Efficient Location-Based Decision Supporting Content Distribution to Mobile Groups

Mhameed Aezladen   Reuven Cohen   Danny Raz
Department of Computer Science
Technion
Israel

***Abstract*— The paper deals with efficient location-based decision supporting content distribution to mobile groups. We consider the case where a set of Information Dissemination Devices (IDDs) broadcast a limited amount of location-based information to passing mobile nodes that are moving along well-defined paths. We develop a novel model that captures the main aspects of the problem, and define a new optimization problem we call MBMAP (Maximum Benefit Message Assignment Problem). We study several variants of this problem in the case where the IDDs are cooperative and in the case where they are not. We develop new approximation algorithms for these variants and then focus on the practical effects of using them in realistic networking scenarios.**

I. Introduction

With the advance of mobile communication technologies, many new applications depend on the ability of the network to deliver location-based information to the mobile nodes in real time. Such applications can be found in the context of Intelligent Transportation Systems (ITSs), network centric operations (NCOs) and cellular networks. This work proposes a mathematical model and algorithms to determine how to distribute content in such networks.

The model we consider for location-based decision supporting content distribution to mobile groups can be characterized as follows:

(C1) **Infrastructure to mobile:** Information is transmitted from static IDDs to the mobile nodes. The mobile nodes do not pass information to each other, as they do, for example, in [24].

(C2) **Location dependent multicast:** The same information is multicast by an IDD to all the nodes in its vicinity. Moreover, the same information can be delivered to a mobile node by several different IDDs. However, the mobile node benefits from this information only once.

(C3) **Swarm mobility:** Instead of assuming random mobility, as in [4] and many other papers, we follow recent studies that indicate predictable mobility in mobile applications [15], [22], which can be used to improve communication protocol performance. To capture this property, we assume that a mobility pattern is defined by flows as proposed by [26]. Namely, each mobile device belongs to one or more flows, and all the nodes of the same flow use the same path.

In the considered model, every flow is a group of mobile nodes moving along the same path. A set of static Information Dissemination Devices (IDDs) is distributed throughout the network. Each IDD can deliver location-based content to the mobile nodes in the flows it is traversed by. A benefit function determines the benefit $B(f, m, i)$ a flow f can obtain from a message m from IDD i. This benefit depends on many factors, such as the volume of f, the location of i, and the content of m. The problem is to determine what information each IDD should broadcast. This new optimization problem is referred to as MBMAP (Maximum Benefit Message Assignment Problem).

MBMAP can be solved with or without cooperation among the IDDs. When there is no cooperation, every IDD makes a local decision regarding the most important information to broadcast to the flows. This version of MBMAP is referred to as l-MBMAP (local MBMAP), and its most important property is that no communication infrastructure is needed between the IDDs. This property is important, for instance, when the IDDs are sensors in a sensor network.

When cooperation between the IDDs is possible, a global decision can be made while taking into account the fact that flows pass through several IDDs. This version of MBMAP is referred to as g-MBMAP (global MBMAP). It is easy to see that the best solution for l-MBMAP can never be better than the best solution for g-MBMAP. However, the improved performance of g-MBMAP comes at the cost of coordinating the broadcast of different IDDs, which requires a centralized management entity and a communication infrastructure that connects the IDDs.

The rest of the paper is organized as follows. In Section II we present application scenarios for the models and problems considered in the paper and discuss related work. In Section III we formally define l-MBMAP and g-MBMAP and discuss their computational complexity. In Section IV we present an algorithm that achieves a constant factor approximation for g-MBMAP, and analyze its performance and running time complexity. In Section V we extend MBMAP to the case where the different messages are correlated. In this case, we also distinguish between two variants: l-E-MBMAP (local extended MBMAP) and g-E-MBMAP (global extended MBMAP). In Section VI we present a simulation study for the

This is a journal version of a paper presented in Infocom 2009 under the title: "Locally vs. Globally Optimized Flow-Based Content Distribution to Mobile Nodes."

various models discussed throughout the paper. In Section VII we discuss several approaches for defining flows. Finally, Section VIII concludes the paper.

II. Application Scenarios and Related Work

A. Application Scenarios

We now describe two application scenarios for the considered model and problem. The first is in the context of an Intelligent Transportation System (ITS) [13], [25]: a collection of technologies intended to make surface transportation safer and more efficient. Such systems are often divided into two communication classes: vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I). The application we describe is related to V2I. Consider a set of electronic road signs, located every few miles along the highway. Each sign posts a small number of short messages, relevant to the passing mobile nodes, on traffic jams, closed exits, and so forth. Future intelligent transportation systems, which use V2I communication, are likely to replace these electronic signs with IDDs that broadcast information to the passing mobile nodes [13], [25]. This information will be displayed on the dashboard, using any language chosen by the driver, and will be available to the drivers for a much longer time than the information posted on the electronic signs. Regardless of the broadcast capabilities of an IDD, the information it delivers to the passing mobile nodes should be minimized for two main reasons: the limited space on the mobile node screen, and the desire not to distract drivers with nonessential information. The problem of determining what message(s) every road IDD should broadcast is exactly MBMAP.

In the application described above, principle (C1) from Section I clearly holds. Location dependent multicast (C2) is also part of this application because all the vehicles of the same flow (group) are supposed to get the same information about traffic, weather and road conditions from each IDD they pass by. Finally, (C3) holds because all the vehicles moving along the same highway can be considered as a single flow that goes through the same set of IDDs. This approach does not require information about the exact route taken by each car, but only approximated statistics about the average volume of traffic that uses each highway during each hour and the percentage of the traffic that leaves the highway at each exit. When more detailed information is available, better optimization can be obtained.

The above application is relevant both for l-MBMAP and for g-MBMAP. If the electronic signs (IDDs) are connected to a backbone, they can share information and make global decisions. But the IDDs might be self-contained sensors that obtain local information by sensing the passing traffic, in which case the decision about what to broadcast must be local.

The second application scenario for the considered model is in the context of network-centric operations (NCO) [9], [10]. NCO is a theory that uses networking to improve both the efficiency and effectiveness of military operations. While there is no single NCO architecture, in a typical system there might be hundreds of radar stations and sensors that gather tactical information. This information is then processed, centrally or distributively, and delivered to thousands of mobile nodes (soldiers, vehicles, tanks, etc.) [9]. Due to reliability and simplicity considerations, such systems usually use only infrastructure to mobile communication. Information is usually delivered according to the nodes' geographical location (C2). For instance, a high resolution video stream of a certain area is mainly relevant to the forces that are close to this area. Finally, because a military force usually consists of many individual nodes that move along the same route, the group mobility characteristic (C3) holds as well.

It seems that for this application l-MBMAP is more relevant than g-MBMAP, mainly because it is hard to predict the route to be taken by every flow. Thus, each IDD (sensor/radar) is more likely to make a local decision about what information to distribute, even if backbone communication between the IDDs is available.

B. Related Work

We start by discussing earlier work related to the mobility model considered in this paper. Our model is similar to the "virtual track" model considered by [26]. Another approach proposed in the past for modeling the mobility of nodes is the Random Wapiti (RWP) model [4], where each node selects a random speed and moves to a random destination. After reaching its destination, the node remains idle for some time and then repeats the same process.

A model for group mobility, called RPGM, is presented in [15]. In this model, each node in a group has two components in its movement vector. The first is an individual component, which is based on RWP. The second is a group component, which is shared by all nodes in the same group and is also based on RWP.

There are many applications of wireless communications in vehicular networks; the authors of [5] provide a broad overview of these. However, most research efforts have focused on using wireless communication technology for enhancing the safety and efficiency of urban and highway traffic.

Broadcast-based communication schemes for mobile nodes are presented in [19], [23]. These schemes implement intelligent rules to reduce the number of redundant transmissions and increase broadcast reliability. In [23], mechanisms for priority access to reduce redundant retransmissions are discussed. In [19], the forwarding task is assigned only to one mobile node in every dissemination direction. Yet these approaches do not guarantee full coverage and or a bounded delay.

Cyclic Data Broadcast (CDB), also known as broadcast disks or data carousel, is a well-accepted approach for reliable broadcast for unidirectional channels. The idea is that the broadcasting node broadcasts the required data items in a cyclic manner, according to some predetermined program. Several models have been proposed in this context [1], [17], [20].

Recall that in this paper we solve the problem of deciding what data items should be broadcast by every broadcasting device (IDD). To some extent this problem is related to the problem solved in [7]. However, the main difference is that here we consider many (hundreds or even thousands) broadcasting nodes, as opposed to one broadcasting node considered in [7]. The large number of broadcasting nodes complicates the decision because what data each node should broadcast depends on the data to be broadcast by the other nodes.

III. MBMAP and Its Computational Complexity

We are given a triplet $(F,\ I,\ M)$, where F is a set of flows, I is a set of IDDs and M is a set of messages. For a message $m \in M$, $size(m)$ denotes the size of m, and $F(m) \subseteq F$ denotes the set of flows for which this message is relevant. For an IDD $i \in I$, $size(i)$ is the capacity of i, namely the amount of data the device can deliver to nearby mobile nodes while they are within its transmission range. $B(f, m, i)$ indicates the benefit flow f obtains from receiving message m from IDD i. Thus, $f \in F(m)$ if and only if there exists an i such that $B(f, m, i) > 0$. For an IDD i, $F(i)$ indicates the set of flows that can benefit from assigning a message to i, because each of these flows are within the transmission range of IDD i, and $B(m, i)$ is the benefit of assigning a message m to IDD i. Thus,

$$B(m, i) = \sum_{f \in\ F(i)} B(f, m, i). \tag{1}$$

Let T be an assignment of messages to IDDs. We say that $(m, i) \in T$ if T contains an assignment of message m to IDD i. Of course, the same message can be assigned to multiple IDDs. Denote by $T(i)$ the subset of messages that are assigned to i by T. We say that assignment T is legal if and only if for every i, $\sum_{m \in T(i)} size(m) \leq size(i)$ holds.

