wired backbone formed by some fixed base stations. Mobile users (or nodes) can subscribe to a base station and communicate directly with it. The main advantage of cellular networks is that they provide large cell coverage; on the other hand, however, they often offer very limited bandwidth (e.g., 3G networks support bandwidth of 384Kbps to 2Mbps), and base stations are potential performance bottlenecks when there are a large number of users in the same cell.

To overcome the limitations of cellular networks, there has been extensive research work on integrating ad hoc networks and cellular networks. In wireless ad hoc networks, mobile nodes form a network instantly without infrastructure support. Each mobile node could act as a router in the network, relaying data packets for others over multi-hop wireless links. Thus, in a hybrid cellular-ad hoc network, mobile hosts can be used to relay packets for other nodes to increase the coverage of base stations or avoid dead spots [13, 2]. They can also redirect traffic within a congested cell to a neighboring non-congested cell [19, 5]. In addition, system throughput can be increased when base stations switch between ad hoc mode and cellular mode [10], or when mobile users experiencing good channel quality (called proxies) relay packets for other nodes [14, 21].

Inspired by the above work for point-to-point (i.e., “unicast”) traffic, a number of hybrid architectures have been proposed for group communication applications when the same content is sent to multiple receivers [8, 9, 16]. In these architectures, proxies (i.e., mobile hosts with high receiving rate) use “multicast” to send packets to a group of users. Multicast is an efficient mechanism for point (or multi-point) to multi-point communication, which has been widely used in both wired [17, 6, 3, 15] and wireless networks [4, 12, 18, 7][1]. By exploiting the benefits of multicasting among mobile users (i.e., ad hoc multicast), these architectures can enhance the network performance, especially for applications with heterogenous receivers. However, previous work mainly focuses on individual multicast groups, and does not consider the Quality of Service (QoS) when multiple groups co-exist in the networks.

Our work, on the other hand, targets at the scenarios in which base stations manage multiple groups simultaneously. We consider all types of groups: groups solely

---

[1] In multicast, a tree or mesh structure is usually used for data delivery. Packets are duplicated only at branching nodes, thus reducing identical packets on network links and preserving link bandwidth.

formed by local users in one cell; groups consisting of users from other cells[2]; virtual groups implicitly formed by users receiving the same content[3]. When there are a large number of users and groups in a cell, the bandwidth requirement for the base station will be very demanding. As we know, bandwidth is scare resource for a base station, thus many groups obviously will overload the base station since all the traffic goes through this "bottleneck" point (we call this communication mode "cellular mode"). To reduce multicast traffic load on base stations, we also propose using ad hoc multicast (called "ad hoc mode") as in [8, 9, 16]. However, due to the capacity limit of the ad hoc network, if all the groups are routed in ad hoc mode, the network may become congested and the performance of these groups may be degraded significantly. Thus, it is critical to determine which groups should be admitted into the ad hoc network so that the work load on base stations is minimized and the utilization of the ad hoc network (without exceeding its capacity) is maximized at the same time. We refer to this problem as "group selection problem".

In this paper, we formulate the group selection problem as a multidimensional knapsack problem, and present an ILP (Integer Linear Programming) formulation and a dynamic algorithm to solve the hard problem efficiently. In both of the approaches, we take wireless interference among neighboring nodes into consideration. We conduct simulations to evaluate the proposed algorithms. The results demonstrate that these algorithms can achieve very competitive performance. The ILP approach gives the optimal solution, but it is not feasible for large networks with high capacities. The proposed dynamic algorithm, on the other hand, can achieve near-optimal solutions, and it is more appropriate for on-line dynamic systems.

The remainder of this paper is organized as follows. In Section 2, we briefly review some related work. We formulate the group selection problem in Section 3 and present our solutions in Section 4. Then we evaluate the performance of the proposed algorithms in Section 5 by simulations. Finally, we conclude the paper in Section 6.

2 Related Work

The recognition of the trade-offs between cellular and ad hoc networks has stimulated a lot of efforts to integrate these two types of networks for various purposes.

Some proposals focus on increasing cell coverage or balancing load among cells [13, 2, 19, 5]. In [13], the authors use mobile nodes to relay packets between base stations and mobile nodes in order to reduce the number of base stations and increase cell coverage. In [2], an ad hoc GSM (A-GSM) network platform is developed to improve GSM coverage and avoid dead spots. Wu et al. propose a mobile-assisted data forwarding (MADF) scheme to direct traffic in a hot cell to neighboring cold cells using an ad-hoc overlay, thus improving system throughput and reducing end-to-end delay [19]. In [5], the authors propose a load balancing scheme called Integrated Cellular and Ad Hoc Relay (iCAR) for cellular networks, which places ad hoc relay nodes at strategic locations to relay traffic from congested cells to non-congested ones.

Some other research efforts explore the benefits of both types of networks and aim to improve system throughput [10, 11, 14, 21]. In [10], Hsieh and Sivakumar investigate the performance trade-offs between cellular and multi-hop ad hoc models and present a simple hybrid model, in which a base station and mobile hosts switch between cellular and multi-hop modes to maximize the total throughput. Their further study in [11] indicates that using ad hoc models in cellular networks may degrade the throughput and fairness performance. To address these drawbacks, they propose several approaches, including assisted scheduling, hybrid station, and dual mode service. In [14], a Unified Cellular and Ad-Hoc Network (UCAN) architecture has been proposed to allow mobile nodes with high data rates (called proxies) to relay data for users experiencing poor channel quality in order to improve aggregate throughput. A more recent approach, QAWBA [21], aims at providing better QoS support for hybrid networks by a QoS-aware wireless bandwidth aggregation mechanism.

