Sabrina Müller, Hussein Al-Shatri, Matthias Wichtlhuber, David Hausheer and Anja Klein, ”Computation Offloading in Wireless Multi-Hop Networks: Energy Minimization via Multi-Dimensional Knapsack Problem,” in *Proc. of the 25th International Symposium on Personal, Indoor and Mobile Radio Communications - (PIMRC 2015)* , Hong Kong, China, August 2015.

©2015 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this works must be obtained from the IEEE.

Computation Offloading in Wireless Multi-Hop Networks: Energy Minimization via Multi-Dimensional Knapsack Problem

Sabrina Müller*, Hussein Al-Shatri*, Matthias Wichtlhuber†, David Hausheer†, Anja Klein*
*Communications Engineering Lab, TU Darmstadt, Germany, {s.mueller, h.shatri, a.klein}@nt.tu-darmstadt.de
†Peer-to-Peer Systems Engineering Lab, TU Darmstadt, Germany, {mwichtlh, hausheer}@ps.tu-darmstadt.de

Abstract—**Computation offloading is an upcoming approach to increase battery life of mobile devices overburdened by resource-consuming applications. In multi-hop networks, computation offloading poses new challenges since intermediate devices are required to relay tasks of others along the path to the server. The decision of a device about whether to offload or not depends thus on the provided energy of relay devices and on the decisions of other offloading devices since relay resources need to be shared. This also implies that for energy minimization, optimal decisions are topology-dependent. This paper introduces a novel theoretical framework for energy minimization of computation offloading in multi-hop wireless networks which formulates the energy minimization problem as a binary linear problem. Proving its equivalence to a multi-dimensional knapsack problem allows us to specify a greedy heuristic, which shows very good performance, with a maximal deviation of less than 5% from the optimal results. From simulations and analytical results for different topologies, we derive under which conditions computation offloading in multi-hop networks is beneficial.**

I. Introduction

In recent years, mobile devices utilize many computation-intensive and heavily energy-consuming applications, such as video processing, voice recognition or gaming. Executing such applications can be challenging since mobile devices in general are less resourceful than static devices [1] because of their lower processor capabilities and limited battery life. In fact, studies indicate that for users, longer battery life is the most desired characteristic of mobile devices [2].

A possible solution to extend battery life of mobile devices is *computation offloading* [3], i.e., migrating computation-intensive tasks to a resourceful remote server. For this purpose, the necessary amount of data to remotely process the tasks is transmitted to the server.

Computation offloading is usually considered in single-hop networks where devices are directly connected to the server to utilize its computation resources. However, coverage in single-hop networks is limited and transmission may require high power. Since multi-hop can extend coverage and reduce required transmission power [4], we consider multi-hop networks in the context of computation offloading. Specifically, we investigate the problem of energy minimization.

A lot of work has already been done on computation offloading (for a survey, see [3]), including the design of mechanisms to decide whether and which parts of an application to offload. The majority of these works focuses on energy savings in single-hop networks from a single mobile device's point of view. In [2] and [5], settings are investigated under which computation offloading is beneficial for a mobile device. Their calculations show that energy savings depend on the ratio of "communication vs. computation". Especially applications with high computational requirements but low amount of data to be transmitted are suitable. In [6], a timeout scheme is presented allowing mobile devices to save energy by offloading parts of their computation. In [7], an analytical solution is given for minimizing the consumed energy by optimally configuring the clock-frequency and the rate for transmission over time. In [8], energy savings for a mobile device are achieved by deciding which components of a software should be executed remotely using Lyapunov optimization. In [9], a policy for energy-optimal remote processing in a client-server system based on Markov models is proposed.

In comparison, there exists less work on the dynamics among several mobile devices performing computation offloading, resulting e.g. from traffic induced by computation offloading or the competition for shared resources. In [10], a game-theoretic mechanism for a computation offloading system is introduced which models energy and time costs of each single device based on the decisions of others. In [11], energy minimization in a computation offloading system is investigated, where mobile devices can choose between several servers. While these previous works on multiple devices consider single-hop networks, a multi-hop scenario has only been considered in a slightly different context of "communication vs. computation". In multi-media sensor networks, data can be compressed at sensor nodes before communicating it to a central entity in a multi-hop fashion [12], [13].