Consider the case where a message m is assigned to two different IDDs, i_1 and i_2, and these two IDDs are traversed by the same flow f. If we ignore the fact that f is exposed to the same message twice, then the total benefit obtained by f from m is $B(f, m, i_1) + B(f, m, i_2)$. The solution for this model is trivial, because there is no dependency between different IDDs. Thus, optimization is obtained by running the Knapsack algorithm for every IDD. However, for the application scenarios we described earlier, adding the benefit from the same message multiple times does not make a lot of sense. Throughout this paper we consider a more reasonable and difficult model, where the benefit is $\max\{B(f, m, i_1), B(f, m, i_2)\}$. The rationale for the $\max$ model is simple: if a flow receives the same message multiple times from different IDDs, then it acquires each time a residual benefit which is upper bounded by the maximum of all benefits. For example, if the benefit from receiving the message m from IDDs i_1, i_2, i_3 and i_4 is 3, 5, 8 and 6 respectively, then a flow that goes through these 4 IDDs in the same order will acquire 3 from i_1, 5-3=2 from i_2, 8-5=3 from i_3, and 0 from i_4. The total benefit in this case is 3+2+3=8=$\max 3, 5, 8, 6$.

Figure 1 shows a simple example of a network with two IDDs (i_1 and i_2), four flows (f_1, f_2, f_3 and f_4) and two possible messages (m_1 and m_2) that are related to L_1 and L_2 respectively. For this example, suppose that $B(f_1, m_1, i_1) = 1$, $B(f_2, m_2, i_1) = 4$, $B(f_2, m_2, i_2) = 6$, $B(f_3, m_2, i_2) = 2$ and $B(f_4, m_1, i_2) = 5$. For the rest of the assignments, the benefit is 0. The benefit of a particular message for each flow is determined by the importance of the message content to

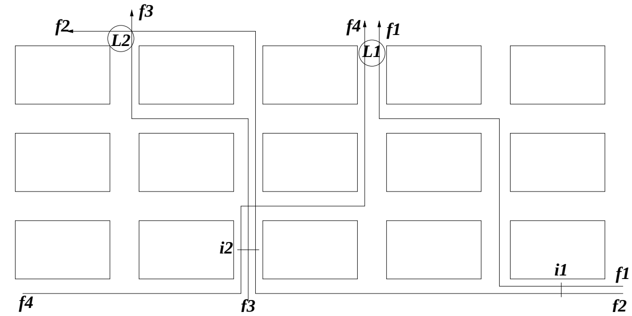


Fig. 1. A simple scenario: m_1 and m_2 are messages indicating traffic jams in L_1 and L_2 respectively

the mobile nodes comprising the flow and by the size of the flow (in terms of mobile nodes per second). For instance, in the example above, f_2 obtains no benefit from m_1 because it is not affected by the traffic jams in L_1. In addition, as shown above, the same flow may obtain different benefits from receiving the same message in different places.

We start our discussion with l-MBMAP, where we seek to maximize the local benefit obtained for each IDD, and we do not take into consideration the dependency between IDDs that are visible to the same flow. In this model, the benefit of a legal assignment T is:

$$B_L(T) = \sum_{(m,i) \in T} B(m, i) = \sum_{(m,i) \in T} \sum_{f \in F(i)} B(f, m, i)$$

$$= \sum_{i \in I} \sum_{m \in T(i), f \in F(i)} B(f, m, i).$$

The resulting optimization problem is:

$$\text{maximize} \qquad B_L(T),$$

$$\text{subject to:} \quad \sum_{m \in T(i)} size(m) \leq size(i) \quad \text{for every i.}$$

Considering the computational complexity of the problem, we can distinguish between two cases: the case where all messages have the same size, and the one where different messages might have different sizes. In the former case, an optimal solution can be found in polynomial time using a simple greedy algorithm that assigns the most profitable messages to each device. In the latter case, the problem is equivalent to the well-known NP-complete Knapsack problem. The greedy algorithm is only a 2-approximation solution, but there exist algorithms that find the optimal solution in pseudo-polynomial time [12]. Moreover, this problem is known to have an FPTAS[1] [16].

We now focus on the global version of the problem, g-MBMAP, where the assignment of messages to IDDs is performed while taking into account the fact that flows are likely to traverse many IDDs. Thus, the benefit obtained from assigning a message to an IDD depends on whether the

[1]An FPTAS is an algorithm which given a Knapsack instance and any ϵ, returns, in polynomial time, a solution that is within (1 - ϵ) of the optimum.

same message is also assigned to other IDDs. Specifically, we assume that if a message m is assigned to two IDDs, i and i', and a flow f passes in the transmission range of i and i', then the benefit obtained by f from these assignments is the greater of the two. Hence,

$$B(\{(m,i)\} \cup \{(m,i')\}) = \sum_{f \in F(i) \setminus F(i')} B(f,m,i) + \sum_{f \in F(i') \setminus F(i)} B(f,m,i') + \sum_{f \in F(i) \cap F(i')} \max\{B(f,m,i), B(f,m,i')\}.$$

Under this model, the benefit obtained by a flow f from a message m in a legal assignment T is:

$$B(T,f,m) = \max_{(m,i) \in T} \{B(f,m,i)\},$$

and the total benefit of assignment T is:

$$B_G(T) = \sum_{f \in F} \sum_{m} B(T,f,m). \quad (2)$$

The optimization problem in this case is:

$$\text{maximize} \quad B_G(T),$$

$$\text{subject to:} \quad \sum_{m \in T(i)} size(m) \le size(i) \quad \text{for every i.}$$

Consider again the example of Figure 1, and assume that $size(i_1) = size(i_2) = 1$. For the assignment $\{(m_1,i_1),(m_1,i_2)\}$, the total benefit is $1+5=6$. For the assignment $\{(m_1,i_1),(m_2,i_2)\}$, the total benefit is $1+6+2=9$. For the assignment $\{(m_2,i_1),(m_1,i_2)\}$, the total benefit is $4+5=9$, and for the assignment $\{(m_2,i_1),(m_1,i_2)\}$, the total benefit is $2+\max\{4,6\}=8$.

IV. An Approximation Algorithm for g-MBMAP

A. On the computational complexity of g-MBMAP

We now show that g-MBMAP is not only NP-Hard, but also cannot be approximated within a factor better than $\frac{e}{e-1}$. We show this by a reduction from the well-know Maximum Coverage Problem (MCP) [14], defined as follows: Given a collection of subsets $L = \{S_1 ... S_m\}$ of the universal set $U = \{1, ..., n\}$, and a positive integer p, find a subset $H \subseteq L$ such that $|H| = p$ and the number of covered elements $| \cup_{h \in H} h|$ is maximum.

Theorem 1: g-MBMAP cannot be approximated within a factor better than $\frac{e}{e-1}$, even if all messages are of fixed-size, unless $NP \subseteq DTIME(n^{\log\log n})$.

Proof: It is shown in [11] that MCP cannot be approximated within a factor better than $\frac{e}{e-1}$ unless $NP \subseteq DTIME(n^{\log\log n})$. We now show how to convert an instance of MCP into an instance of g-MBMAP. Given an instance of MCP, we define the set of flows in g-MBMAP to be $U = \{1 \dots n\}$ and the set of IDDs to be $\{i_j\}_{j=1}^{p}$. For every i_j, we set $size(i_j) = 1$ and $F(i_j) = U$. This implies that each device can carry exactly one message, which will be visible to all flows. In addition, for every subset $S_i \in L$ we define a message m_i such that $F(m_i) = S_i$. Finally, we set $B(f,m,i) = 1$ for every flow f, message m and device i. Observe that under this setting $B(T) = |\bigcup_{(m,i) \in T} F(m))|$. It is easy to see that an optimal solution for the MCP instance is translated into an optimal solution for the g-MBMAP instance and vice versa. Since the transformation is size preserving and can be performed in linear time, no approximation algorithm for g-MBMAP can do better than the bound for MCP. ∎

B. Our new approximation algorithm

At first glance, it seems that g-MBMAP is similar to the Generalized Assignment Problem (GAP) [8]. The input for GAP is a set of bins (knapsacks) and a set of items. Each bin has a limited capacity, and for each item i and bin j, $s(i,j)$ and $p(i,j)$ indicate the size and benefit of item i in bin j. The objective is to find the subset of items to be assigned to each bin such that the overall benefit is maximized. However, there are two important differences between g-MBMAP and GAP:

- In GAP every item can be selected only for one bin, whereas in g-MBMAP an item (a message) can be selected for multiple bins (IDDs).
- In GAP the benefit associated with the selection of an item for a bin is independent of the selection made for other bins. In contrast, in g-MBMAP there is a strong correlation between assignments. As explained before, if a message is selected for multiple IDDs and the same flow passes through some of them, the benefit this flow obtains from this message is not equal to the sum of the benefits, but to the maximum benefit.

We solve g-MBMAP using a technique similar to the one presented in [8] for GAP. That is, given an α-approximation algorithm ALG for the Knapsack problem, we build a $(1+\alpha)$-approximation algorithm for g-MBMAP using the concept of Local Ratio [3]. In our case we need a small generalization of the Local Ratio Theorem presented in Theorem 9 of [3]. This theorem uses a benefit function of the type $w \cdot x$, where $w = w_1 + w_2$. Our benefit function cannot be presented in this way and thus we restate the theorem to make it slightly more general. The proof, however, is very similar to the proof of Theorem 9 in [3] and is not repeated here.