While the above work centers on point-to-point traffic, there have been some recent proposals on enhancing cellular multicast performance using ad hoc networks. Hauge et al. propose a hybrid network architecture to increase the coverage of high bandwidth group service [8, 9]. Inspired by UCAN, Park and Kasera develop a routing algorithm to find ad hoc paths from proxies to cellular multicast receivers [16]. Unlike these two schemes which attempt to improve the performance of individual multicast groups, the goal of our work in this paper is to maximize the performance of the whole network when there are many co-existing multicast groups. Our basic idea is to minimize the load of cellular base stations by selecting appropriate multicast groups to be handled by ad hoc networks.

3 The Group Selection Problem

3.1 Problem Description

We consider a single cell with a base station serving a number of mobile nodes. Each mobile node has two wire-

---

[2] In this case, for a local cell, the base station is treated as one member of the group.

[3] Examples for virtual groups are web content fetching and streaming media distribution. When there are multiple clients in the local cell, web caching proxies and media streaming servers can be selected among the mobile nodes and the base station.

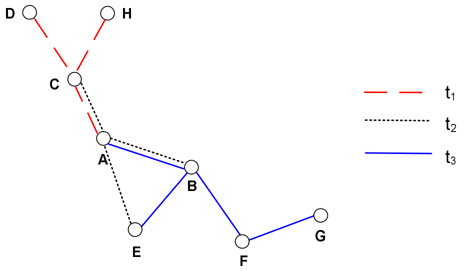


Figure 1. An example of the multicast group selection problem.

less network interfaces: a high data rate (HDR) interface and an IEEE 802.11 interface. When there is a need for a set of nodes (including mobile nodes and the base station) within the cell to set up a group for communication among themselves, each mobile node connects to the base station, sending data to and receiving data from the base station using point-to-point links[4]. We refer to this relaying scheme as "cellular mode". Alternatively, we have "ad hoc mode", in which group members can set up an ad hoc multicast tree using 802.11 network interface. Mobile nodes in the cell can duplicate and relay packets for this group. In this way, the base station does not need to relay packets (or relays at most once if it is a member of the group), thus saving some bandwidth. The amount of bandwidth saved is proportional to the number of users within the group (i.e., the group size) and the group data rate. For convenience, we only consider the case when all the members are mobile nodes in the rest of the paper. Our solutions can be easily adapted to the case when the base station is also a member of the group, which will be explained in Section 3.2.3.

A group can use either cellular mode or ad hoc mode. Intuitively, it is good for the base station if all the groups are in ad hoc mode. However, when there are multiple coexisting groups in the network, the ad hoc network may not be able to accommodate all the groups due to its limited capacity. In this case, the base station needs to determine how many groups and which groups to be switched to ad hoc mode in order to reduce its bandwidth consumption while satisfying the QoS requirements of the groups. We call this problem the **group selection** problem in this paper. We give an example in Fig. 1. In the illustrated network, there are three multicast groups g_1, g_2, and g_3, and their corresponding trees t_1, t_2, and t_3. Group g_1 has members {A, D, H}, g_2 has members {A, B, C, E}, and g_3 has members {A, B, E, G}. Obviously, node A is very likely to be overloaded,

Table 1. Notations and Definitions

Notation	Definition
N	network
V	nodes
E	links
G	multicast groups
C_i	capacity of node i
$\|g_j\|$	size of group g_j
r_j	required data rate of group g_j
bw_j	bandwidth vector of group g_j
bw_{ij}	required bandwidth on node i for group g_j
m	number of nodes
n	number of links
n_g	number of multicast groups

and we assume this is the case. In other words, the ad hoc network cannot accommodate all three groups. Thus, the base station has to select and accommodate one or two multicast groups (e.g., group g_2) in order to maximize its bandwidth reduction without violating the capacity limit of each mobile node. To solve this problem, we have to first compute the amount of resource (i.e., bandwidth in this paper) usage for each multicast group, and then select a subset of groups based on their bandwidth consumption and the ad hoc network capacity.

3.2 Problem Formulation

3.2.1 Network Model

We model the network as a graph $N = (V, E)$, where V denotes the set of mobile nodes and E denotes the set of links (a link exists between two mobile nodes if their distance is smaller than their transmission range). We define $m = |V|$ and $n = |E|$. For convenience, we assume all the mobile nodes in the network are in the coverage of the base station (in fact, our solutions can be easily extended to the scenarios where some nodes cannot be covered by the base station, as shown in Section 3.2.3). Each mobile node has a bandwidth capacity C_i. The set of multicast groups is represented by G and we define $n_g = |G|$. Each group $g_i \in G$ has a required data rate r_i. The notations used in this paper are summarized in Table 1.

Note that our focus in this paper is not multicast tree computation; thus, we assume a multicast routing algorithm is available. This assumption is justified by the large number of existing algorithms and protocols. In addition, our proposed solutions are also applicable when the routing algorithm or protocol computes meshes instead of trees.

Wireless Interference Given a network N, a group g_i and its required data rate r_i, it is straightforward to compute bandwidth requirement for this group in wired networks:

---

[4] In this study, we assume the base station only uses point-to-point link for reliability and heterogeneity handling. Note that even though Broadcast/Multicast Services (BCMCS) have been proposed for 3G networks[1], it has been shown in [16] that the receiver goodput degrades considerably with the increased heterogeneity of mobile nodes.

