TOMP: Opportunistic Traffic Offloading Using Movement Predictions

Patrick Baier, Frank Dürr, Kurt Rothermel
Institute of Parallel and Distributed Systems
Universität Stuttgart
70569 Stuttgart, Germany
Email: {baier, duerr, rothermel}@ipvs.uni-stuttgart.de

Abstract*—Recent forecasts predict that the amount of cellular data traffic will significantly increase within the next few years. The reason for this trend is on the one hand the high growth rate of mobile Internet users and on the other hand the growing popularity of high bandwidth streaming applications. Given the fact that cellular networks (e.g. UMTS) have only limited capacity, the existing network infrastructure will soon reach its limits. As a result, the concept of traffic offloading attracts more and more attention in research since it aims at the reduction of cellular traffic by shifting it to local-area networks like Wi-Fi. One particular form of traffic offloading is known as opportunistic traffic offloading and follows the basic idea to shift traffic from the cellular network to the level of inter-device communication of mobile devices. To perform opportunistic traffic offloading in an efficient way, assumptions about the prospective inter-device connectivity of the mobile devices have to be made. In general, the more inter-device connections are possible the more traffic can be offloaded. To utilize this fact, we developed the TOMP system. TOMP is the first opportunistic traffic offloading system that uses movement predictions of mobile users to analyze the prospective inter-device connectivity. In this paper we propose three different metrics for analyzing movement predictions and present an algorithm, which uses these metrics to utilize an efficient opportunistic traffic offloading. To evaluate TOMP, we show by simulation that we can save up to 40% of cellular messages in comparison to a typical cellular network.

***Index Terms*—Mobile Computing, Energy-aware systems**

I. Introduction

Within the last few years the vision of an ubiquitous Internet access came true. With the introduction of powerful smartphones and the increasing availability of reasonable mobile data rates, the number of mobile Internet users significantly grew. Simultaneously, applications like audio or video streaming, which demand high bandwidth, are getting more and more popular. These two trends led to an enormous increase in the amount of data that is transmitted via cellular networks (e.g. UMTS or HSDPA), which have only limited capacity. According to the latest forecasts, these trends will continue and the volume of cellular data traffic will further increase within the next years. For instance, Cisco predicted that the number of mobile Internet users is expected to double every year until 2015 [1]. In accordance with that, Ericsson recently forecasted that the amount of smartphone traffic will increase by factor ten until 2016 [2]. Both studies show that the traffic load on cellular networks may soon reach the networks' critical limit. Some mobile service providers already reacted by decreasing network cell sizes or by going back to volume-based pricing models [3]. Apart from that, some first research publications propose an alternative way for reducing cellular traffic, which is known as *cellular traffic offloading*.

The basic idea of cellular traffic offloading is to automatically shift traffic from the cellular network to a local-area network that provides higher bandwidth and is usually less loaded. Given a message m and a set of mobile devices that should receive m, the goal is to reduce the total amount of cellular traffic that the message delivery causes, in order to unburden the cellular network. Most of the prevailing approaches rely on the availability of publicly accessible Wi-Fi hotspots that can be used to relay m (e.g. [4]). Instead of sending m via the cellular network, message m can be sent to these hotspots, which distribute m via WLAN to the devices. Obviously, this is not suitable when no Wi-Fi hotspots are available or the hotspots are closed to public access. In contrast, another set of approaches uses the fact that mobile devices can set up inter-device connections (e.g. via Bluetooth) to exchange data (e.g. [5],[6]). For instance, we consider an application that runs in the infrastructure and wants to send a message m with a size of several MB (e.g. a video clip) to a set of mobile devices N that are located within the same neighborhood. Instead of individually sending m via the cellular network to all $n \in N$, the application can send it to only a subset $T \subset N$ of the mobile devices. These devices then start to locally distribute m to the other devices that have not received m so far. Obviously, this reduces the load on the cellular network and works without any public Wi-Fi hotspots. Since the inter-device communication is of opportunistic nature, this method is known as *opportunistic traffic offloading*.

A typical scenario for opportunistic traffic offloading is the delivery of information to a particular interest group. For instance, consider the mobile subscribers of a news portal which publishes articles that include audio and video files. Whenever a new article should be distributed to the subscribers, the news agency can use opportunistic traffic offloading to reduce the cellular network load. Another application is the distribution of sensing tasks, which is relevant in the context of our Public Sensing system *Com'N'Sense*[1]. In this context, opportunistic

[1] http://www.comnsense.de

Published in the Proceedings of the of the 37th IEEE Conference on Local Computer Networks (LCN)

traffic offloading helps to efficiently distribute sensing tasks to mobile devices that are eligible for recording sensor data (e.g. [7], [8], [9]).

In general, the goal in opportunistic traffic offloading is to find an optimal subset of prospective message receivers, called *target set* T, to which m should be sent via the cellular network, in order to ensure that m is opportunistically forwarded to as many receivers as possible. For identifying an optimal target set, some information about the prospective inter-device connectivity has to be assumed. For instance, Han et al. [5] use the history of social relations of the mobile device owners to identify those devices that are most likely to meet many other devices. This increases the chance that m is widely distributed. Obviously, the drawback of this solution lies in the fact that knowledge about the social history is not available in most cases. Moreover, even if the system would be able to collect this data over time, the data collection would be very critical with respect to privacy issues. In contrast to the prevailing approaches in opportunistic traffic offloading, TOMP is the first that uses predictions about the future movement of the mobile devices to estimate the inter-device connectivity. Therefore, it only needs information about the position and speed of the mobile devices and thus is easily deployable in existing cellular networks. Based on the movement predictions, TOMP delivers the cellular messages to those devices that are most likely to meet many other devices.