The generalized version of the Local Ratio Theorem is as follows. Let F be a set of constraints and let B, B_1, and B_2 be benefit functions as defined in Equation 2 on a set of vectors X, such that for every $x \in X$, $B(x) = B_1(x) + B_2(x)$ holds. If x is an r-approximate solution with respect to (F, B_1) and with respect to (F, B_2), then it is also an r-approximate solution with respect to (F, B).

We now present an algorithm for g-MBMAP. Given the benefit function $B(f,m,i)$ for every flow f, a message m and a device i, the algorithm returns a legal assignment T that is $(1+\alpha)$-approximation to the optimal solution.

Algorithm 1: (an approximation algorithm for g-MBMAP) Denote by $B_k(f,m,i)$ the benefit for flow f from assigning message m to device i at the k^{th} iteration of the algorithm. Initially, for every message m, device i and flow f, set $B_1(f,m,i) \leftarrow B(f,m,i)$. Then, for $k=1$ to $|I|$ do:

1) Run algorithm ALG for the Knapsack problem on the following instance. The knapsack size is $size(i_k)$. The

items to be packed are the set of messages M. The benefit for every $m \in M$ is $\sum_{f \in F} B_k(f, m, i_k)$, and the weight for every $m \in M$ is $size(m)$. Let N_k be the set of messages selected by ALG.

2) If $k = |I|$ return $T = \bigcup_{j=1}^{|I|} \{N_j \times \{i_j\}\}$.
3) For every message m, device i, and flow $f \in F$:
   - Decompose the benefit function $B_k(f, m, i)$ into two functions $B_k^1(f, m, i)$ and $B_k^2(f, m, i)$ such that the following holds:
     a) $B_k^1(f, m, i) = \begin{cases} B_k(f, m, i) & \text{for } i = i_k \\ \min\{B_k(f, m, i), B_k(f, m, i_k)\} & \text{for } i \neq i_k \text{ and } m \in N_k \\ 0 & \text{otherwise} \end{cases}$
     b) $B_k^2(f, m, i) = B_k(f, m, i) - B_k^1(f, m, i)$.
   - Set $B_{k+1}(f, m, i) = B_k^2(f, m, i)$. ■

C. Execution Example

We now show an example of the execution of the algorithm. Let the initial benefit matrix be:

	f_1	f_2	f_3	f_4
$B(f, m_1, i_1)$	0	0	0	5
$B(f, m_1, i_2)$	1	0	0	0
$B(f, m_2, i_1)$	0	4	2	0
$B(f, m_2, i_2)$	0	6	0	5

where the size of all messages and IDDs is 1.

To simplify the presentation, for each device and message, we consider only flows with nonzero benefit. Consequently, the initial B matrix looks as follows:

B:

	m_1	m_2
i_1	$\{(f_4, 5)\}$	$\{(f_2, 4), (f_3, 2)\}$
i_2	$\{(f_1, 1)\}$	$\{(f_2, 6)\}$

We set B_1 to be equal to B:

B_1:

	m_1	m_2
i_1	$\{(f_4, 5)\}$	$\{(f_2, 4), (f_3, 2)\}$
i_2	$\{(f_1, 1)\}$	$\{(f_2, 6)\}$

Now, we build a Knapsack instance for i_1. The size of this knapsack is 1, the items are m_1 and m_2, the benefit of m_1 is 5 and of m_2 is 6 (4 for f_3 and 2 for f_2); the weight of every item is 1. We run a Knapsack algorithm to solve this instance. The output of this algorithm is m_2. Thus, we have $N_1 = \{m_2\}$.

Next, we decompose the benefit function B_1 into B_1^1 and B_1^2 as follows:

B_1^1:

	m_1	m_2
i_1	$\{(f_4, 5)\}$	$\{(f_2, 4), (f_3, 2)\}$
i_2	$\emptyset$	$\{(f_2, 4)\}$

B_1^2:

	m_1	m_2
i_1	$\emptyset$	$\emptyset$
i_2	$\{(f_1, 1)\}$	$\{(f_2, 2)\}$

Then, B_2 is set to be equal to B_1^2 (only nonempty rows are shown):

B^2:

	m_1	m_2
i_2	$\{(f_1, 1)\}$	$\{(f_2, 2)\}$

Flow f_2 has already gained a benefit of 4 from the assignment of m_2 to i_1. Therefore, the extra benefit for f_2 from the assignment of m_2 to i_2 is only $6 - 4 = 2$.

Again, we build and solve a Knapsack instance for i_2. We get $N_2 = \{m_2\}$, and by decomposing B_2 correspondingly we get:

B_2^1:

	m_1	m_2
i_2	$\{(f_1, 1)\}$	$\{(f_2, 2)\}$

B_2^2:

	m_1	m_2
i_2	$\emptyset$	$\emptyset$

The final assignment is, therefore, m_2 to i_1 and m_2 to i_2. The total benefit is 8: 2 of which is due to $B(f_3, m_2, i_1)$ and 6 due to $B(f_2, m_2, i_2)$.

D. A proof of worst-case performance guarantee

In order to use the Local Ratio Theorem, we first prove that $B_k(T)$ satisfies the required conditions.

Lemma 1: For each iteration of the algorithm, $k = 1$ to $|I|$, and for $T_k = \bigcup_{j=k}^{|I|} \{N_j \times \{i_j\}\}$, $B_k(T_k) = B_k^1(T_k) + B_k^2(T_k)$.

Proof:

$$B_k(T) = \sum_{f \in F} \sum_m \max_{(m,i) \in T} \{B_k(f, m, i)\}$$

$$= \sum_{f \in F} \sum_m \max_{(m,i) \in T} \{B_k^1(f, m, i) + B_k^2(f, m, i)\}.$$

Given a flow f, we now prove that for every message m:

$$\max_{(m,i) \in T} \{B_k^1(f, m, i) + B_k^2(f, m, i)\} =$$

$$\max_{(m,i) \in T} \{B_k^1(f, m, i)\} + \max_{(m,i) \in T} \{B_k^2(f, m, i)\}.$$

We distinguish between two cases:

1) The case where $(m, i_k) \in T$, i.e., m is assigned to i_k in T. Since $B_k(f, m, i) = B_k^1(f, m, i) + B_k^2(f, m, i)$, for every message m $\max_{(m,i) \in T} \{B_k^1(f, m, i)\} + \max_{(m,i) \in T} \{B_k^2(f, m, i)\} \geq \max_{(m,i) \in T} B_k(f, m, i)$. It remains to show that $\max_{(m,i) \in T} B_k(f, m, i) \geq \max_{(m,i) \in T} \{B_k^1(f, m, i)\} + \max_{(m,i) \in T} \{B_k^2(f, m, i)\}$. If $\max_{(m,i) \in T} B_k^2(f, m, i) = 0$, then we are done. Otherwise, $\max_{(m,i) \in T} B_k^2(f, m, i) = l > 0$. Suppose that the maximum holds for i_j $(\neq i_k)$, i.e.,

$B_k^2(f,m,i_j) = l$. Then, according to the construction, $B_k^1(f,m,i_j) = B_k^1(f,m,i_k) = \max_{(m,i)\in T} B_k^1(f,m,i)$. Thus, $\max_{(m,i)\in T}\{B_k(f,m,i)\} \geq B_k(f,m,i_j) = B_k^1(f,m,i_j) + B_k^2(f,m,i_j) = \max_{(m,i)\in T}\{B_k^1(f,m,i)\} + \max_{(m,i)\in T}\{B_k^2(f,m,i)\}$.

2) The case where $(m,i_k) \notin T$, i.e., m is not assigned to i_k in T. According to the construction, $\max_{(m,i)\in T}\{B_k^1(f,m,i)\} = 0$. This is because the only entry that does not equal to 0 is $B_k^1(f,m,i_k)$, but $(m,i_k) \notin T$. Thus we get also in this case that $\max_{(m,i)\in T}\{B_k^1(f,m,i) + B_k^2(f,m,i)\} = \max_{(m,i)\in T}\{B_k^1(f,m,i)\} + \max_{(m,i)\in T}\{B_k^2(f,m,i)\}$.

We conclude that for both cases the following holds:

$$B_k(T)$$
$$= \sum_{f\in F}\sum_{m} \max_{(m,i)\in T}\{B_k^1(f,m,i) + B_k^2(f,m,i)\} =$$
$$\sum_{f\in F}\sum_{m} \max_{(m,i)\in T}\{B_k^1(f,m,i)\} + \sum_{f\in F}\sum_{m} \max_{(m,i)\in T}\{B_k^2(f,m,i)\}$$
$$= B_k^1(T) + B_k^2(T).$$

■

Theorem 2: For every $1 \leq k \leq |I|$, the assignment $T_k = \bigcup_{j=k}^{|I|}\{N_j \times \{i_j\}\}$ is a $(1+\alpha)$-approximation with respect to $B_k(f,m,i)$.

Proof: We prove this by a reverse induction on the value of k, starting with $k = |I|$. The induction basis ($k = |I|$) follows from the validity of ALG, which produces an $\alpha-$approximation for the Knapsack problem.

For the inductive step, assume that T_{k+1} is a $(1+\alpha)$-approximation with respect to $B_{k+1}(f,m,i)$. We now prove in two steps that T_k is a $(1+\alpha)$-approximation with respect to $B_k(f,m,i)$. If T_k is a $(1+\alpha)$-approximation with respect to $B_k^2(f,m,i)$, and T_k is also a $(1+\alpha)$-approximation with respect to $B_k^1(f,m,i)$, then since by Lemma 1 $B_k(T_k) = B_k^1(T_k) + B_k^2(T_k)$, we can use the Local Ratio Theorem to prove Theorem 2. It is therefore left to prove that T_k is indeed a $(1+\alpha)$-approximation with respect to $B_k^1(f,m,i)$ and with respect to $B_k^2(f,m,i)$.