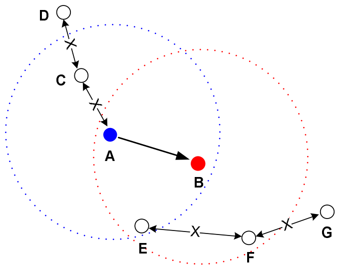
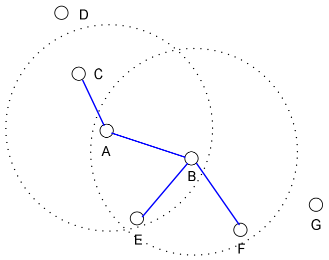


Figure 2. An illustration of wireless communication interference.



Link	Affected nodes
C-A	A, B, C, D, E
A-B	A, B, C, E, F
B-E	A, B, E, F
B-F	A, B, E, F, G

Figure 3. An example of the bandwidth requirement of multicast trees.

we just need to compute a multicast tree, since each link requires the same amount of bandwidth r_i. In wireless ad hoc networks, however, due to the shared wireless channel, data transmission between two wireless nodes may affect the available bandwidth at their neighbors. Thus, we have to take wireless interference into consideration.

To avoid packet collision caused by wireless interference, IEEE 802.11 employs Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA) mechanism for unicast communication[5]. For example, when a node wants to send data packets to another node, they have to exchange Request To Send (RTS) and Clear To Send (CTS) messages before data transmission, and after successful transmission of data packets, the receiver needs to reply an acknowledgement to the sender. During this RTS/CTS/DATA/ACK message exchange period, neither the sender's neighbors nor the receivers's neighbors can transmit packets.

We show an example in Fig. 2. When node A is transmitting packets to node B, the neighboring nodes C, E, and F that are one hop away from A or B cannot send or receive packets. In other words, packet transmission on link A-B consumes channel capacity not only at A and B, but also at their neighboring nodes. As a result, if we want to reserve one unit of bandwidth on link A-B, we need to reserve one unit of bandwidth at nodes A, B, C, E, and F[6]. In this way, we can obtain the bandwidth required on each node when transmitting packets on a link. It is important to point out that this interference model is based on the CSMA/CA mechanism of 802.11. If a different MAC protocol is used, the model needs to be adapted accordingly.

Bandwidth Requirement for Multicast Trees For a multicast group g_j with rate r_j, we can compute its multicast tree, and obtain the amount of bandwidth required on each node by all the links in the tree. In this way, we get the bandwidth requirement for this group. We can represent this requirement by a *bandwidth vector* $bw_j = (bw_{1j}, ..., bw_{ij}, ..., bw_{mj})$, where bw_{ij} denotes the amount of bandwidth required on node i for group g_j. Fig. 3 shows an example of a multicast group with members $\{$A, B, C, E, F$\}$ and its delivery tree. Each link in the tree interferes with a number of nodes (shown in the table). If assuming the data rate of this multicast group is one unit, we can sum up, for each node, the required bandwidth of these tree links and obtain the bandwidth vector $(4, 4, 2, 1, 4, 3, 1)$ for nodes (A, B, C, D, E, F, G). Correspondingly, if a subset of groups G' reserve bandwidth in the ad hoc network, the total bandwidth reserved on node i can be computed as $\sum_{g_j \in G'} bw_{ij}$, which should not exceed the given capacity C_i.

3.2.2 Formulating the Problem

As mentioned earlier, the group selection problem is to select a subset of multicast groups (which will be switched to ad hoc mode) to maximize the bandwidth saving at the base station while satisfying the capacity constraint of the ad hoc network. When the base station switches a multicast group g_i from cellular mode to ad hoc mode, the bandwidth saving for the base station is $|g_i| \times r_i$, where $|g_i|$ is the size of group g_i. Given the above network model, we can formulate the multicast group selection problem as follows.

Input: a set of multicast groups G, the required data rate r_j and the bandwidth vector bw_j for each multicast group $g_j \in G$, and the node capacity C_i for each node $i \in V$.

Output: a subset of groups $G' \subset G$ to maximize the bandwidth savings $\sum_{g_j \in G'} |g_j| \times r_j$ under the capacity constraint $\sum_{g_j \in G'} bw_{ij} \leq C_i$, where $i \in V$.

This problem is essentially a multidimensional knapsack problem: given a set of items $S = 1, ..., n$, where item i has an m-dimensional size $(r_{i1}, ..., r_{im})$ and a value v_i, and a knapsack with an m-dimensional size $(b_1, ..., b_m)$, find a subset $S' \subset S$ that maximizes the values of all the items that fit into the knapsack. Intuitively, if we consider each group as an item with a size (i.e., bandwidth consumption) and a value (i.e., bandwidth saving), we want to pack some

---

[5] In ad hoc networks, even though multicast can take advantage of the broadcast nature of wireless links, we choose not to use the broadcast mechanism for multicast in our work, because broadcast does not involve collision avoidance and data acknowledgement, and hence provides no reliability support.

[6] This observation is motivated by [20].

groups into the ad hoc network with a given size (i.e., capacity constraint) such that the total values (i.e., bandwidth savings) of these groups is maximized.