More precisely, our contributions are as follows:

1) We introduce the TOMP systems for opportunistic traffic offloading that is easily deployable in a conventional cellular network.
2) We provide a target set selection algorithm that chooses an appropriate target set for message distribution.
3) We present three different metrics for predicting the future movement of mobile devices. These metrics serve as input to the target set selection algorithm.
4) We show by extensive simulation that TOMP helps to reduce the number of cellular messages by up to 40% in comparison to a conventional cellular network.

The rest of this paper is structured as follows: In Section II we introduce our system model and then give a problem statement in Section III. For an easy solution to this problem, we first introduce an naive algorithm in Section IV. Section V introduces the basic message delivery process of the algorithm used in TOMP. Section VI defines three basic metrics upon which the target set selection algorithm, presented in Section VII, works. In Section VIII we introduce a simple random node selection algorithm, which we use for comparing our system in the following evaluation in Section IX. In Section X we give an overview of the related work in traffic offloading, before Section XI concludes this work.

II. System Model

For reasons of scalability, the geographical covered area of TOMP is subdivided into adjacent and non-overlapping service areas $SA_1, \cdots, SA_n$. Each service area SA_i has a responsible server S_i that operates in the infrastructure. A service area can contain an arbitrary number of mobile nodes. We assume that a server always knows the set of mobile nodes that are currently located in its service area by some node registration mechanism (e.g. using the one introduced by Farrell et al. [10]). Moreover, the system provides a central input interface, which is described in Section IV in more detail. An overview of the system components is depicted in Figure 1.

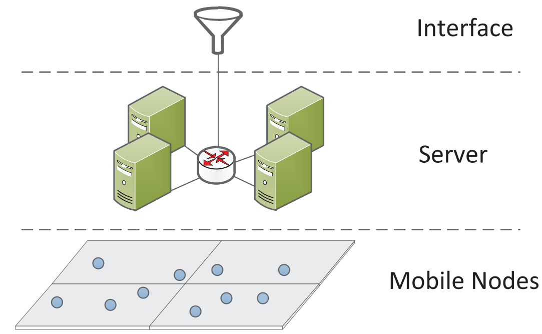


Fig. 1. System Components

A mobile node is equipped with a GPS sensor for determining its current position and speed. It is carried by a person that moves according to an underlying road graph. Neither movement direction nor speed of that person can be influenced by the system. The servers are connected with each other through a fast broadband communication network. A server can communicate with the mobile devices in its service area via a cellular network. To estimate the delay of this network, we introduce a parameter τ_m that describes the estimated message delivery time for message m. This parameter describes the time span from the start of sending m via the cellular network until successfully receiving m on a mobile devices. Note that τ_m gives only an estimation on the message delivery time since in general no real-time guarantees for the message delivery time can be provided in cellular networks. To get a feasible estimation for τ_m, we refer to Section IX in which we determine τ_m for a concrete scenario.

Mobile nodes can use ad-hoc communication (e.g. Wi-Fi Direct or Bluetooth) to exchange data with each other. The range of this communication is limited to r_{adhoc}. We assume that the size of a message that should be delivered with TOMP is limited in such a way that it can be exchanged during the meeting time of two mobile nodes. This highly depends on the particular technology that is used for the ad-hoc communication (for more details see Section IX). We assume that all devices in the system cooperate by running a corresponding app that manages the opportunistic data forwarding. For instance, the willingness for cooperation can be achieved by providing incentives by the cellular service provider, who is interested in avoiding overloaded cellular networks.

III. Problem Statement

The problem that TOMP tackles can be described as follows. Given is a message m, a message delivery time t_d and a

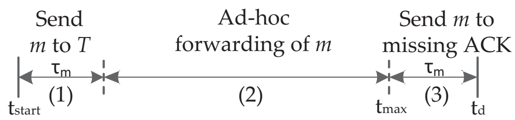


Fig. 2. Three Phases of Message Delivery

set of mobile nodes N. Message m should be delivered to all nodes $n \in N$ before time t_d. As real-time guarantees for the message delivery time cannot be provided, t_d is not a strict deadline, i.e. delays are tolerated but should be avoided. A mobile node can receive m either directly from the server via cellular communication or from another node via ad-hoc communication. The goal of our system is to provide an algorithm that ensures that all nodes receive m while the amount of cellular traffic is minimal and the delay of m with respect to t_d is minimal. In the remainder, we are going to present different algorithms to tackle this problem. We start by presenting a straightforward algorithm, before we introduce the optimized algorithms we developed for TOMP.

IV. Naive Message Delivery

First, we introduce a naive message delivery approach. TOMP takes inputs of the format $\langle m, N, t_d \rangle$ via its input interface. The input interface is deployed in the infrastructure and publicly accessible (e.g. realized through a web service). For this and all the following approaches, we assume that all $n \in N$ are located in the same service area and the request can be processed by a single server, referred to as S_i. Nevertheless, all the presented concepts can be extended straightforward to deal with the case in which multiple servers are involved.

Upon receiving an input message $\langle m, N, t_d \rangle$, the system assigns the message to server S_i. The subsequent message delivery process is different in each approach. In the naive approach, after S_i has received $\langle m, N, t_d \rangle$ it immediately sends m individually to all $n \in N$ via cellular communication. Note that this approach corresponds to a conventional message delivery, which is applied in cellular networks nowadays. Therefore, this approach will serve as a reference for the following approaches.