By construction, for every message m, device i and flow f, $B_{k+1}(f,m,i)$ is identical to $B_k^2(f,m,i)$. By the induction assumption, T_{k+1} is a $(1+\alpha)$-approximation with respect to $B_k^2(f,m,i)$. T_k contains all the assignments in T_{k+1}. Thus, T_k is a $(1+\alpha)$-approximation with respect to $B_k^2(f,m,i)$.

Now, we prove that T_k is a $(1+\alpha)$-approximation with respect to $B_k^1(f,m,i)$. The benefit function $B_k^1(f,m,i)$ has three components. The first component, c_1, is the benefit of assigning a message m to i_k, namely $\sum_{f\in F} B_k(f,m,i_k)$. The second component, c_2, is the benefit of messages selected to N_k whose benefit in $B_k^1(f,m,i)$ is set to be not larger than the benefit they obtain from their assignment to i_k. The third component, c_3, consists of all the remaining entries, which are set to 0. Algorithm ALG guarantees α-approximation for device i_k with respect to c_1. Therefore, the best solution with respect to c_1 would obtain at most $\alpha \cdot \sum_{m\in N_k}\sum_{f\in F} B_k^1(f,m,i)$. Any solution with respect to c_2 will gain at most $\sum_{m\in N_k}\sum_{f\in F} B_k^1(f,m,i)$, since the benefit of every entry in this component is set to be not larger than the benefit of $B_k^1(f,m,i_k)$. Note also that the benefit a flow obtains is at most the greatest between these benefits, no matter how many devices cover this flow. Finally, component c_3 has no contribution to the benefit. Thus, N_k is a $(1+\alpha)$-approximation with respect to $B_k^1(f,m,i)$. Since T_k contains all the assignments in N_k, it is a $(1+\alpha)$-approximation with respect to $B_k^1(f,m,i)$, which concludes the proof. ■

Corollary 1: Algorithm 1 is a $(1+\alpha)$-approximation for g-MBMAP.

Observe that when the messages are of equal size, we can replace algorithm ALG by an algorithm that finds an optimal assignment for each IDD. In this case $\alpha = 1$, and we obtain a 2-approximation for g-MBMAP.

Recall that ALG is an algorithm for solving the Knapsack problem. To analyze the running time complexity of Algorithm 1, assume that the running time of ALG is $g(|M|)$. In step 1 we build a knapsack instance. The running time of this construction is bounded by $|M| \cdot |F|$. In step 3 we construct the benefit functions. The number of relevant entries, namely nonzero values in $B_k^1(f,m,i)$ and updated values in $B_k^2(f,m,i)$, is upper bounded by $2 \cdot size(i_k) \cdot (|I|-k)$. Thus, the running time of this step for all of the $|I|$ iterations is

$$\sum_{k=1}^{|I|} 2 \cdot size(i_k) \cdot (|I|-k) \cdot |F| \leq 2 \cdot c_{max} \cdot \sum_{k=1}^{|I|} (|I|-k) \cdot |F|$$
$$= O(c_{max} \cdot |F| \cdot |I|^2),$$

where $c_{max} = \max_{i\in I} size(i)$. Consequently, the total running time of Algorithm 1 is:

$$O\left((g(|M|) + |M| \cdot |F|) \cdot |I| + c_{max} \cdot |F| \cdot |I|^2\right).$$

We can improve the algorithm's running time by making a minor change in step 3. Instead of calculating the whole benefit function $B(f,m,i)$, it is sufficient to calculate the value of the function for i_k. However, in this calculation one should consider the effect of all previous assignments. To this end, we define a new matrix Q, such that $Q[f,m]$ indicates the benefit already obtained by flow f from message m.

V. The Extended MBMAP

Our definition of MBMAP assumed no dependency between the benefit of different messages in the same or different IDDs. However, when such a dependency exists, it will certainly affect the optimal assignment. For example, consider the application scenario where the IDDs are electronic signs and the mobile nodes are vehicles. Let m_1 and m_2 be two messages posted in two consecutive IDDs, i_1 and i_2. Suppose that m_1 informs motorists that "the next exit to Hwy. 200 is closed." Suppose that m_2 informs that "the next 3 exits are closed due to construction." In this example, the message of m_1 is not fully covered by m_2 because this message emphasizes that it is not possible to take Hwy 200. Still, it is clear that if the same flow is exposed to both messages, the benefit is less than the sum of benefits.

We extend MBMAP to capture possible dependency between different messages and define two new problems: g-E-MBMAP (global extended MBMAP) and l-E-MBMAP (local extended MBMAP). In both new problems, we consider a given set E of events and assume that each message from M covers a subset of events from E. Each flow has a benefit from being notified about every event in every message. For example, if a flow is notified about the same event by two different IDDs, it obtains only the maximum benefit associated with this event and the two IDDs.

Let $\overrightarrow{B}(f,m,i)$ be a *benefit vector* representing all possible events in the considered system. Entry e in this vector indicates the benefit obtained by flow f when it is notified about e through message m in IDD i. If message m does not give any information regarding e, this entry is 0. Given a set U of such vectors, let $\overrightarrow{v} = \max\{U\}$ be a vector such that for every event e, $\overrightarrow{v}[e] = \max\limits_{\overrightarrow{u}\in U}\{\overrightarrow{u}[e]\}$. In addition, given any vector $\overrightarrow{v}$, let $|\overrightarrow{v}| = \sum_e \overrightarrow{v}[e]$.

We start with the definition of l-E-MBMAP. The benefit vector obtained by a flow f from a legal assignment T is

$$\overrightarrow{B_L}(T,f) = \sum_{(m,i)\in T} \overrightarrow{B}(f,m,i).$$

The total benefit from assignment T is: $\widehat{B_L}(T) = \sum_{f\in F} |\overrightarrow{B_L}(T,f)|$, and we seek to:

$$\begin{aligned} &\text{maximize} && \widehat{B_L}(T), \\ &\text{subject to:} && \sum_{m\in T(i)} size(m) \le size(i) \quad \text{for every } i. \end{aligned}$$

For g-E-MBMAP, the benefit vector obtained by a flow f from a legal assignment T is

$$\overrightarrow{B_G}(T,f) = \max_{(m,i)\in T}\{\overrightarrow{B}(f,m,i)\}.$$

The total benefit from assignment T is: $\widehat{B_G}(T) = \sum_{f\in F} |\overrightarrow{B_G}(T,f)|$, and we seek to:

$$\begin{aligned} &\text{maximize} && \widehat{B_G}(T), \\ &\text{subject to:} && \sum_{m\in T(i)} size(m) \le size(i) \quad \text{for every i.} \end{aligned}$$

For example, consider two messages, m_1, m_2, two IDDs, i_1 and i_2, and one flow f that passes through both i_1 and i_2. Suppose that there are 5 events and that: $\overrightarrow{B}(f,m_1,i_1) = (3,0,8,0,0)$, $\overrightarrow{B}(f,m_2,i_1) = (3,4,0,0,1)$, $\overrightarrow{B}(f,m_1,i_2) = (1,2,5,0,3)$ and $\overrightarrow{B}(f,m_2,i_2) = (1,0,0,0,3)$. Each entry in $\overrightarrow{B}$ represents the benefit obtained from a specific event. By assigning both m_1 and m_2 to i_1 and only m_1 to i_2, g-E-MBMAP gets the benefit $\overrightarrow{B_G}(T,f) = \max_{(m,i)\in T}\{\overrightarrow{B}(f,m,i)\} = (3,4,8,0,3)$. Since f is the only flow in this example, the total benefit is $\widehat{B}(T) = |\overrightarrow{B_G}(T,f)| = 18$.

We solve l-E-MBMAP using an algorithm for a generalization of the Budgeted Maximum Coverage Problem (BMCP) [18]. BMCP is defined as follows. Let S be a collection of sets with associated costs defined over a domain of weighted elements. Let L be a budget. Find a subset $S' \subseteq S$ such that the total cost of sets in S' does not exceed L, and the total weight of elements covered by S' is maximized.

We define the Generalized Budgeted Maximum Coverage Problem (GBMCP) as follows. Let S be a collection of sets with associated costs, where every set includes weighted elements. Find a subset of $S' \subseteq S$ such that the total cost of sets in S' does not exceed the budget L, and the total weight of elements covered by S' is maximized. In the generalized problem an element may have different weights in different sets. The algorithm for BMCP described in [18] can be easily extended for GBMCP while guaranteeing the same approximation ratio $(1-\frac{1}{e})$. In [6] the authors introduce a more generalized variation for BMCP, referred to as "The Generalized Maximum Coverage Problem," and present an $(\frac{e}{e-1}+\epsilon)$ approximation for every $\epsilon > 0$. In this variation, not only does the benefit of the element differ from one subset to another but also its weight.

The reduction of l-E-MBMAP to GBMCP is simple: the budget is $size(i_k)$; for every $m \in M$, let $\{f_j|$ such that $\overrightarrow{B}(f,m,i_k)[j] > 0\}$ is a subset with a cost of $size(m)$, where the weight of element f_j is $\overrightarrow{B}(f,m,i_k)[j]$.

To solve g-E-MBMAP, we use the Budgeted Maximum Coverage Problem (BMCP) [18] as a subroutine. Let ALG' be a β-approximation algorithm for GBMCP. We now build a $(1+\beta)$-approximation for g-E-MBMAP. This algorithm is a generalization of Algorithm 1 to vectors.