3.2.3 Discussions

We simplify the network model by assuming all the members in a group are located within the same cell. When a group has members dispersed across cells or wired/wireless heterogeneous networks, we can regard the base station in each cell as a "super" group member. In a global scope, the base station receives data from or sends data to other base stations or the Internet on behalf of the remaining members in its cell. Within a cell, when in cellular mode, the base station still communicates with all members using point-to-point links; when in ad hoc mode, it only needs to directly communicate with one member that acts as a relay node. Hence, our model can be easily applied in this scenario: we only need to decrease the bandwidth savings for a group g_i from $|g_i|r_i$ to $(|g_i|-1)r_i$ and update the bandwidth vector to include the required bandwidth for the link between the base station and the relay member.

The second assumption we make is that all the nodes can be covered by the base station. When some nodes are out of the transmission range of the base station, they can find other members as proxies to relay data for them. In this case, since these nodes must use ad hoc mode no matter which mode the group is in, they need to reserve bandwidth for the paths from the proxies to themselves (i.e., the capacity of each node affected by these paths needs to be reduced whether the group is selected or not). Then the base station can compute bandwidth savings and bandwidth vector for this group excluding these "isolated" members.

4 Algorithms

In this section, we first present an Integer Linear Programming (ILP) formulation for the multicast group selection problem, which gives the optimal solution. This approach is applicable when all the groups are known beforehand. However, it is not appropriate for on-line systems when groups dynamically join and leave. Thus, we propose a dynamic algorithm with polynomial-time complexity. For the comparison purpose, we also present a naive dynamic algorithm. Finally, we suggest a distributed implementation of the dynamic algorithm.

4.1 An ILP Formulation

For the ILP formulation, we define the following variables:

$$x_j = \begin{cases} 1, & \text{if group } g_j \text{ is selected} \\ 0, & \text{otherwise} \end{cases} \qquad \forall j \in [1, n_g],$$

where n_g is the number of multicast groups.

The objective is to maximize the amount of bandwidth savings:

$$\max \sum_{j=1}^{n_g} |g_j| \times r_j \times x_j,$$

subject to the node capacity constraint:

$$\sum_{j=1}^{n_g} bw_{ij} \times x_j \le C_i \qquad \forall i \in [1, m],$$

where m is the number of mobile nodes in the network.

The complexity of this ILP formulation is $O(n_g + m)$ in terms of the number of variables and the number of constraints. By solving the ILP, we can obtain solutions for x_j ($j \in [1, n_g]$), from which we are able to determine the optimal set of groups to be selected to ad hoc mode.

In this ILP solution, we assume that multicast groups and their requirements are known a priori. In real systems, multicast groups can join or leave the network at any time. If we use this ILP approach, we must do re-computation every time a group joins or leaves to obtain optimal solutions for the dynamic scenarios. However, this approach requires intensive computation and may be very time consuming, especially when the network is large (i.e., m is big) or there are many groups (i.e., n_g is large). In addition, since the solutions computed at different times might be very different, some groups may need to switch their modes from time to time, causing data delivery disruptions. To overcome these difficulties, we develop an on-line algorithm to handle group dynamics (referred to as "dynamic algorithm" in this paper), which is presented in the following subsection.

4.2 A Dynamic Algorithm

Before presenting the dynamic algorithm, we define a utility function for a group g_j:

$$u_j = \frac{|g_j| \times r_j}{\sum_{i \in V} bw_{ij}}.$$

In this equation, the numerator stands for the bandwidth savings at the base station if this group is selected, and the denominator represents the total bandwidth required for this group. Intuitively, our goal is to maximize the bandwidth savings, but we do not want to spend too much resource for a single group. Thus, assuming every group has the same data rate, a large group which requires a small amount of bandwidth tends to have higher utility and is potentially better than another group with smaller utility. The utility metric is used by the dynamic algorithm as a major criteria to select groups in a "greedy" way.

Algorithms 1 and 2 show the dynamic algorithm as groups join and leave the network, respectively. As shown

Algorithm 1 Join(g)

```
1:  for $g$, compute multicast tree, bandwidth vector, and utility
2:  if admit($g$) then
3:      reserve_bandwidth($g$)
4:      $g.mode \leftarrow ad\_hoc$
5:  else
6:      // find a candidate group to be swapped out
7:      $g_c \leftarrow null$
8:      for all $g' \in G$ do
9:          if $g'.mode = ad\_hoc$ AND $u(g') < u(g)$ AND swap($g$, $g'$) succeeds then
10:             if $g_c = null$ OR $u(g') < u(g_c)$ then
11:                 $g_c \leftarrow g'$
12:             end if
13:         end if
14:     end for
15:     if $g_c \neq null$ then
16:         release_bandwidth($g_c$)
17:         $g_c.mode \leftarrow cellular$
18:         reserve_bandwidth($g$)
19:         $g.mode \leftarrow ad\_hoc$
20:     end if
21: end if
```

in Algorithm 1, when a group g joins the network, we first compute the multicast tree, the required bandwidth vector, and the utility. The *admit* function is called to verify whether the ad hoc network has enough capacity to accept g. If so, g reserves bandwidth and switches to ad hoc mode. Otherwise, we try to find an appropriate candidate group g' currently in the ad hoc network and swap it with g. Group g' should satisfy the following conditions (Lines 9 and 10): 1) g' should have a smaller utility than g; 2) if g' releases its reserved bandwidth, g should be able to be admitted by the ad hoc network; 3) if there are more than one such candidates, the final g' should have the minimum utility. The *swap* function in Line 9 checks the second condition. After the best candidate g' is found, g' releases its resource and switches to cellular mode, whereas group g reserves bandwidth and uses ad hoc mode for multicast communication.