V. Basic Opportunistic Forwarding

To reduce the amount of cellular traffic compared to the naive approach, TOMP uses opportunistic message forwarding. We say a message m is opportunistically forwarded if a node $n_i \in N$ successfully sends m via ad-hoc communication to another node $n_j \in N$, which did not receive m before. While the message delivery in the naive approach only consists of a single step in which m is sent to all $n \in N$, the message delivery with opportunistic forwarding can be divided into three phases that are introduced subsequently.

A. Three Phases of Message Delivery

Upon receiving an input message, the server S_i initiates the following message delivery process (see Figure 2):

1) S_i queries each $n \in N$ for its position and chooses a target set $T \subseteq N$ (see Section VII). This point in time is indicated as t_{start}. Subsequently, the server sends m along with value t_d to each $n \in T$.
2) Upon receiving m from S_i, a node sends an ACK message to S_i. The node starts forwarding m opportunistically to all nodes $n \in N$ it encounters until time t_d. Each node $n \in N$ that receives m sends an ACK to the server and also starts forwarding m.
3) The server checks after time $t_{max} = t_d - \tau_m$ if it received an ACK from all $n \in N$. If that is not true for a node n, the server sends m directly to n.

Note that Phase (3) ensures that finally all $n \in N$ receive m, which is not guaranteed by the opportunistic forwarding used in Phase (2). In comparison to the naive approach, this approach uses additional cellular messages, namely for getting the positions of the nodes in Phase (1) and for sending the ACKs in Phase (2). Unless the size of m is not very small, the size of such control messages is negligible since the savings in cellular traffic for sending m outweighs this overhead (see Section IX). Next, we investigate the message delivery process in more detail to find an optimization criterion for reducing the amount of cellular traffic.

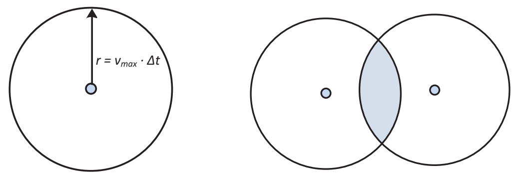
B. Optimization Criterion

For finding an optimization criterion, we have to take into account that the only system parameter that we can influence in the message delivery process is the choice of the target set $T \subseteq N$ in Phase (1). Since the composition of T has an indirect impact on the number of messages sent in Phase (3), we have to choose T in a way that the sum of cellular messages sent in Phase (1) and (3) is minimal. To investigate the relation between these phases we look in the following at the number of cellular messages sent in each phase separately.

Let $cell_1(T)$ and $cell_3(T)$ indicate the number of cellular messages for sending m in Phase (1) and (3) for a chosen target set T. Moreover, let $ah(T)$ indicate the number of successful message forwardings in Phase (2) for target set T. The total number of cellular messages $cell(T)$ that are based on the target set T can then be described as:

$$\begin{aligned} cell(T) &= \underbrace{cell_1(T)}_{|T|} + \underbrace{cell_3(T)}_{|N|-|T|-ah(T)} \\ &= |N| - ah(T) \end{aligned}$$

Note that the server sends m in Phase (3) to all $n \in N$ that are not in T and that did not receive m from another node. Since N is given, we can conclude from this formula that we need to maximize the number of message forwardings in order to minimize the number of cellular messages. The problem is that we cannot determine how many message forwardings will take place if a particular target set T was chosen since we cannot foresee the future movements of mobile nodes. As a result, conventional optimization techniques for finding an optimal target set T cannot be applied to this problem. Thus, to find a T that results in a high number of message forwardings we



(a) Possible Positions (b) Possible Meeting Points

Fig. 3. Free Space Coverage Metric

use a heuristic. In the next section, we introduce three different metrics that estimate for each pair of nodes how good the chances for a message forwarding between these nodes are. In the subsequently following Section VII, we then present an algorithm that selects a target set T that minimizes the number of cellular messages with respect to these metrics.

VI. Coverage Metrics

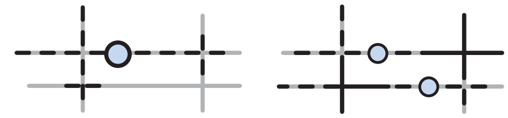
Before we present the coverage metrics, we first define the notion of coverage upon which these metrics are based. We say node n_i covers node n_j, if n_i is able to send m to n_j before time t_{max}. Note that in contrast to the notion of forwarding, coverage only states that two nodes are able to exchange m at some time before t_{max}. To quantify this coverage, we define for each pair of nodes a coverage relation that is stored in a $|N| \times |N|$ matrix, referred as coverage matrix. Each entry (n_i, n_j) in this matrix takes values from the range $[0, 1]$ and describes the confidence that n_i can cover n_j. Since the actual coverage relations cannot be foreseen, the entries of this matrix are of a probabilistic nature. For defining the coverage matrix, we introduce three different coverage metrics. The resulting matrix serves as input for the target set selection algorithm, which is introduced in the subsequent section.

A. Static Coverage

The static coverage metric analysis the coverage relations of two nodes based on their current positions. It does not take into account the future movement of the nodes and is therefore especially applicable if no information about the nodes' speed is available. The idea of the metric is to determine for each node n the set of other nodes to which n could immediately send m at the start of Phase (2). These relations are stored in the $|N| \times |N|$ matrix *s-cover*. With $dist(n_i, n_j)$ depicting the Euclidean distance between the nodes n_i and n_j, the matrix has the following entries:

$$\textit{s-cover}(n_i, n_j) = \begin{cases} 1, & \text{if } dist(n_i, n_j) \leq r_{adhoc} \\ 0, & \text{else} \end{cases}$$

Note that this metric only analyzes the nodes' positions at the start of the message delivery process and does not consider that a node may move around and cover further nodes until time t_{max}. To also include this aspect, we present in the following two mobility-based coverage metrics.