Algorithm 2: (an approximation algorithm for g-E-MBMAP)

Denote by $\overrightarrow{B}_k(f,m,i)$ the benefit vector obtained by flow f from assigning message m to device i at the k^{th} iteration of the algorithm. Initially, for every message m, device i and flow f, set $\overrightarrow{B}_1(f,m,i) \leftarrow \overrightarrow{B}(f,m,i)$. Then, For $k=1$ to $|I|$ do:

1) Run algorithm ALG' for GBMCP on the following instance: the budget is $size(i_k)$; for every $m \in M$, let $\{f_j|$ such that $\overrightarrow{B}(f,m,i_k)[j] > 0\}$ be a subset with a cost of $size(m)$, where the weight of element f_j is $\overrightarrow{B}(f,m,i_k)[j]$. Let N_k be the set of messages selected by ALG'.
2) If $|I| = k$, return $T = \bigcup_{j=1}^{|I|}\{N_j \times \{i_j\}\}$.
3) For every flow f, let $\overrightarrow{B}_k(f) = \max\{\overrightarrow{B}_k(f,m,i_k)|m \in N_k\}$, which is the benefit vector for flow f from assigning the messages in N_k to i_k.
4) For every message m, device i, and flow f:
   - Decompose the benefit vector $\overrightarrow{B}_k(f,m,i)$ into two vectors, $\overrightarrow{B}^1_k(f,m,i)$ and $\overrightarrow{B}^2_k(f,m,i)$, such that the following holds:
     a) $\overrightarrow{B}^1_k(f,m,i) = \begin{cases} \overrightarrow{B}_k(f,m,i_k) & \text{for } i = i_k \\ \min\{\overrightarrow{B}_k(f,m,i), \overrightarrow{B}_k(f)\} & \text{for } i \neq i_k, \end{cases}$

     where the minimum of two benefit vectors is a vector, such that the value of every element is the minimum between its values in the two vectors.

b) $\overrightarrow{B}^2_k(f,m,i) = \overrightarrow{B}_k(f,m,i) - \overrightarrow{B}^1_k(f,m,i)$.

- Set $\overrightarrow{B}_{k+1}(f,m,i) \leftarrow \overrightarrow{B}^2_k(f,m,i)$. ■

We analyze the worst-case performance of Algorithm 2 using the same method used for Algorithm 1. In order to use the Local Ratio theorem, we first prove that $\widehat{B_{Gk}}(T)$ satisfies the required conditions.

Lemma 2: For every iteration of the algorithm, $k = 1$ to $|I|$, and for $T_k = \bigcup_{j=k}^{|I|}\{N_j \times \{i_j\}\}$, $\widehat{B_{Gk}}(T_k) = \widehat{B_{Gk}}^1(T_k) + \widehat{B_{Gk}}^2(T_k)$.

The proof is presented in the Appendix.

Theorem 3: For every $1 \leq k \leq |I|$, $T_k = \bigcup_{j=k}^{|I|}\{N_j \times \{i_j\}\}$ is a $(1+\beta)$-approximation with respect to $\overrightarrow{B}_k(f,m,i)$.

The proof is presented in the Appendix.

Corollary 2: Algorithm 2 is a $(1+\beta)$-approximation for g-E-MBMAP.

In order to analyze the running time of Algorithm 2, assume that the running time of any operation we make on two vectors is $O(k)$, where k is the number of possible events. Assume also that the running time of algorithm ALG' is $g(|M|, k \cdot |F|)$. In step 1 we build a GBMCP instance. The running time of this construction is bounded by $O(|M| \cdot |F| \cdot k)$. In step 3 we calculate the benefit obtained by each flow, which takes $O(|F| \cdot |N_k| \cdot k) = O(|F| \cdot |M| \cdot k)$. The running time of step 4 is $O(|M| \cdot |F| \cdot |I| \cdot k)$. Hence, the total running time of Algorithm 2 is:

$$O(|M| \cdot |F| \cdot |I|^2 \cdot k + |I| \cdot g(|M|, k \cdot |F|)).$$

VI. Simulation Study

In this section we introduce Monte Carlo simulation results for the various models and algorithms presented in the earlier sections. The simulation is built from scratch on a Linux machine using C++. We consider the following four models: (a) the l-MBMAP model; (b) the g-MBMAP model; (c) the l-E-MBMAP model; (d) the g-E-MBMAP model. While it is clear that in general the global algorithms perform better than the local algorithms, our main purpose in this section is two-fold: first, to study the effect of the extended model on the performance; second, to better understand the effect of some network parameters on each of the various models and algorithms.

We simulate the various models in the context of IDDs broadcasting messages to cars in a vehicular network. The network of roads is simulated as a weighted graph, with 200 vertices spread out on a grid. Each vertex has a potential edge with each of 4 potential neighbors. However, we randomly determine for each potential edge whether it appears in the graph. Every vertex represents an intersection of roads, and every edge represents a directional road between two intersections. A vertex can be connected to at most four edges. The weight associated with each edge represents the delay encountered by a vehicle upon traversing the corresponding road. This delay is a function of the distance and the volume of traffic on the road.

We then uniformly place IDDs along the busiest roads and consider a list of events that are likely to be of interest to the passing vehicles (see further details below on how events are chosen). The average size of each IDD is 40. From the chosen events we derive a set of messages. The average length of a message is 10. The benefit of a message depends on the events it announces, the location of the IDD and the volume of the flow to which it is relevant. For example, if a message m is assigned to an IDD i to which the vehicles of a flow f are exposed only after they are affected by the event, then $B(f,m,i) = 0$.

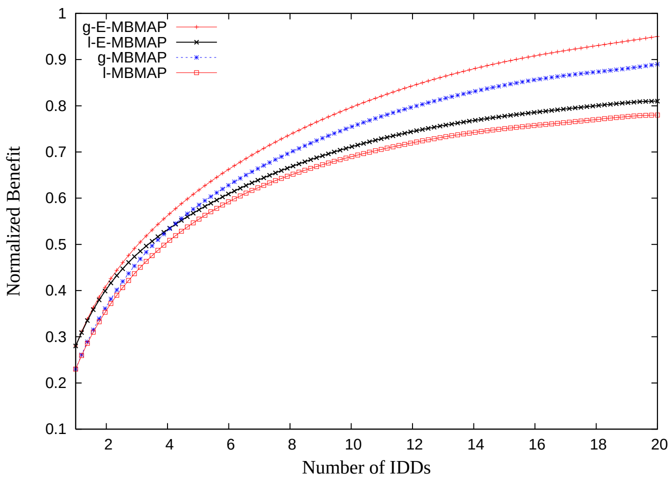
Group mobility is simulated as follows. For every flow we choose random source and destination points. Then, we compute the shortest path P between these points. This is the path traversed by the flow's nodes. We also choose a random size for each flow, which indicates the number of mobile devices in this flow per minute. In this section we consider each of the flows independently. In the next section we show how the number of flows can be minimized by merging some of them, in a similar fashion to that proposed by the virtual track (VT) group mobility model [26].

Events and messages are generated in the following way. A random place is chosen on the graph. It can be either a node (junction) or an edge (road). An event is defined for each road and junction. Then, 15 or 50 events are randomly chosen and the other are ignored. Each flow obtains a benefit if it is informed about a relevant event. The sooner the flow is exposed to a message on a relevant event, the higher the corresponding benefit is. However, we received similar results when we used different benefit assignment functions, such as a uniform one. We then compute $\overrightarrow{B}(f,m,i)$, namely, the vector of benefits from the assignment of m to i for every f, m and i. Recall that in l-MBMAP and g-MBMAP we ignore the dependency between different messages broadcast by the same IDD. Hence, for these models we set $B(f,m,i) = |\overrightarrow{B}(f,m,i)|$.

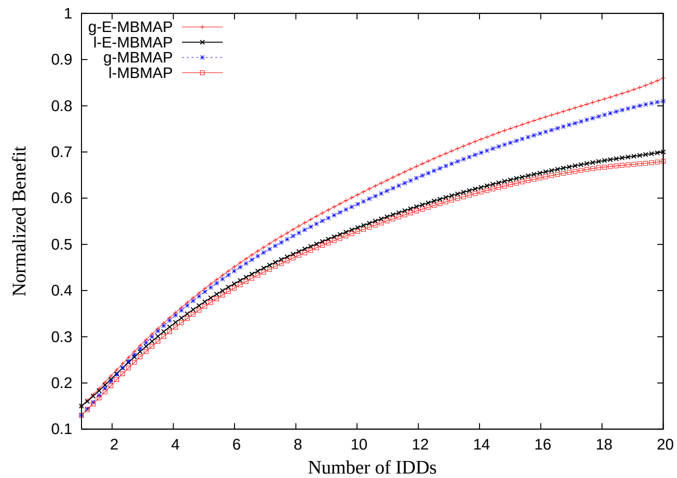
For each model we execute the corresponding algorithm on the corresponding input. The output of each execution is an assignment $T \subseteq \{M \times \{I\}\}$. To measure the quality of each assignment, we divide its total benefit by the maximum theoretical benefit that can be obtained when there is no limit on the volume of information each IDD can broadcast, in which case all messages are broadcast by all IDDs.

For l-MBMAP, we solve the Knapsack problem using the dynamic programming algorithm described in [21]. This algorithm finds an optimal solution. It has a pseudo-polynomial running time of $O(|I| \cdot |M|)$, which is reasonable for all the models considered in this paper. Therefore, Algorithm 1 is a 2-approximation for g-MBMAP. For l-E-MBMAP and g-E-MBMAP, we solve the Budgeted Maximum Coverage Problem using the $\frac{e}{e-1}$-approximation proposed in [18]. This guarantees an $\frac{e}{e-1}$-approximation for l-E-MBMAP and a $\frac{2e-1}{e-1}$-approximation for g-E-MBMAP.