Similarly, in Algorithm 2 when a group g with ad hoc mode leaves, its reserved bandwidth is first released. Then we try to switch a candidate group from cellular mode to ad hoc mode to reduce the work load at the base station. As shown in Line 5, only those groups that pass admission control become candidates, and the one with the maximum utility is finally selected to switch to ad hoc mode.

Note that in both the join and leave sub-algorithms, the number of candidate groups to be swapped in and out is restricted to be one. Alternatively, multiple candidate groups whose union maximizes the utility function can be selected. However, this approach induces significantly higher computational overhead due to combinatorial explosion. As a result, in this work, we adopt the former method which chooses a single candidate group. But we do expect the latter approach with multiple-group selection yields better results. Thus, there is indeed a trade-off between computation overhead and performance. We would investigate this in our future study.

Algorithm 2 Leave(g)

```
1:  release_bandwidth($g$)
2:  // find a candidate group to be swapped in
3:  $g_c \leftarrow null$
4:  for all $g' \in G$ do
5:      if $g'.mode = cellular$ AND admit($g'$) then
6:          if $g_c = null$ OR $u(g') > u(g_c)$ then
7:              $g_c \leftarrow g'$
8:          end if
9:      end if
10: end for
11: if $g_c \neq null$ then
12:     reserve_bandwidth($g_c$)
13:     $g_c.mode \leftarrow ad\_hoc$
14: end if
```

Complexity Analysis In the join sub-algorithm, the complexity of multicast tree computation depends on the specific multicast routing algorithm; thus, we do not include it in the complexity analysis. In the $admit$ function, computing the bandwidth requirement for each on-tree link takes at most $O(m)$ time (in the worst case, a link may interfere with all the nodes in the network). Consequently, the time for bandwidth computation and admission control of the new group is bounded by $O(mn)$, since there are no more than n links in the tree (or mesh). If the admission fails, the algorithm tries to find a best candidate group. For each existing group g' in ad hoc mode, given their bandwidth vectors and utilities, the utility comparison takes constant time, and the $swap$ function takes $O(m)$ time. Therefore, the time complexity for finding a candidate tree is at most $O(mn_g)$. If a candidate group is obtained, both $release\_bandwidth$ and $reserve\_bandwidth$ only take $O(m)$ time. Summing up these computation requirements, we conclude that the time complexity of the join algorithm is $O(mn + mn_g)$.

In the leave sub-algorithm, the most time-consuming step is to find the best group to be switched into the ad hoc network. Since admission control for each group takes $O(mn)$ time (as explained earlier) and the total number of candidate groups is bounded by $O(n_g)$, the time complexity for this algorithm is $O(mnn_g)$.

Member Dynamics In the above dynamic algorithm, we basically consider the scenarios when group members join and leave a group at the same time. In reality, members may join or leave separately. To be more useful, the above dynamic algorithm needs to be extended to address this member dynamic issue. For example, when a member joins a group currently in ad hoc mode, if there is insufficient capacity to admit this member, we have the following two choices to comply with the capacity constraint. If a candidate group is found, the candidate group is switched to cellular mode to release bandwidth required by the new member and the new member can be admitted into ad hoc mode. Otherwise, this group itself is switched to cellular mode[7]. Similarly, when a member leaves a group, we can try to switch a candidate group to cellular mode to take advantage of the released bandwidth.

4.3 A Naive Dynamic Algorithm

Here we present a naive algorithm, which can be used as a reference to evaluate the performance of the proposed dynamic algorithm. It works as follows. When a new group joins, if the capacity of the ad hoc network allows, admit this group and reserve bandwidth for it. Otherwise, the group is put in cellular mode. When a group in ad hoc mode leaves, simply release its bandwidth. In essence, this naive approach is a simplified version of the proposed dynamic algorithm: it neither swaps groups in different modes nor differentiates groups based on their utilities.

Complexity Analysis In this naive dynamic algorithm, when a group joins the network, it takes $O(mn)$ time to determine whether it can be accepted or not, and there is no need to find candidate group. On the other hand, when a group leaves, it takes $O(m)$ time in the worst case for the group to release bandwidth on all the nodes in the network.

4.4 Implementation Issues

To implement the dynamic algorithm in a distributed way, each node needs to maintain the following information: 1) for every multicast group that reserves bandwidth at this node, the amount of reserved bandwidth and the utility; 2) the total amount of unused bandwidth at this node. For simplicity, we assume each multicast group has a coordinator (e.g., source or initiator).

Group Join When a new group comes, it first uses an ad hoc multicast routing protocol to set up a multicast tree. After the tree is established, the group coordinator instructs the multicast tree nodes to reserve bandwidth for the corresponding links in the tree. Bandwidth reservation can be done locally, since every link only involves two nodes and interferes with their one-hop neighbors.

If some nodes do not have enough bandwidth, they will report to the group coordinator the information of the groups that reserve bandwidth on these nodes (such as utilities and reserved bandwidth). After collecting this information, the group coordinator tries to find if there are any candidate groups which satisfy the following conditions: 1) at each node where the reservation for the new group fails, the candidate group reserves bandwidth, and the sum of the bandwidth reserved by this candidate group and the unused bandwidth at this node is no less than the bandwidth desired by the new group; 2) the candidate group has a lower utility than the new group. The first condition guarantees that the new group can be accepted after the candidate group switches to cellular mode, while the second condition makes sure that the "swap" will bring gain for the base station. In this way, the candidate group with the smallest utility is finally chosen to be swapped with the new group. In the case that no candidate group is found, this new group has to connect to the base station for data delivery.