(a) Possible Positions (b) Possible Meeting Points

Fig. 4. Graph-based Coverage Metric

B. Free Space Coverage

The free space coverage metric is an extension of the static coverage metric, which also considers the possible movement of the nodes in free space. Even though we assume that the mobile nodes move according to an underlying road graph, this metric works without any knowledge about this graph. As stated in the system model, we assume that we do not know anything about the future movement of a node, i.e. neither its prospective speed nor its direction. Therefore, we have to estimate the coverage relation between two nodes on base of their current position and speed.

For the free space metric, we define the $|N| \times |N|$ matrix *fs-cover*. If node n_i and n_j can exchange a message directly at time t_{start} (i.e. *s-cover*$(n_i, n_j) = 1$), we set *fs-cover*$(n_i, n_j) = 1$. If that is not the case, we use the meeting probability of two nodes at time t_{max}, referred to as *fs-*p_{ij}, as a heuristic for their coverage relation. The higher this probability is, the higher is the chance that their movement trajectory may overlap and they can exchange m. Knowing a node's n position (x_n, y_n) and its maximum speed v_{max}, we can determine all possible positions of n at time t_{max}. These positions constitute a circle that is centered at (x_n, y_n) and has as radius the maximum distance $d_{max} = v_{max} \cdot \Delta t$ that node n can pass in time $\Delta t = t_{max} - t_{start}$ (see Fig. 3(a)). We refer to this circle as $C(n)$ and to the area of this circle as $A(C(n))$. For defining the meeting probability of two nodes n_i and n_j, we construct the respective movement circles $C(n_i)$ and $C(n_j)$. The area of intersection of these circles constitutes the possible meeting points of the two nodes (see Fig. 3(b)). Note that this area can be the empty set if the two nodes are located far away from each other. The meeting probability in free-space is given as:

$$\textit{fs-}p_{ij} = \frac{A(C(i) \cap C(j))}{A(C(i)) \cdot A(C(j))}$$

Where the numerator constitutes the number of possible common points, while the denominator denotes the number of combinations of all possible positions of n_i and n_j. As a result, we define the entries of the matrix *fs-cover* as:

$$\textit{fs-cover}(n_i, n_j) = \begin{cases} 1, & \text{if } dist(n_i, n_j) \leq r_{adhoc} \\ \textit{fs-}p_{ij} & \text{else} \end{cases}$$

Note that each matrix entry takes values from the range $[0, 1]$. Since an entry (n_i, n_j) only has value 1 if *s-cover*$(n_i, n_j) = 1$, this case implies the highest confidence in a coverage.

C. Graph-based Coverage Metric

The graph-based coverage metric follows the same idea as the free space metric, but in contrast also takes the structure of the underlying road graph into account. With the help of this additional information, we can limit the movement prediction of the mobile nodes to the road graph. This results in a better estimation for the coverage relations of the nodes.

Again, we use the meeting probability of two nodes at time t_{max} as heuristic for determining their coverage relation. Given the maximum distance $d_{max} = v_{max} \cdot \Delta t$ that a node can pass until t_{max}, we determine for every node n those points $P(n)$ on the road graph that can be reached by the node from its starting point (x_n, y_n). As a result, $P(n)$ contains all road points that are not more than d_{max} away from (x_n, y_n) when traversing the shortest path (shown as the dashed segments in Fig. 4(a)). Given the two sets $P(n_i)$ and $P(n_j)$, we can determine the set of road points $P(n_i) \cap P(n_j)$ that can be reached by both nodes n_i and n_j (shown as solid segments in Fig. 4(b)). With the help of these definitions, we define the graph-based meeting probability of the two nodes n_i and n_j as:

$$\textit{gb-p}_{ij} = \frac{|P(n_i) \cap P(n_j)|}{|P(n_i)| \cdot |P(n_j)|}$$

For determining the graph-based coverage metric we introduce the $|N| \times |N|$ matrix *gb-cover*, which has the entries:

$$\textit{gb-cover}(n_i, n_j) = \begin{cases} 1, & \text{if } dist(n_i, n_j) \leq r_{adhoc} \\ \textit{gb-p}_{ij} & \text{else} \end{cases}$$

Analogous to the free space coverage metric, a matrix entry $(n_i, n_j) = 1$ implies the highest confidence in a coverage.

VII. Target Set Selection Algorithm

Having defined the coverage metrics, the last step is to choose those nodes out of N that are most promising to reduce the total amount of cellular messages. Before we present an algorithm for this selection, we have a closer a look at the complexity of this problem.

A. Problem Analysis

As already pointed out, in order to reduce cellular traffic we have to maximize the number of ad-hoc forwardings. Since the coverage metrics give an indication if such a forwarding between two nodes is possible or not, we select those nodes for the target set that are most promising to result in a high number of forwardings. To analyze the complexity of this selection, we first have a look on the target set selection when using the matrix *s-cover* that is based on the static coverage metric.

Matrix *s-cover* contains only binary relations, i.e. n_i covers n_j ($(n_i, n_j) = 1$) or not ($(n_i, n_j) = 0$). In order to minimize the number of cellular traffic, the target set T should contain the minimal set of nodes that cover all other nodes, i.e.:

$$\forall n \in N, \exists n' \in T : \textit{s-cover}(n', n) = 1$$

This condition ensures that each node in N is covered by at least one node that is in T. Delivering m at time t_{start} to all $n \in T$ then means that all nodes are immediately covered and $cell(T) = cell_1(T) = |T|$. Looking at the complexity of finding such a minimal set T, we see that this problem is equivalent to the well-studied set-coverage problem [11]. Unfortunately, the set-coverage problem is known to be NP-hard, which makes the computational effort for an algorithm that always returns the optimal T unfeasible.