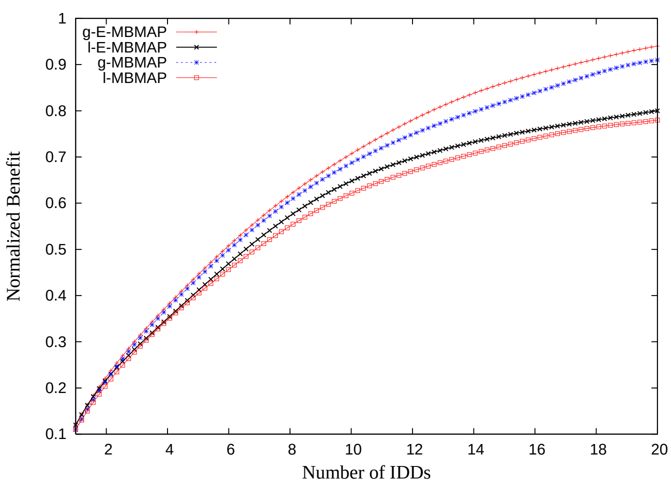
The g-MBMAP model is better than the l-MBMAP because it is aware of the dependency between messages assigned to different IDDs. In particular, g-MBMAP knows that assigning the same message to closely related IDDs, i.e., IDDs that are traversed by a similar set of flows, allows the traversing flow to benefit only from one message, while l-MBMAP does not take



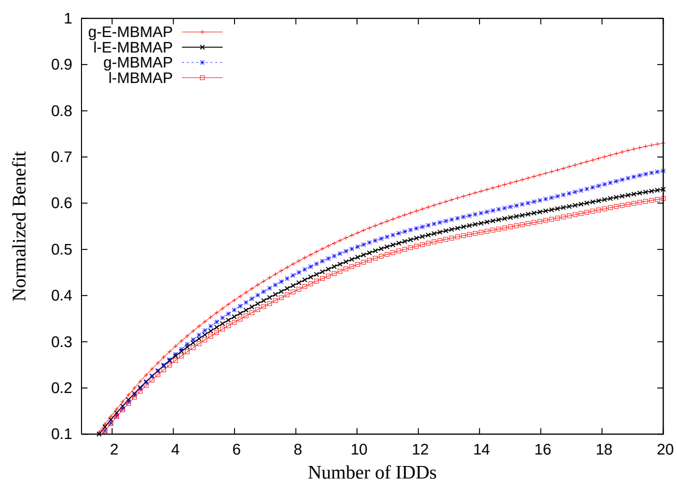
(a) Original setting



(b) Increasing the number of possible events



(c) Decreasing the average number of messages that are relevant to every flow



(d) Decreasing the average number of IDDs that are traversed by every flow

Fig. 2. The normalized benefit as a function of the number of IDDs

such considerations into account. Thus, we expect g-MBMAP to perform better than l-MBMAP. For the same reason, g-E-MBMAP is better than l-E-MBMAP. If we compare the two global models, g-MBMAP and g-E-MBMAP, the latter is obviously better than the former. While both models are aware that it is pointless to assign the same message to two closely related IDDs, g-E-MBMAP is also aware that two different messages might give similar information. To summarize, we expect g-E-MBMAP to exhibit the best performance and l-MBMAP the worst.

Figure 2(a) depicts the normalized benefit as a function of the number of IDDs for 30 messages, 30 flows, 20 IDDs, 10 relevant messages for every flow on the average, and 10 IDDs traversed by every flow on the average. The number of possible events is 15, and for every message there are 5 possible events on the average. The ratio between the number of messages and the number of event indicates the dependency between the different messages. Obviously, for all the models, the total benefit increases with the number of IDDs. However, the growth rate declines because, when a flow traverses more and more IDDs, it is less likely to receive a new message. When comparing the curves for the different models, we see that l-MBMAP indeed exhibits the worst performance, whereas g-E-MBMAP exhibits the best. The graph also shows that when the number of IDDs is small, l-E-MBMAP outperforms g-MBMAP. However, with more IDDs, g-MBMAP is better than l-E-MBMAP.

We now increase the number of possible events from 15 to 50 without changing the other parameters. This reduces the dependency between the different messages because two messages are less likely to announce similar events. The results are presented in Figure 2(b). The performance for all the models is lower than what we saw in Figure 2(a) because every message covers relatively few events. Since the IDD capacity is limited, fewer events are covered in this case. In addition, we see that the advantage of g-E-MBMAP over g-MBMAP is now smaller, because of the decreased dependency between the different messages. For the same reason, we see that the advantage of l-E-MBMAP over l-MBMAP is much smaller.

Figure 2(c) shows the results for the same parameters considered in Figure 2(a), except that the average number of messages relevant to each flow is reduced from 10 to 5. As in Figure 2(a), we see again that the performance of g-MBMAP is closer to the performance of g-E-MBMAP. This is because fewer messages can be broadcast by the same IDD for every flow, so there is less dependency between messages for the same flow. This reduces the advantage of g-E-MBMAP over g-MBMAP and of l-E-MBMAP over l-MBMAP.

Finally, Figure 2(d) shows the results for the same parameters considered in Figure 2(a), except that the average number of IDDs traversed by each flow is reduced from 10 to 5. As expected, the benefit for all the models is smaller than the benefit shown in Figure 2(a). In addition, we can see that all four curves are much closer to each other. This is because the probability that the same message will be assigned to different IDDs is smaller, which reduces the advantages of g-MBMAP and g-E-MBMAP.

We conclude that the global models and algorithms are significantly better than the local ones for large-scale settings. Nonetheless, the actual setting is what determines whether it is advantageous to consider the dependency between events in different messages. For the local model, we see that l-E-MBMAP has no real advantage over l-MBMAP for almost every setting. g-E-MBMAP has an advantage over g-MBMAP for almost every setting, but the performance gain is more significant when the number of dependencies is small and when the number of messages affecting every flow is large.

VII. Flow Definitions

In the previous sections we assumed that we are given a set of flows as a parameter. However, defining such flows is not a trivial task. We seek a scheme that defines flows in a way that is not only accurate, but also scalable. We compare the accuracy and scalability of the following four schemes:

- Scheme A: In this scheme we ignore flows whose volume, in terms of mobile nodes per second, is small. The rationale is that such flows are not likely to obtain a lot of benefit and thus can be ignored.
- Scheme B: Here we ignore flows whose volume multiplied by the number of IDDs they pass is small. This is because it is more difficult to obtain benefit for these flows without affecting the benefit obtained for other flows.
- Scheme C: Here we ignore flows whose volume multiplied by the number of messages they encounter is small. This is for the same reason stated in Scheme B.
- Scheme D: Here the considered metric is the benefit every flow could obtain if it was the only flow in the network, normalized by the resources it requires to obtain this benefit. Again, those flows for which this metric is small are ignored.

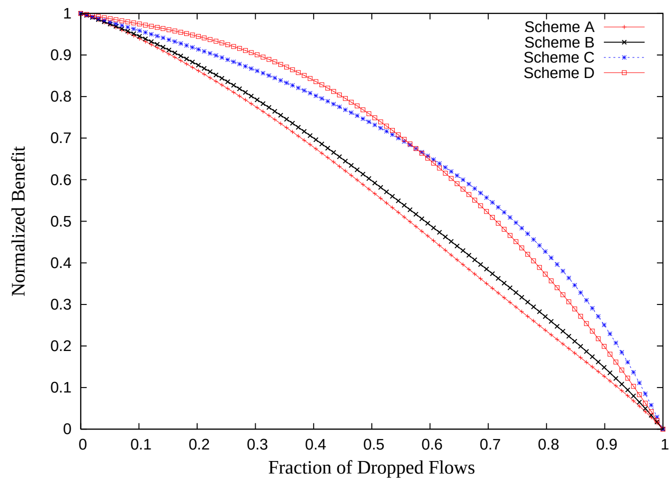
We compare the proposed schemes by running each of the four algorithms with each scheme. For lack of space, we report here only the results obtained for g-E-MBMAP. However, the results obtained for each of the other algorithms were similar. Figure 3 shows the simulation results. In all the graphs, the x-axis indicates the fraction of ignored flows while the y-axis indicates the "normalized benefit." The normalized benefit for each scheme indicates the total benefit obtained for this scheme divided by the total benefit obtained when all the flows are considered. Hence, we see that for all the schemes in all the graphs the normalized benefit is 1 when no flow is ignored, and this benefit decreases with the number of dropped flows. The decrease rate depends on the scheme and on the parameters of each scenario.

Figure 3(a) depicts the normalized benefit as a function of the fraction of dropped flows for 50 IDDs and 50 messages. Schemes C and D exhibit the best performance in this case, because with these parameters there are many available IDDs but not many messages to disseminate. Schemes C and D keep flows that can benefit from more messages and consequently have higher potential benefit when there are enough IDDs. Comparing the curves of schemes C and D, we see that Scheme D achieves better performance for low drop rates, while for high drop rates the performance decreases. This is because, for low drop rates, the competition between the flows on the available IDDs is fairly high, and Scheme D keeps flows that obtain higher benefit from their messages. For high drop rates, the competition is significantly lower, and Scheme C, which keeps flows that need many messages, achieves better performance. Schemes A and B do not take into account the small number of messages, so they consider many flows that obtain a low benefit.

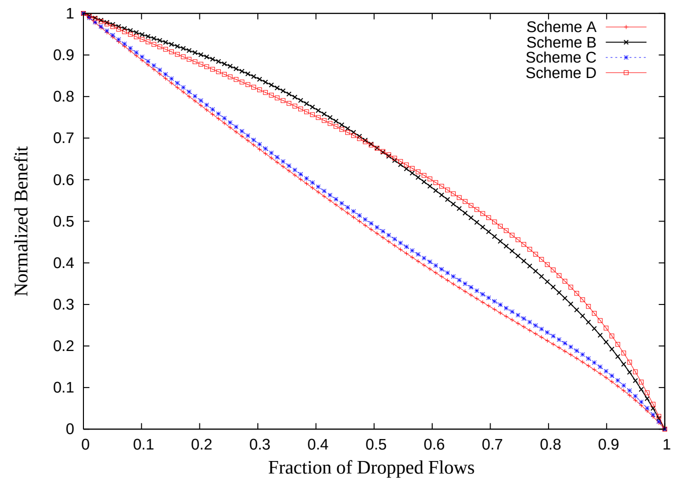
Figure 3(b) shows the simulation results for 20 IDDs and 120 messages. In this case, schemes B and D achieve the best performance. This is because most flows encounter a small number of IDDs but need many messages. Thus, the IDDs are a scarce resource. Scheme B keeps flows that pass through many IDDs. Thus, this scheme decreases the competition on the IDDs and increases the number of relevant IDDs for every message. Comparing Scheme B with Scheme D, we see that Scheme B exhibits a better performance for lower drop rates whereas Scheme D is better for higher drop rates. This is because Scheme D seeks to increase the local benefit, whereas Scheme B seeks to decrease the competition among different flows and to increase the global benefit. Schemes A and C show poor performance in this case because they mostly consider flows that pass through few IDDs, thereby increasing the competition and decreasing the total benefit.