Group Leave When a group leaves the network and releases its resource, the base station can notify the groups that it is currently serving to set up multicast trees using the ad hoc multicast routing protocol. If a group succeeds in reserving bandwidth, it can switch to ad hoc mode. To reduce the control overhead, the base station can randomly choose a small number of groups to notify.

Mobility Handling In the above discussion, we assume that the network is relatively static. However, when nodes move around, link breakage and bandwidth violation may occur. For instance, when two neighbor nodes move apart, the link between them is likely to break. Link breakage can be detected by periodic exchange of "live" messages among neighbors. To repair the disrupted path, the downstream node can use expanded-ring search to locate nearby on-tree nodes, perform admission control and reserve bandwidth for each affected group.

On the other hand, when a node moves into the communication range of other nodes, the traffic it carries may increase the interference level of its new neighbors and result in bandwidth violation. The presence of the newcomer or extra traffic can also be detected by the live message exchange or by channel monitoring. In order to recover from bandwidth violation, among the affected groups, those with the lowest utilities have to choose alternate paths that avoid the congested nodes. The alternate path discovery and establishment can be accomplished in a similar way to link breakage recovery.

In the case that some groups cannot find paths with enough bandwidth for either link breakage or bandwidth violation, these groups need to switch back to cellular mode and use the base station to relay traffic.

---

[7] In this paper, we only consider the case that all the members in a group are in the same mode: either cellular or ad hoc. It might be interesting to investigate the scenario where members are in "hybrid" mode: some members are in cellular mode, while others are in ad hoc mode. We leave this study for future work.

5 Performance Evaluation

In this section, we evaluate the performance of the proposed algorithms through simulations. We first describe the experiment settings for our evaluation, and then compare the optimal solution obtained from ILP and the results of the dynamic algorithm and the naive algorithm.

5.1 Simulation Settings

We implement the algorithms in C++ and use lp_solve Version 5.0 to solve the ILP formulation. In the simulations, mobile nodes are located in a cell of $500 \times 500\ m^2$. The communication range in the ad hoc network is $115\ m$. The total number of nodes in the cell is 120 unless otherwise specified. The channel capacity for each node is in the range of $(100, 500)$ units. Without loss of generality, each multicast group is assumed to require one unit of bandwidth. For multicast groups, we select some nodes uniformly at random as group members and one member as the source. The group size is uniformly distributed within the range of $(mean - 10, mean + 10)$, where the "mean" is varied from 20 to 100. We assume group arrivals follow a Poisson distribution, and group lifetime follows an exponential distribution. In steady state, there are 80 concurrently active groups in the network. The simulation time is 100 seconds, and we collect data every 5 seconds after the steady state is reached. For each experiment, the results are averaged over 20 sets of network topologies and group traces.

Recall that the objective of the proposed algorithms is to maximize the bandwidth savings for the base station. Since the required data rate for each group is the same, we use *the number of admitted members* to evaluate the effectiveness of the proposed algorithms. To gain some insight about the difference between these algorithms, we also present the results for *the number of admitted groups*.

5.2 Results and Analysis

5.2.1 Impact of Network Density

In the first set of experiments, we fix the average group size to be 20 and the channel capacity 500. We vary the total number of nodes from 40 to 120. Fig. 4 and 5 illustrate the number of admitted members and groups, respectively.

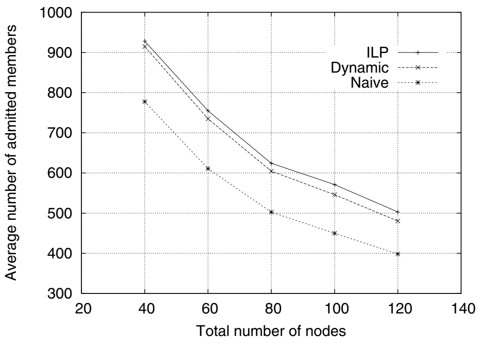


Figure 4. Average number of admitted members vs. total number of nodes.

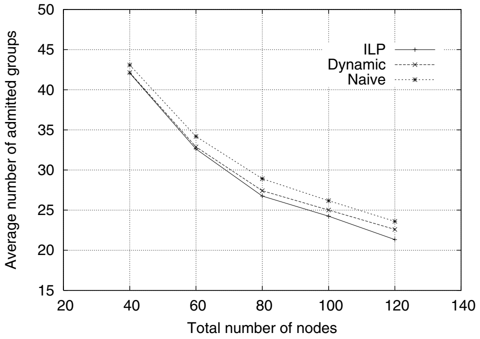


Figure 5. Average number of admitted groups vs. total number of nodes.

From Fig. 4, it is clear that the dynamic algorithm achieves very close performance to the ILP optimal solution (only approximately 3% worse). In contrast, the naive algorithm gives very poor results. On average, the dynamic algorithm accepts $20 - 30\%$ more members than the naive algorithm. The reason behind this is simple: the dynamic algorithm intelligently selects groups according to their utilities in order to maximize the benefit, whereas the naive algorithm blindly accepts groups according to the order they join the network. Nevertheless, due to the "greedy" group selection, the dynamic algorithm favors large groups and "neglects" small groups; thus, it cannot achieve the optimal solution in the cases when some small groups can fit in the network better than a smaller number of large groups.