For the target set selection based on *fs-cover* and *gb-cover*, we face a similar problem with the difference that the respective matrix can contain all values from the range $[0, 1]$. Nevertheless, we again want to find a minimal set of nodes that covers all other nodes. Facing the same problem but dealing with more possible values, we can conclude that the aforementioned problem is a special case of this more complex problem. As a result, this problem is NP-hard as well. To tackle this problem in an efficient way, we present an intuitive greedy algorithm for the selection of T.

```
Require: cover(n_ij)
T ← ∅
CV[1...|N|] ← CALC-CV(cover(n_ij)) {// see Eq.(1)}
while ∃i : CV[i] ≥ 1 do
    n_max ← GET-MAX(CV)
    T ← T ∪ n_max
    for j = 1 → |N| do
        if cover(n_max, n_j) > 0 then
            sub ← cover(n_max, n_j)
            for i = 1 → |N| do
                cover(n_i, n_j) ← max(0, cover(n_i, n_j) − sub)
            end for
        end if
    end for
    CV[n_max] ← 0
    CV[1...|N|] ← CALC-CV(cover(n_ij))
end while
return T
```

Fig. 5. Greedy Target Set Selection Algorithm

B. Greedy Algorithm

For solving the target set selection problem, we extend the greedy set cover algorithm of Johnson [12] to deal with non-integer values and apply it to our problem. As input we use one of the coverage matrices defined in the previous section (*s-*, *fs-*, or *gb-cover*). We refer to this input as the *cover* matrix. Given the *cover* matrix, we first define the coverage value $cv(n_i)$ for a single node n_i by calculating the sum of its matrix line i:

$$cv(n_i) = \sum_{j=1}^{|N|} cover(n_i, n_j) \tag{1}$$

This coverage value constitutes the selection criterion for our greedy algorithm. The general idea of the algorithm is to greedily choose the node with highest coverage value for the target set T since this promises the most message forwardings. After selecting a node n for T, we have to adapt the coverage

	n_0	n_1	n_2	n_3	n_4	$cv(n_i)$
n_0	1	0	0	1	0.5	2.5
n_1	0.5	1	0	0	0.3	1.8
n_2	0	0	1	0.1	0.9	2.0
n_3	1	0	0.1	1	0	2.1
n_4	0	0.3	0.9	0	1	2.2

n_0 ε T

	n_0	n_1	n_2	n_3	n_4	$cv(n_i)$
~~n_0~~	~~1~~	~~0~~	~~0~~	~~1~~	~~0.5~~	~~0~~
n_1	0	1	0	0	0	1
n_2	0	0	1	0	0.4	1.4
n_3	0	0	0.1	0	0	0.1
n_4	0	0.3	0.9	0	0.5	1.7

Fig. 6. Example Matrix Manipulation

matrix, since the coverage relations of n should no longer be considered for the next selections. The single steps of the algorithm are presented in the following.

The algorithm (see Fig. 5) starts by computing the corresponding coverage value for every line of the matrix (Line 2), before it enters a loop that contains the greedy selection (Line 3–16). After having added the node n_{max} with the highest coverage value to T (Line 5), the algorithm checks which other nodes n_j are covered by n_{max} (i.e. $cover(n_{max}, n_j) > 0$). For all covered nodes n_j the algorithm subtracts the cover value of n_{max} from all entries in the column of n_j. Since this column contains the cover value of all other nodes that also cover n_j, we assume for further iterations that n_j is already cover by n_{max}. An example of this operations for a 5×5 matrix is shown in Fig. 6. Therein, n_0 it chosen to be added to T since it has the highest coverage value (left matrix). Accordingly, the respective cover values are subtracted from all other nodes (right matrix). After one iteration the coverage value of n_{max} is set to 0 to exclude it from the following greedy selections (Line 14). Furthermore, the coverage value of each line is recalculated since the matrix entries changed (Line 15). The iteration ends if there is no coverage values left that is bigger than 1. Since in every coverage metric a node covers itself (i.e. $cover(n_i, n_i) = 1$), a coverage value smaller than 1 means that the node is covered by some other node and should not be included in T.

VIII. Random Node Selection

For comparing the concepts developed for TOMP, we briefly introduce a simple algorithm that selects the target set in a probabilistic way. One basic way to do this is utilized by Han et al. [5] who randomly select k nodes from N, where k is fixed. In contrast, we argue that for a good target set selection the size of k should depend on the size of N and thus choose a more flexible solution. We define a fixed parameter $r \in (0, 1)$ as a tuning parameter. Based on r, the server randomly chooses $r \cdot |N_A|$ nodes from N_A to form T. Accordingly, smaller values of r result in a small size of the target set (i.e. $|T| << |N|$). All other operations are the same as in TOMP.

IX. Evaluation

To compare the efficiency of message delivery with the proposed coverage metrics, we implemented our system using the ns-2 network simulator. The results of these simulations are presented after the following simulation setup.

A. Simulation Setup

For simulating the movement of mobile nodes, we based our simulations on movement traces generated by the trace file generator CanuMobiSim [13]. As input, we used the road graph of the inner city of Stuttgart, which has a size of 2 km x 2 km. This road graph also constituted the base for the graph-based coverage metric. In the simulation, we set the size of message m to 1 MB, the message delivery time t_d to 300 s and the number of nodes were varied from 100 to 900.