Figure 3(c) shows the results for 50 IDDs and 120 messages. In this setting, the competition between the messages on the IDDs is high, with only a small benefit for many of the flows. Scheme A is the worst for this setting, whereas the other three schemes perform similarly. When the number of flows decreases, Scheme C shows the best results because it is more likely to keep flows that can benefit even in a highly competitive environment.

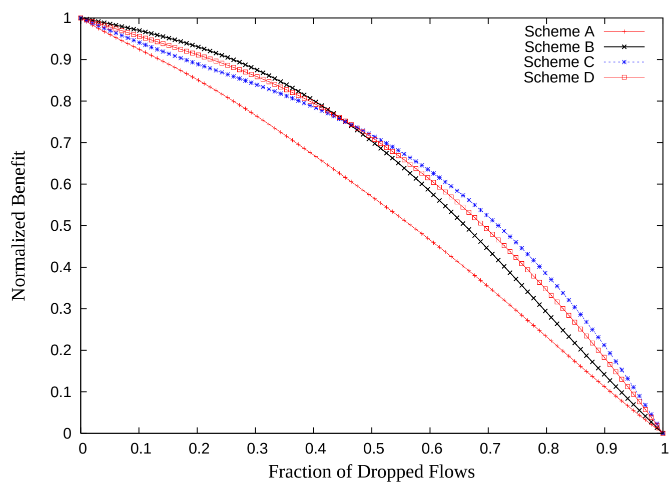
Figure 3(d) shows simulation results for 20 IDDs and 50 messages. Since we have few IDDs and messages, the number of flows that encounter many IDDs or are interested in many messages is small. Thus, all the schemes except D are likely to keep flows with a low potential benefit that either pass through few IDDs or are interested in few messages. Scheme



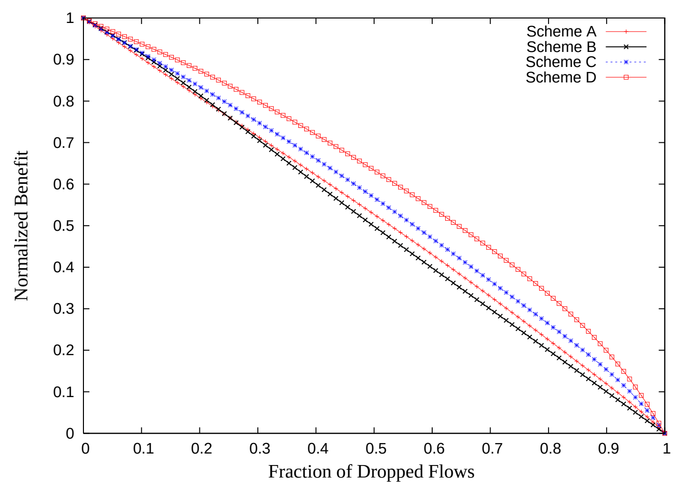
(a) 50 IDDs and 50 messages



(b) 20 IDDs and 120 messages



(c) 50 IDDs and 120 messages



(d) 20 IDDs and 50 messages

Fig. 3. The normalized benefit as a function of rate of the dropped flows

D is more likely to keep flows with high potential benefit, and it therefore has the best performance in this case.

To conclude this section, we believe that Scheme C and Scheme D yield the best trade-off between scalability and performance.

VIII. Conclusions

The paper presented models and algorithms for efficient location-based decision supporting content distribution to mobile groups. The considered system consists of Information Dissemination Devices (IDDs), which broadcast a limited amount of location-based information to passing mobile nodes that are moving along well-defined paths. We concentrated on the optimization problem of assigning messages to the IDDs. The IDDs disseminate these messages to groups of passing mobile nodes whose mobility pattern is well defined. We formulated several related models and studied both the theoretical aspects of the resulting optimization problems and the practical implications of deploying efficient algorithms for these models in realistic networking scenarios.

Our results indicated that the cooperative solutions, in which the assignment is made for all the IDDs together, are significantly better than the non-cooperative ones. In contrast, the contribution of the extended models, that also capture the dependencies between messages broadcast by the same IDD, is relatively minor.

References

[1] S. Acharya, R. Alonso, M. Franklin, and S. Zdonik. Broadcast disks: Data management for asymmetric communication environments. In *Proceedings of ACM SIGMOD*, pages 199–210, 1995.

[2] M. Aezalden, R. Cohen, and D. Raz. Locally vs. globally optimized flow-based content distribution to mobile nodes. Technical report, Technion, Dept. of Computer Science, MSC-2009-04, 2009.

[3] R. Bar-Yehuda, K. Bendel, A. Freund, and D. Rawitz. Local ratio: A unified framework for approximation algorithms. *ACM Computing Surveys*, 36(4):422 – 463, Dec. 2004.

[4] C. Bettstetter, G. Resta, and P. Santi. The node distribution of the random waypoint mobility model for wireless ad hoc networks. *IEEE Trans. on mobile computing*, 2(3):257 – 269, 2003.

[5] R. Bishop. A survey of intelligent vehicle applications worldwide. In *The IEEE Intelligent Vehicles Symposium*, pages 25–30, 2000.

[6] R. Cohen and L. Katzir. The generalized maximum coverage problem. *Information Processing Letters (IPL)*, 108(1):15–22, September 2008.

[7] R. Cohen, L. Katzir, and D. Raz. Scheduling algorithms for a cache pre-filling content distribution network. In *INFOCOM'2002*, June 2002.

[8] R. Cohen, L. Katzir, and D. Raz. An efficient approximation for the generalized assignment problem. *Information Processing Letters*, 100:162–166, 2006.

[9] A. Dekker. Centralisation and decentralisation in network centric warfare. *Journal of Battlefield Technology*, 6(2), July 2003.

[10] F. Dressler. Network-centric actuation control in sensor/actuator networks based on bio-inspired technologies. In *IEEE Mobile Adhoc and Sensor Systems (MASS)*, 2006.

[11] U. Feige and J. Kilian. Making games short (extended abstract). In *Annual ACM Symposium on Theory of Computing*, pages 506–516, 1997.

[12] N. Fisher and S. Baruah. A fully polynomial-time approximation scheme for feasibility analysis in static-priority systems with arbitrary relative deadlines. In *ECRTS '05: Proceedings of the 17th Euromicro Conference on Real-Time Systems (ECRTS'05)*, pages 117–126, 2005.

[13] N. Hirakata, M. Imai, A. Kamemura, T. Maehata, H. Takahashi, K. Tanaka, and N. Yumoto. DSRC using OFDM for roadside-vehicle communication system. *VTC (Vehicular Technology Conference)*, 1:148–152, 2003.

[14] D. S. Hochbaum, editor. *Approximation Algorithms for NP-Hard Problems*. PWS Publishing Co., Boston, MA, USA, 1997.

[15] X. Hong, M. Gerla, G. Pei, and C. Chiang. A group mobility model for ad hoc wireless networks. In *The 2nd ACM international workshop on Modeling, analysis and simulation of wireless and mobile systems*, 1999.

[16] O. H. Ibarra and C. E. Kim. Fast approximation algorithms for the knapsack and sum of subset problems. *J. ACM*, 22(4), 1975.

[17] S. Khanna and V. Liberatore. On broadcast disk paging. *SIAM Journal on Computing*, 29(5), 2000.

[18] S. Khuller, A. Moss, and J. S. Naor. The budgeted maximum coverage problem. *Information Processing Letters*, 70(1):39–45, 1999.

[19] G. Korkmaz, E. Ekici, F. Özgner, and U. Özgner. Urban multi-hop broadcast protocol for inter-vehicle communication systems. In *Vehicular Ad Hoc Networks*, pages 76 – 85, 2004.

[20] V. Liberatore. Caching and scheduling for broadcast disk systems. *Journal of Experimental Algorithmics*, 6(5), 2001.

[21] U. Pferschy. Dynamic programming revisited: Improving knapsack algorithms. *Computing*, 63(4):419–430, 1999.

[22] S. Samal. Mobility pattern aware routing in mobile ad hoc networks. Technical report, Virginia Polytechnic Institute and State University, 2003.

[23] M. Torrent-Moreno, D. Jiang, and H. Hartenstein. Broadcast reception rates and effects of priority access in 802.11-based vehicular ad-hoc networks. In *Vehicular Ad Hoc Networks*, pages 10–18, 2004.

[24] H. Wu, R. Fujimoto, R. Guensler, and M. Hunter. MDDV: a mobility-centric data dissemination algorithm for vehicular networks. In *VANET'04: Proceedings of the 1st ACM International Workshop on Vehicular Ad Hoc Networks*, pages 47–56, 2004.

[25] Q. Xu, K. Hedrick, R. Sengupta, and J. VanderWerf. Effects of vehicle-vehicle/roadside-vehicle communication on adaptive cruise controlled highway systems. In *Proceedings IEEE Vehicular Technology Conference*, pages 1249– 1253, 2002.