Compared to single-hop networks, computation offloading poses new major challenges in multi-hop networks. If a mobile device decides to offload a computation task to the server, other mobile devices may have to serve as relay. However, these relays dispose of limited batteries and have own computation tasks as well. If a mobile device aims at minimizing its energy consumption, the optimal decision thus also depends on the provided resources of relay devices. Moreover, since several devices might have relay devices in common, such

978-1-4673-6782-0/15/$31.00 ©2015 IEEE

resources have to be shared.

Addressing these issues, the contributions of this paper are as follows. We introduce a novel theoretical framework for energy minimization of computation offloading in multi-hop wireless networks. We formulate the problem as a binary linear program. By proving its equivalence to a multi-dimensional knapsack problem, we are able to apply the well-known primal greedy heuristic for knapsack problems to our problem, which shows very good performance. We also derive conditions under which computation offloading in multi-hop networks is beneficial by comparing numerical and analytical results for different topologies.

The rest of the paper is organized as follows. We describe the system model in Section II and introduce the energy minimization problem in Section III. In Section IV, we provide an analysis of the optimization problem. We give algorithms for its solution in Section V and include analytical results for special topologies in Section VI. Finally, we present numerical results in Section VII and conclude the paper in Section VIII.

II. System Model

In this section, we introduce the system model of our framework for computation offloading in a multi-hop network.

A. Network Model

We consider an ad-hoc wireless multi-hop network consisting of $n \geq 2$ mobile devices, labeled as *nodes* 1 to n. Additionally, there is a static *server* connected to a stable energy supply, which is capable of parallel task processing. We consider a computation offloading session which takes place in a small period of time. The location of devices can change in between different sessions, but is assumed to be fixed during one computation offloading session. With each computation offloading session, a routing table for access to the server is associated depending on the current positions of the nodes and the channel conditions they face. In such a routing table, nodes close to the server have direct access to the server, while other nodes can access the server only via a unique multi-hop route given in the routing table. A graph is used to represent the network with fixed routing table, see Figure 1. Since each node has a unique route to the server, the corresponding graph is a rooted tree whose root represents the server. The internal nodes of this tree are called *relay nodes* and the set of all relay nodes is called R. By basic tree properties [14], the number $m := |R|$ of relay nodes satisfies $m \leq n-1$. The leaf nodes of the tree are called *non-relay nodes*. We refer to the route R_i of node i, $i = 1, ..., n$, to the server as the sequence of relay nodes on the multi-hop path to the server. All nodes on route R_i are called node i's *predecessors*. The direct predecessor of node i is called node i's *parent*. All nodes for which node i is a predecessor, are called node i's *successors*.

B. Task Model and Decision of Nodes

In one computation offloading session, each node i has a non-splittable task suitable for computation offloading given by an amount of L_i CPU cycles to be processed. The node can either compute the task locally or, if enough resources are available at relay nodes, it can use computation offloading by transmitting the task via multi-hop route R_i to the server for remote processing. For this, a corresponding amount of B_i bits necessary for remote processing is sent. The ratio $\frac{B_i}{L_i}$ is application-specific. For exemplary ratios, see [5]. Node i's decision is denoted by the binary variable α_i with

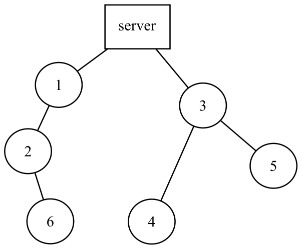


Figure 1. A sample network with 6 nodes in graph representation. Nodes 1, 2 and 3 are relay nodes, i.e., $R = \{1, 2, 3\}$. Nodes 4, 5 and 6 are non-relay nodes. The route from node 6 to the server is $R_6 = (2, 1)$, i.e., nodes 2 and 1 are predecessors of node 6. Node 2 is the parent of node 6. Since node 3 has no predecessor node, the route from node 3 to the server is $R_3 = \emptyset$. The successors of node 3 are nodes 4 and 5.

$$\alpha_i := \begin{cases} 1, & \text{if node } i \text{ transmits to the server} \\ 0, & \text{if node } i \text{ computes locally.} \end{cases}$$

The vector of all decisions is given by $\alpha := (\alpha_i)_{i=1,...,n}$. We denote by $S_{\mathrm{T}} := \{i : \alpha_i = 1\}$ the set of nodes transmitting their task to the server and by $S_{\mathrm{C}} := \{i : \alpha_i = 0\}$ the set of nodes computing locally. In one computation offloading session, after decision-making, the nodes in S_{T} transmit the tasks to the server using some given scheduling which avoids interference. The server then processes the tasks in parallel and sends the result back to the nodes.