For ad-hoc communication we simulated a Bluetooth communication with a range of 10 m. Message m can be transferred via *Bluetooth 3.0+HS* in less than one second [14], which is a suitable time for an opportunistic message exchange considering the speed of human movement. Note that the latest Bluetooth standards and *Wi-Fi-Direct* promise even higher data rates by using the *802.11n* specification, which can reach data rates up to 600 Mbps [15]. These techniques can be used for delivering messages of a much larger size.

For the cellular network, we simulated a HSDPA network. To cover the road graph, we followed the assumption from [16] and simulated 16 base stations located 500 m away from each other. Based on this, we can estimate the message delivery time for the network τ_m as follows: Assuming that all nodes $n \in N$ are uniformly distributed, each base station has to send m to $|N|/16$ nodes on average to delivery m concurrently to all nodes. Considering an average downlink bandwidth of 2.5 Mbps [16] for each base station, a base station can send m of size 1 MB to all its nodes in time $\tau_m = |N|/16 \cdot (8\,Mbit/2.5\,Mbps) = |N|/5\,s$. For instance, running a simulation with $|N| = 900$, we set $\tau_m = 180$. Note that this is only a best case estimation, which is a priori needed for defining the start of Phase (3). Since nodes are in general not uniformly distributed, it is possible that a base station has to send m to more than $|N|/16$ nodes. This will result in a larger message delivery time than in the best case estimation.

For the evaluation, we investigate the subsequently listed approaches:

1) NAIVE – Using the naive approach (see Section IV).
2) RAND – Using the random cellular offloading approach presented in Section VIII. We set the tuning parameter to the value of $r = 0.1$, which turned out to be the best after running several simulations.
3) S-COVER – Using the TOMP system with the static coverage metric (see Section VI-A).
4) FS-COVER – Using the TOMP system with the free space coverage metric (see Section VI-B).
5) GB-COVER – Using the TOMP system with the graph-based coverage metric (see Section VI-C).

As already mentioned, the naive approach works without position messages and ACK messages that are used in the other approaches. Since the size of such ACK messages is negligible in case of a message sizes of 1 MB, we do not consider them in the following analysis. Therefore, we only analyze the number of cellular messages that were sent in each of the approaches and not the amount of cellular data traffic.

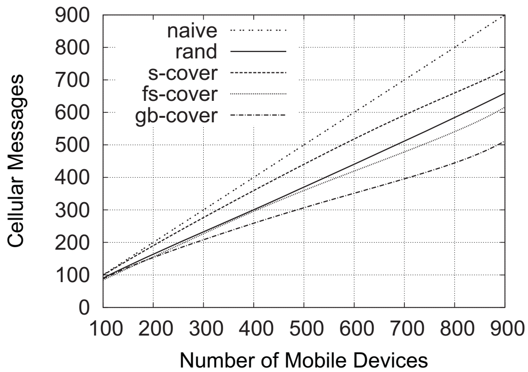


Fig. 7. Total Number of Cellular Messages

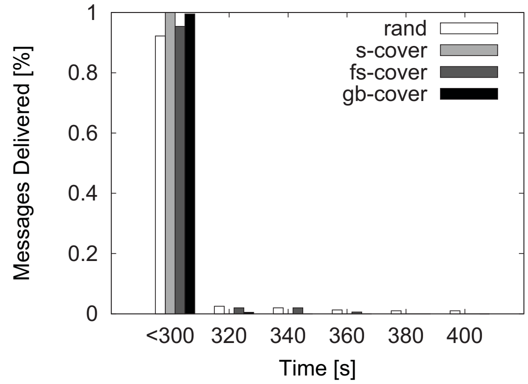


Fig. 8. Message Delay

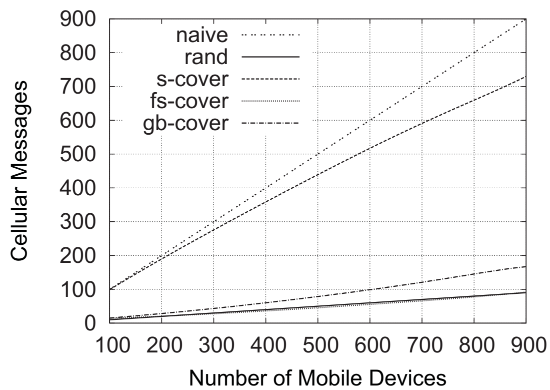


Fig. 9. Messages in Phase (1)

This has the advantage that the amount of cellular traffic for other message sizes can be derived from the following results.

B. Simulation Results

At first, we analyze the number of cellular messages that were sent in each of the approaches. In a second step, we check how many messages arrived before time t_d. Finally, we look at the number of cellular messages that were sent in the different phases of message delivery individually.

1) Total Number of Cellular Messages: We see from Fig. 7 that especially for a large number of nodes the graph-based approach uses the least amount of messages. Compared to the naive approach it saves up to 40% of cellular messages and therefore clearly underlines the advantages for using TOMP. Even the free space coverage approach, which can be used if the road graph is unknown, saves up to 30% compared to the naive approach. Furthermore, we see that the static coverage approach performs better than the naive approach but not better than the random approach. This stems from the fact that in the random approach the nodes can distribute m until time t_{max}, while with the static coverage metric m is directly delivered to all $n \in N$ at time t_{start}. As a result we can conclude that it is better to fully utilize the whole time until t_{max} to distribute m instead of statically calculating the coverage relations.