[26] B. Zhou, K. Xu, and M. Gerla. Group and swarm mobility models for ad hoc network scenarios using virtual tracks. In *IEEE MILCOM*, 2004.

Appendix

The proof of Lemma 2: By definition

$$\widehat{B_G}_k(T) = \sum_{f \in F} |\overrightarrow{B}_k(T,f)| = \sum_{f \in F} |\max_{(m,i)\in T}\{\overrightarrow{B}_k(f,m,i)\}|$$

$$= \sum_{f \in F} |\max_{(m,i)\in T}\{\overrightarrow{B}^1_k(f,m,i) + \overrightarrow{B}^2_k(f,m,i)\}|.$$

Given a flow f, a message m and an index j, we need to prove that:

$$\max_{(m,i)\in T}\{\overrightarrow{B}^1_k(f,m,i) + \overrightarrow{B}^2_k(f,m,i)\}[j]$$

$$= \max_{(m,i)\in T}\{\overrightarrow{B}^1_k(f,m,i)\}[j] + \max_{(m,i)\in T}\{\overrightarrow{B}^2_k(f,m,i)\}[j].$$

Since $\overrightarrow{B}_k(f,m,i) = \overrightarrow{B}^1_k(f,m,i) + \overrightarrow{B}^2_k(f,m,i)$, and all the values in all the vectors are non-negative, the following holds for every flow f and index j

$$\max_{(m,i)\in T}\{\overrightarrow{B}^1_k(f,m,i)\}[j] + \max_{(m,i)\in T}\{\overrightarrow{B}^2_k(f,m,i)\}[j]$$

$$\geq \max_{(m,i)\in T} B_k(f,m,i)[j]$$

Thus, it remains to show that for every flow f and index j

$$\max_{(m,i)\in T} B_k(f,m,i)[j] \geq$$

$$\max_{(m,i)\in T}\{B^1_k(f,m,i)\}[j] + \max_{(m,i)\in T}\{B^2_k(f,m,i)\}[j].$$

By definition, $\overrightarrow{B}_k(f)$ is equal to $\max_{m\in N_k} \overrightarrow{B}_k(f,m,i_k)$, which is the benefit vector for flow f from assigning the messages in N_k to i_k. If $\max_{(m,i)\in T} \overrightarrow{B}^2_k(f,m,i)[j] = 0$, the claim is proven. Otherwise, $\max_{(m,i)\in T} \overrightarrow{B}^2_k(f,m,i)[j] = l > 0$. Suppose that the maximum holds for $i_j \neq i_k$, i.e., $\overrightarrow{B}^2_k(f,m,i_j)[j] = l$. Then, by construction, $\overrightarrow{B}^1_k(f,m,i_j)[j] = \overrightarrow{B}_k(f)[j] = \max_{m\in N_k} \overrightarrow{B}^1_k(f,m,i_k)[j]$. Thus,

$$\max_{(m,i)\in T}\{\overrightarrow{B}_k(f,m,i)\}[j] \geq$$

$$\overrightarrow{B}_k(f,m,i_j)[j] = \overrightarrow{B}^1_k(f,m,i_j) + \overrightarrow{B}^2_k(f,m,i_j)[j] =$$

$$\max_{(m,i)\in T}\{\overrightarrow{B}^1_k(f,m,i)\}[j] + \max_{(m,i)\in T}\{\overrightarrow{B}^2_k(f,m,i)\}[j].$$

Now we can conclude that

$$\overrightarrow{B}_k(T) = \sum_{f\in F} |\max_{(m,i)\in T}\{\overrightarrow{B}^1_k(f,m,i) + \overrightarrow{B}^2_k(f,m,i)\}|$$

$$= \sum_{f\in F}\sum_i \max_{(m,i)\in T}\{\overrightarrow{B}^1_k(f,m,i) + \overrightarrow{B}^2_k(f,m,i)\}[j]$$

$$= \sum_{f\in F}\sum_i \max_{(m,i)\in T}\{\overrightarrow{B}^1_k(f,m,i)\}[j] +$$

$$+ \sum_{f\in F}\sum_i \max_{(m,i)\in T}\{\overrightarrow{B}^2_k(f,m,i)\}[j]$$

$$= \sum_{f\in F} |\max_{(m,i)\in T}\{\overrightarrow{B}^1_k(f,m,i)\}| + |\max_{(m,i)\in T} \overrightarrow{B}^2_k(f,m,i)\}|$$

$$= \widehat{B_G}^1_k(T) + \widehat{B_G}^2_k(T).$$

The proof of Theorem 3: We prove this by a reverse induction on the value of k, starting with $k = |I|$. The induction basis ($k = |I|$) follows from the validity of ALG', which produces a β-approximation for the GBMCP problem.

For the inductive step, assume that T_{k+1} is a $(1+\beta)$-approximation with respect to $\overrightarrow{B}_{k+1}(f,m,i)$. We now prove in two steps that T_k is a $(1+\beta)$-approximation with respect to $\overrightarrow{B}_k(f,m,i)$. If T_k is a $(1+\beta)$-approximation with respect to $\overrightarrow{B}^2_k(f,m,i)$, and T_k is also a $(1+\beta)$-approximation with respect to $\overrightarrow{B}^1_k(f,m,i)$, then since $\widehat{B_G}_k(T_k) = \widehat{B_G}^1_k(T_k) + \widehat{B_G}^2_k(T_k)$ (by Lemma 2) we can use the Local Ratio Theorem to complete the proof. Thus, it remains to prove that T_k is

$(1+\beta)$-approximation with respect to $\overrightarrow{B}^1_k(f,m,i)$ and with respect to $\overrightarrow{B}^2_k(f,m,i)$.

By the construction of $\overrightarrow{B}_k(f,m,i)$, for every message m, device i and flow f, $\overrightarrow{B}_{k+1}(f,m,i)$ is identical to $\overrightarrow{B}^2_k(f,m,i)$. By the induction assumption, T_{k+1} is a $(1+\beta)$-approximation with respect to $\overrightarrow{B}^2_k(f,m,i)$. T_k contains all of the assignments in T_{k+1}. Thus, T_k is $(1+\beta)$-approximation with respect to $\overrightarrow{B}^2_k(f,m,i)$.

Now, we prove that T_k is also a $(1+\beta)$-approximation with respect to $\overrightarrow{B}^1_k(f,m,i)$. The benefit function $\overrightarrow{B}^1_k(f,m,i)$ has three components. The first component c_1 is the benefit of assigning a message m to i_k, namely $\sum_{f\in F}\overrightarrow{B}_k(f,m,i_k)$. The second component c_2 is the benefit of messages selected to N_k, whose benefit in $\overrightarrow{B}^1_k(f,m,i)$ is set to be not larger than the benefit from their assignment to i_k. The third component c_3 consists of all the remaining entries, which are set to 0. Algorithm ALG' guarantees β-approximation for i_k with respect to c_1. Therefore, the best solution with respect to c_1 has benefit of at most $\beta\cdot\sum_{m\in N_k}\sum_{f\in F}|\overrightarrow{B}^1_k(f,m,i)|$. Any solution with respect to c_2 will have benefit of at most $\sum_{m\in N_k}\sum_{f\in F}|\overrightarrow{B}^1_k(f,m,i)|$, since the benefit of every entry in this component is set to be not larger than the benefit of $\overrightarrow{B}^1_k(f,m,i_k)$. Note also that the benefit a flow obtains is at most the maximum between these benefits, no matter by how many IDDs it is covered. Finally, component c_3 has no contribution to the benefit. Thus, N_k is a $(1+\beta)$-approximation with respect to $\overrightarrow{B}^1_k(f,m,i)$. Since T_k contains all the assignments in N_k, it is a $(1+\beta)$-approximation with respect to $\overrightarrow{B}^1_k(f,m,i)$, which concludes the proof.

PLACE PHOTO HERE

Mhameed Aezladen received his B.Sc. degree in Computer Science from the Technion - Israel Institute of Technology in 2004. Since 2005, he has been an Ms.C. student in Computer Science Department in the Technion, working on flow-based content distribution to mobile nodes.

PLACE PHOTO HERE

Reuven Cohen (M'93, SM'99) received the B.Sc., M.Sc. and Ph.D. degrees in Computer Science from the Technion - Israel Institute of Technology, completing his Ph.D. studies in 1991. From 1991 to 1993, he was with the IBM T.J. Watson Research Center, working on protocols for high speed networks. Since 1993, he has been a professor in the Department of Computer Science at the Technion. He has also been a consultant for numerous companies, mainly in the context of protocols and architectures for broadband access networks. Reuven Cohen has served as an editor of the IEEE/ACM Transactions on Networking and the ACM/Kluwer Journal on Wireless Networks (WINET). He is the co-chair of the technical program committee of Infocom 2010. Reuven Cohen is a senior member of the IEEE and heads the Israeli chapter of the IEEE Communications Society.

PLACE PHOTO HERE

Danny Raz received his doctoral degree from the Weizmann Institute of Science, Israel, in 1995. From 1995 to 1997 he was a post-doctoral fellow at the International Computer Science Institute, (ICSI) Berkeley, CA, and a visiting lecturer at the University of California, Berkeley. Between 1997 and 2001 he was a Member of Technical Staff at the Networking Research Laboratory at Bell Labs, Lucent Technologies. In October 2000, Danny Raz joined the faculty of the computer science department at the Technion, Israel.

Danny Raz served as the general chair of OpenArch 2000, and as a TPC member for many conferences including INFOCOM 2002-2003, OpenArch 2000-2001-2003, IM-NOMS 2001-2007. He is an Editor of the IEEE/ACM Transactions on Networking (ToN) and has co-chaired the Technical Program committee in IM 2009. His primary research interest is the theory and application of management related problems in IP networks, with a special emphasis on efficient resource utilization.