C. Energy Consumption for Task Processing and Transmission

For local computing at node i, depending on the processor speed M_i (in cycles / s) and the processing power $P_{\mathrm{C},i}$ (in W), the energy per CPU cycle $e_{\mathrm{C},i}$ (in J / cycle) is given by

$$e_{\mathrm{C},i} = \frac{P_{\mathrm{C},i}}{M_i}. \tag{1}$$

For transmission of node i, let $P_{\mathrm{T},i}$ (in W) be the transmit power at node i. Let b_i be the bandwidth of node i, let $|h_i|^2$ be the channel gain from node i to its parent and let σ_i^2 be the noise power at node i. Then, by Shannon's formula, the energy per bit $e_{\mathrm{T},i}$ (in J / bit) node i consumes is

$$e_{\mathrm{T},i} = \frac{P_{\mathrm{T},i}}{b_i \log_2 \left(1 + \frac{P_{\mathrm{T},i} \cdot |h_i|^2}{\sigma_i^2}\right)}. \tag{2}$$

If node i computes its task locally, it consumes energy of the amount

$$E_{\mathrm{C},i} = e_{\mathrm{C},i} L_i. \tag{3}$$

If node i performs computation offloading, to transmit its own task to its parent, node i consumes energy of the amount

$$E_{\mathrm{T},i}^i = e_{\mathrm{T},i} B_i. \tag{4}$$

In addition, also any predecessor node $j \in R_i$ consumes energy when relaying node i's task. Node j's consumed energy is given by

$$E^j_{\mathrm{T},i} = e_{\mathrm{T},j} B_i. \tag{5}$$

The total energy spent in the network if node i performs computation offloading is hence given by

$$E_{\mathrm{T},i} = E^i_{\mathrm{T},i} + \sum_{j \in R_i} E^j_{\mathrm{T},i}. \tag{6}$$

As in [10], we neglect the feedback link from server to node, as the amount of feedback information is small for many applications.

Each node in the network has limited energy resources. For one computation offloading session, let $E_{\mathrm{prov},i}$ (in J) be the energy node i provides for this session. While the values $E_{\mathrm{C},i}$ and $E_{\mathrm{T},i}$ result from topology and task, in general $E_{\mathrm{prov},i}$ can be arbitrarily chosen. Our framework assumes that each node provides at least enough energy to handle its own task, i.e.,

$$E_{\mathrm{prov},i} \geq \max(E_{\mathrm{C},i}, E^i_{\mathrm{T},i}) \text{ for all } i = 1, ..., n. \tag{7}$$

III. Formulation of Energy Minimization Problem

In this section, we introduce the energy minimization problem for computation offloading in multi-hop networks. We start with determining the total energy $E_{\mathrm{net}}(\alpha)$ spent in the network for local computing and transmission to the server. It depends on the decisions α of all nodes and can be calculated by the following linear cost function:

$$\begin{aligned} E_{\mathrm{net}}(\alpha) &= \sum_{i=1}^{n} (\alpha_i E_{\mathrm{T},i} + (1-\alpha_i) E_{\mathrm{C},i}) \\ &= \sum_{i=1}^{n} \alpha_i (E_{\mathrm{T},i} - E_{\mathrm{C},i}) + \sum_{i=1}^{n} E_{\mathrm{C},i}. \end{aligned} \tag{8}$$

Note that the last term $\sum_{i=1}^{n} E_{\mathrm{C},i}$ is independent of α. Hence, when solving the optimization problem with objective function $E_{\mathrm{net}}(\alpha)$, this constant has no influence on the optimal decisions and can be neglected for optimization.

To formulate the energy minimization problem, our framework takes into account that the nodes' limited energy resources impose constraints on possible decisions. For each leaf node $j \notin R$, its limited energy resources only have to be sufficient for the processing or transmission of its own task. By assumption (7) this is always the case. For each relay node $j \in R$, assuming that the node is selfish, it wants to ensure that its standard choice of local computing is always possible, regardless of the decisions of its successor nodes. Therefore, if node j reserves an amount of $E_{\mathrm{C},j}$ for itself, it can spend $E_{\mathrm{prov},j} - E_{\mathrm{C},j} \geq 0$ for relaying tasks of successor nodes. The following linear constraint describes that the decisions of successor nodes are restricted by the energy node j provides:

$$\sum_{\{i : j \in R_i\}} \alpha_i E^j_{\mathrm{T},i} \leq E_{\mathrm{prov},j} - E_{\mathrm{C},j}. \tag{9}$$