2) Message Delay: For analyzing the message delay we set the number of nodes to 600. In Fig. 8 we see for each approach the percentage of messages that were delivered to the mobile nodes within the time depicted on the x-axis. For all approaches more than 90% of the messages are delivered before time t_d. Messages that arrive after t_d are delayed due to the congestion on the cellular network in Phase (3) of the message delivery. While the random and the free space approach come with some delay for a small percentage of messages, the graph-based approach comes with almost no delay ($< 1\%$ of the messages) and the static coverage delivers all messages before time t_d. To understand these results, we analyze in the following the number of messages send in each phase of the message delivery process separately.

3) Number of Cellular Messages by Phase: First, we look in Fig. 9 at the number of cellular messages that are sent in Phase (1), i.e. the messages that are sent to the target set. While the naive and the static coverage approach choose the largest target set, out of the three other approaches the graph-based coverage approach delivers to the largest target set. At first, this looks quite surprising because in total this approach causes the smallest number of cellular messages, as seen before. If we also take Fig. 10 into account, we can see that for using this approach the number of ad-hoc messages that are sent is much higher than in the other approaches. Furthermore, from Fig. 11 we can see that this leads to a minimal number of messages that are sent in Phase (3). From this we can conclude that with the help of the graph-based approach a better target set can be chosen, which maximizes the number of ad-hoc message exchanged. Even if the number of cellular messages in Phase (1) is higher than for the other approaches, this pays off in Phase (3) since most of the receiver nodes have already been reached via ad-hoc messages. Again, the free space approach lies between the graph-based metric and the random approach. Therefore, the metric does not represent the coverage relations of the nodes as appropriate as the graph-based metric but performs better than a random node selection.

From Fig. 11 we can also conclude why the graph-based approach does not result in a very high delay in contrast to the random or free-space approach (as depicted in Fig. 8). This can be explained by the fact that the graph-based approach sends

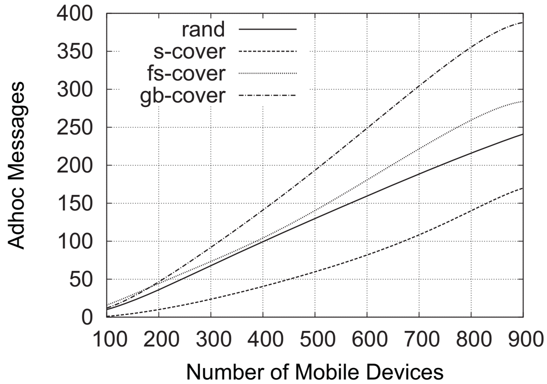


Fig. 10. Messages in Phase (2)

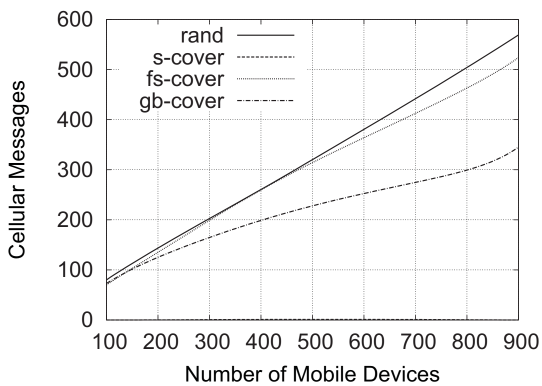


Fig. 11. Messages in Phase (3)

less cellular messages in Phase (3) and therefore reduces the chances for a congestion on the cellular network. As result, almost all of the messages arrive in time.

C. Discussion

As a conclusion for our simulations, we can say that every coverage metric introduced by TOMP significantly lowers the number of cellular messages compared to the naive approach. Furthermore, we showed that TOMP can use information about the road graph efficiently to lower the number of cellular messages. The comparison with the random algorithm also showed that it is better to fully utilize the whole message delivery time t_{max} instead of delivering all message immediately (as the static coverage approach). Furthermore, the delay in message delivery for all approaches is rather minimal or not relevant at all.

X. Related Work

Before we conclude this work, we compare TOMP with current research on cellular traffic offloading. One of the first works that came up with a concept to use local-area networks to unburden data from the cellular network was proposed by Balasubramanian et al. [17]. Predicting the probability of the future connectivity of a mobile device to Wi-Fi hotspots, they decide whether data that is intended to be transferred via the cellular network can be delayed until a Wi-Fi connection is available. A similar approach is described by Dimatteo et al. [18]. In addition to [17], they showed that by a certain number of available Wi-Fi access points certain quality of service requirements for data delivery can be provided. For distributing data from Wi-Fi access points to mobile devices, Ristanovic et al. [4] present a concept using so called *HotZones*. These are areas that are covered by Wi-Fi hotspots and are frequently visited by mobile devices. Instead of communicating data directly to the mobile devices, data is communicated via Wi-Fi in these HotZones. All the aforementioned approaches rely on the availability of publicly accessible Wi-Fi hotspots, which are not required in opportunistic traffic offloading. For instance, to upload data to the infrastructure, Thilakarathna et al. [3] propose the MobiTribe framework. Therein, mobile devices opportunistically replicate content that is intended for an upload to other mobile devices that have more energy left. More similar to TOMP, the following approaches consider traffic offloading by distributing data from the infrastructure to a target set of mobile nodes, which then forward the message via ad-hoc communication. For instance, Li et al. [19] look at the distribution of multiple messages to a set of mobile nodes. Assuming that nodes can only store a limited number of messages, they formulate an optimization problem. The goal is to find an optimal set of nodes that can store the messages and are likely to contact many other nodes. Since they focus on solving the storage problem, they assume that the messages forwarding between the nodes follows a Poisson process. Therefore, they do not utilize any position related information about the nodes. In contrast, Whitbeck et al. [20] and Han et al. [5] propose target set selection strategies that utilize available information about the nodes. Whitbeck et al. [20] use information about the node density to identify the nodes for the target set. Since node density does not indicate coverage relations, which are needed for defining the target set size, they fix the size of the target to a static value that is set a priori. As a result, there is no relation between the size of the target set and the meeting probabilities of the nodes, as in TOMP. Han et al. [5] also take the meeting probabilities of two nodes into account but they derive it from the social history of the nodes, which is not needed in TOMP.