Fig. 5 further demonstrates the effectiveness of the utility function: even though the naive algorithm accepts a larger number of groups than the dynamic algorithm, it does not distinguish between "good" and "bad" groups and therefore admits fewer group members.

Another trend in these two figures is that, as the number of nodes increases, fewer groups and members are accepted. This is due to the shared nature of the wireless channels: the higher network density, the more likely a wireless link will interfere with nearby nodes. Consequently, for networks with higher density, accepting one multicast group will reduce the bandwidth of a larger number of nodes, and a smaller number of groups can be accepted.

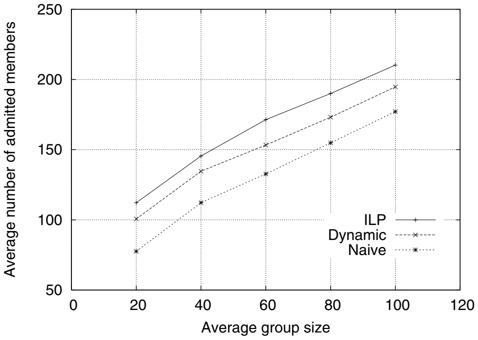


Figure 6. Average number of admitted members vs. group size.

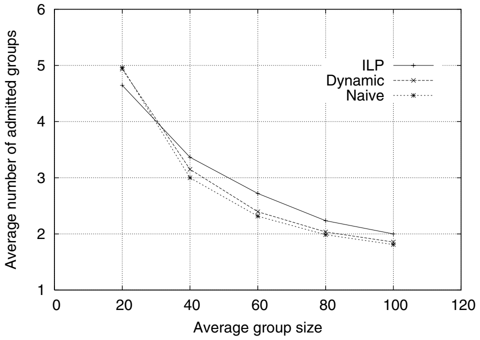


Figure 7. Average number of admitted groups vs. group size.

5.2.2 Impact of Group Size

To investigate the effect of average group size on algorithm performance, we increase group size from 20 to 100 in this set of simulations. The node capacity is fixed at 100, since ILP requires very long computation time for large groups when capacity is high. We plot the results in Fig. 6 and 7.

Again, we observe that the dynamic algorithm outperforms the naive algorithm by a large margin in terms of the number of accepted members. However, as group size increases, the difference between ILP and dynamic tends to slightly enlarge, attributed to the fact that it is more difficult for larger groups to "squeeze" into the ad hoc network, and thus the greedy nature of the dynamic algorithm leads to higher performance penalty for large groups.

Comparing these two figures, the ILP solution and the dynamic algorithm admit more groups and considerably more members than the naive algorithm, which means that they can select a better set of groups to "pack" tightly in the network and on average they select larger groups. We also find out that as group size increases, fewer groups but more members are accepted. Intuitively, the larger the group size, the higher the multicast efficiency. Hence, for the same number of members, larger groups consume less bandwidth. Equivalently, with the same amount of bandwidth, more members can be accepted when group size is bigger.

5.2.3 Impact of Channel Capacity

Finally, we study how channel capacity affects the algorithms. We vary the capacity from 100 to 500. The number of nodes is 120 and the average group size is 20. As depicted in Fig. 8 and 9, the number of admitted groups and members increase linearly with channel capacity, which is consistent with our intuition. Similar to previous figures, the dynamic algorithm achieves near optimal solution while the naive algorithm performs approximately 10% worse than the dynamic algorithm. In addition, the gap between the dynamic and naive algorithms widens for admitted members with capacity increase, indicating the advantage of the dynamic algorithm in high-capacity networks. Furthermore, we observe that the naive algorithm again fails to select good groups because it accepts more groups but fewer members than the other two approaches.

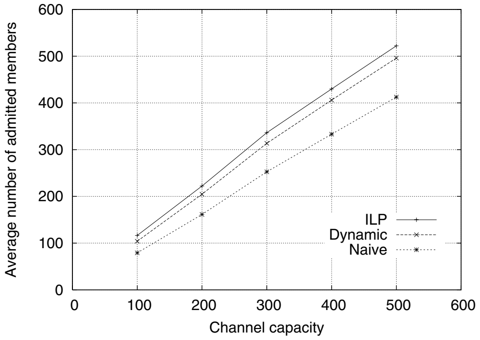


Figure 8. Average number of admitted members vs. channel capacity.

5.3 Summary

Our observations through the simulation experiments can be summarized as below. First of all, ILP always gives optimal solutions, but it becomes time consuming for large groups and high capacities. It is also not appropriate for dynamic on-line systems, since re-computation is required whenever group dynamics occur. Second, the dynamic algorithm achieves near-optimal performance, whereas the naive algorithm gives much worse results. The dynamic algorithm wins by distinguishing between "good" and "bad" groups using a utility function: it either selects more groups because it favors groups which consume less bandwidth, or selects larger groups since larger groups tend to optimize

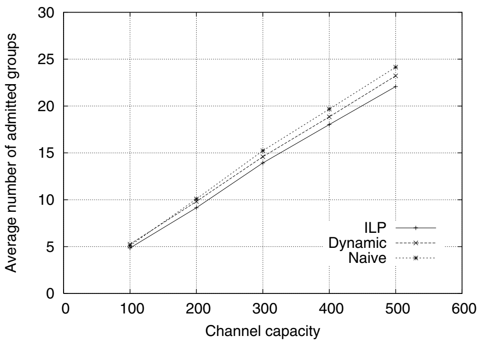


Figure 9. Average number of admitted groups vs. channel capacity.

the objective function. Third, more bandwidth can be saved when the channel capacity is higher, the number of nodes is smaller, and the average group size is larger.