XI. Conclusion & Future Work

Facing the problem of overloaded cellular networks, we introduced TOMP and showed that it can significantly reduce cellular traffic load. By introducing different coverage metrics we presented a range of different offload strategies, which solely require information about the nodes' position and speed.

In future work we will investigate an adaptive message delivery mechanism. So far TOMP makes an a priori decision about the selection of the target set for Phase (1). We will investigate if it is also beneficial to dynamically update the coverage metrics to the state of the message delivery (indicated by the ACK messages). The server could use this information to send additional cellular messages in Phase (2), which could help to enhance the ad-hoc message distribution.

References

[1] *Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 20102015*. Cisco Systems, 2011.

[2] *Traffic and Market Data Report - On the Pulse of the Networked Society*. Ericsson Corporate Public & Media Relations, 2011.

[3] K. Thilakarathna, L. H. Petander, and A. Seneviratne, "Performance of content replication in mobitribe: a distributed architecture for mobile ugc sharing," in *Proc. of the 36th IEEE Conf. on Local Computer Networks (LCN)*, 2011, pp. 554–562.

[4] N. Ristanovic, J.-Y. Le Boudec, A. Chaintreau, and V. Erramilli, "Energy efficient offloading of 3g neworks," in *Proc. of the 8th IEEE Intern. Conf. on Mobile Ad-Hoc and Sensor Systems (MASS)*, 2011, pp. 202–211.

[5] B. Han, P. Hui, V. A. Kumar, M. V. Marathe, G. Pei, and A. Srinivasan, "Cellular traffic offloading through opportunistic communications: a case study," in *Proc. of the 5th ACM Workshop on Challenged networks*, 2010, pp. 31–38.

[6] K. Lee, I. Rhee, J. Lee, Y. Yi, and S. Chong, "Mobile data offloading: how much can wifi deliver?" *SIGCOMM Comput. Commun. Rev.*, vol. 41, pp. 425–426, August 2010.

[7] P. Baier, F. Dürr, and K. Rothermel, "PSense: Reducing Energy Consumption in Public Sensing Systems," in *Proc. of the 26th IEEE Intern. Conf. on Advanced Information Networking and Applications*. IEEE, March 2012, pp. 136–143.

[8] P. Baier, H. Weinschrott, F. Dürr, and K. Rothermel, "MapCorrect: automatic correction and validation of road maps using public sensing," in *Proc. of the 36th Conf. on Local Computer Networks*, 2011.

[9] D. Philipp, F. Dürr, and K. Rothermel, "A sensor network abstraction for flexible public sensing systems," in *Proc. of the 8th Intern. Conf. on Mobile Ad-hoc and Sensor Systems*, 2011.

[10] T. Farrell, R. Lange, and K. Rothermel, "Energy-efficient Tracking of Mobile Objects with Early Distance-based Reporting," in *Proc. of the 4th Intern. Conf. on Mobile and Ubiquitous Systems: Networking and Services*, 2007, pp. 1–8.

[11] U. Feige, "A threshold of ln n for approximating set cover," *J. ACM*, vol. 45, pp. 634–652, July 1998.

[12] D. S. Johnson, "Approximation algorithms for combinatorial problems," in *Proc. of the 5th Annual ACM symposium on Theory of Computing*, 1973, pp. 38–49.

[13] I. Stepanov, P. J. Marron, and K. Rothermel, "Mobility modeling of outdoor scenarios for manets," in *Proc. of the 38th Symposium on Simulation*, 2005, pp. 312–322.

[14] *Bluetooth Enhanced Data Rate (EDR): The Wireless Evolution*. Agilent Technologies, 2006.

[15] *802.11n: Next-Generation Wireless LAN Technology*. Broadcom Corporation, 2006. [Online]. Available: http://www.broadcom.com/collateral/wp/802_11n-WP100-R.pdf

[16] H. Ruser, P. Halvorsen, C. Griwodz, and B. Hestnes, "Performance measurements and evaluation of video streaming in hsdpa networks with 16qam modulation," in *Multimedia and Expo, 2008 IEEE International Conference on*, 23 2008-april 26 2008, pp. 489 –492.

[17] A. Balasubramanian, R. Mahajan, and A. Venkataramani, "Augmenting mobile 3g using wifi," in *Proc. of the 8th Intern. Conf. on Mobile Systems, Applications, and Services*, 2010, pp. 209–222.

[18] S. Dimatteo, P. Hui, H. Bo, and V. Li, "Cellular traffic offloading through wifi networks," in *Proc. of the 8th IEEE Intern. Conf. on Mobile Ad-Hoc and Sensor Systems (MASS)*, 2011, pp. 192–201.

[19] Y. Li, G. Su, P. Hui, D. Jin, L. Su, and L. Zeng, "Multiple mobile data offloading through delay tolerant networks," in *Proc. of the 6th ACM Workshop on Challenged Networks*, ser. CHANTS '11. New York, NY, USA: ACM, 2011, pp. 43–48.

[20] J. Whitbeck, Y. Lopez, J. Leguay, V. Conan, and M. D. de Amorim, "Relieving the wireless infrastructure: When opportunistic networks meet guaranteed delays," *CoRR*, 2010.