6 Conclusions

In this paper, we have studied the multicast group selection problem in integrated cellular and ad hoc networks. Our contributions can be outlined as follows. We first develop a simple while effective model for computing the bandwidth requirement of a multicast group in wireless ad hoc networks, which considers wireless interference issues. Then we formulate the group selection problem as a multidimensional knapsack problem, and propose an ILP formulation and a utility-based dynamic algorithm with polynomial-time complexity to solve this problem. In addition, we examine how the dynamic algorithm can be implemented in a distributed way and discuss node mobility issues. Finally, through extensive simulations, we find that the dynamic algorithm can achieve near-optimal solutions in most scenarios, and it is more appropriate for on-line systems than the ILP approach.

Future Work We plan to implement the distributed dynamic algorithm in real network scenarios, further checking the feasibility and performance of our approach.

References

[1] CDMA2000 high rate broadcast-multicast packet data air interface specification. Feb. 2004.

[2] G. N. Aggelou and R. Tafazolli. On the relaying capability of next-generation GSM cellular networks. *IEEE Personal Communications*, 8(1):40–47, Feb. 2001.

[3] A. Ballardie. Core Based Trees (CBT version 2) multicast routing: protocol specification. *IETF RFC 2189*, Sept. 1997.

[4] E. Bommaiah, M. Liu, A. McAuley, and R. Talpade. AMRoute: Ad-hoc multicast routing protocol. *Internet Draft, draft-talpade-manet-amroute-00.txt*, Aug. 1998.

[5] S. De, O. Tonguz, H. Wu, and C. Qiao. Integrated cellular and ad hoc relay (iCAR) systems: Pushing the performance limits of conventional wireless networks. In *The 35th Hawaii International Conference on System Sciences (HICSS)*, Jan. 2002.

[6] S. Deering, D. Estrin, D. Farinacci, V. Jacobson, C. Liu, and L. Wei. The PIM architecture for wide-area multicast routing. *IEEE/ACM Transactions on Networking*, Apr. 1996.

[7] J. J. Garcia-Luna-Aceves and E. L. Madruga. The core-assisted mesh protocol. *IEEE Journal on Selected Areas in Communications*, 17(8):1380–1294, Aug. 1999.

[8] M. Hauge, A. Hafslund, F. Y. Li, and O. Kure. Multicast-service distribution on a cellular network assisted by local ad hoc networks. In *The Third Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net)*, June 2004.

[9] M. Hauge and O. Kure. Multicast service availability in a hybrid 3G-cellular and ad hoc network. In *International Workshop on Wireless Ad-hoc Networks (IWWAN)*, May 2004.

[10] H.-Y. Hsieh and R. Sivakumar. Performance comparison of cellular and multi-hop wireless networks: A quantitative study. In *Proceedings of ACM SIGMETRICS*, June 2001.

[11] H.-Y. Hsieh and R. Sivakumar. On using the ad-hoc network model in cellular packet data networks. In *ACM MOBIHOC*, June 2002.

[12] S.-J. Lee, W. Su, and M. Gerla. On-demand multicast routing protocol (ODMRP for ad hoc networks). *Internet Draft, draft-ietf-manet-odmrp-01.txt*, June 1999.

[13] Y.-D. Lin and Y.-C. Hsu. Multihop cellular: A new architecture for wireless communications. In *Proceedings of IEEE INFOCOM*, Mar. 2000.

[14] H. Luo, R. Ramjee, P. Sinha, L. Li, and S. Lu. UCAN: A unified cellular and ad-hoc network architecture. In *ACM MobiCom*, Sept. 2003.

[15] J. Moy. Multicast routing extensions to OSPF. *RFC 1584*, Mar. 1994.

[16] J. C. Park and S. K. Kasera. Enhancing cellular multicast performance using ad hoc networks. In *IEEE Wireless Communications and Networking Converence (WCNC)*, Mar. 2005.

[17] C. Partridge, D. Waitzman, and S. Deering. Distance Vector Multicast Routing Protocol. *RFC 1075*, 1988.

[18] C. W. Wu, Y. C. Tay, and C.-K. Toh. Ad hoc multicast routing protocol utilizing increasing id-numbers (amris) functional specification. *Internet Draft, draft-ietf-manet-amris-spec-00.txt*, Nov. 1998.

[19] X. Wu, S.-H. G. Chan, and B. Mukherjee. MADF: A novel approach to add an ad-hoc overlay on a fixed cellular infrastructure. In *IEEE Wireless Communications and Networking Converence (WCNC)*, Sept. 2000.

[20] Q. Xue and A. Ganz. Ad hoc QoS on-demand routing (AQOR) in mobile ad hoc networks. *Journal of Parallel and Distributed Computing*, 63(2):154–165, Feb. 2003.

[21] D. Zhu, M. W. Mutka, and Z. Cen. QoS aware wireless bandwidth aggregation (QAWBA) by integrating cellular and ad-hoc networks. In *The First IEEE International Conference on Quality of Service in Heterogeneous Wired/Wireless Networks (QShine)*, Oct. 2004.