Employing the network's cost function (8) as objective function and taking into account the energy constraints (9) at relay nodes, the optimal decisions for energy minimization can be obtained by solving the following binary linear program:

$$\begin{aligned} \min \ & \sum_{i=1}^{n} \alpha_i (E_{\mathrm{T},i} - E_{\mathrm{C},i}) \\ \text{s.t.} \ & \sum_{\{i : j \in R_i\}} \alpha_i E^j_{\mathrm{T},i} \leq E_{\mathrm{prov},j} - E_{\mathrm{C},j} \text{ for } j \in R \\ & \alpha_i \in \{0, 1\} \text{ for } i = 1, ..., n. \end{aligned} \tag{10}$$

Here, the number of constraints corresponds to the number of relay nodes $m = |R|$ and is hence topology-dependent.

IV. Analysis of Optimization Problem

In this section, we derive properties of problem (10).

A. Feasibility

In contrast to general discrete optimization problems, it is easy to find a feasible solution of problem (10). The choice "all nodes compute locally" is always feasible, as can be checked by inserting $\alpha = (0, ..., 0)$ into (10).

B. Decomposition

Problem (10) is decomposable to specific parts of the network. We divide the original tree into subtrees, each of which consists of one child node of the server and all its successor nodes. For example, the network in Figure 1 is partitioned into a subtree of nodes 1, 2 and 6 and a subtree of nodes 3, 4 and 5. Then, problem (10) can be solved on each of these subtrees separately since only decisions of nodes on the same subtree are coupled by energy constraints of common relay nodes.

C. Variable Reduction

Certain variables of problem (10) may directly be fixed to their optimal values, see [15]. By applying the two rules given below, the problem size may be reduced in a pre-processing step before solving problem (10). The structure of the problem remains the same.

- Rule 1: If any node i has $E_{\mathrm{T},i} \geq E_{\mathrm{C},i}$, then the node should compute locally, because transmitting to the server would not decrease the consumed energy in the network and possibly consumes additional energy at relay nodes. Thus, the optimal decision for such a node i is $\alpha_i = 0$.
- Rule 2: If any relay node $j \in R$ consumes his total energy resources for computing locally, i.e., $E_{\mathrm{prov},j} = E_{\mathrm{C},j}$, then each successor node i of relay node j has to compute locally. Thus, for each node i with $j \in R_i$ the optimal decision is $\alpha_i = 0$.

D. Multi-Dimensional Knapsack Formulation

Next, we prove that problem (10) is equivalent to a specific type of binary linear program, the multi-dimensional knapsack problem, a well-known NP-hard problem [15], [16].

Proposition 1. *After pre-processing, problem* (10) *is a multi-dimensional knapsack problem.*

Proof: We turn the minimization problem (10) into its equivalent maximization problem and we rewrite the energy constraints of problem (10) to eliminate the index sets $\{i : j \in R_i\}$. For that purpose, we define trivial coefficients $E^j_{\mathrm{T},i} := 0$ for all i, j with $j \notin R_i$. Hence, problem (10) is equivalent to the following problem:

$$\begin{aligned} -\max\ & \sum_{i=1}^{n} \alpha_i (E_{\mathrm{C},i} - E_{\mathrm{T},i}) && (11) \\ \text{s.t.}\ & \sum_{i=1}^{n} \alpha_i E^j_{\mathrm{T},i} \leq E_{\mathrm{prov},j} - E_{\mathrm{C},j} \text{ for } j = 1, ..., m \\ & \alpha_i \in \{0, 1\} \text{ for } i = 1, ..., n. \end{aligned}$$

We then apply pre-processing Rules 1 and 2, see Subsection IV-C. The remaining coefficients satisfy $E_{\mathrm{C},i} - E_{\mathrm{T},i} > 0$, $E_{\mathrm{prov},j} - E_{\mathrm{C},j} > 0$ and $E^j_{\mathrm{T},i} \geq 0$. Therefore, problem (10) is a multi-dimensional knapsack problem [15], [16]. ■

V. Algorithms

In this section, we review methods to solve the energy minimization problem (10) for any type of topology.

A. Global Optimum as Benchmark

Since problem (10) is a binary linear program, an integer programming solver is applied to find its global solution, serving as benchmark for the heuristic algorithm shown below.

B. Primal Greedy Heuristic

We apply the well-known primal greedy heuristic for multi-dimensional knapsack problems [15] to problem (10). In our context, the idea of this centralized polynomial-time algorithm is as follows. After pre-processing, we bring the nodes into decreasing order of some carefully chosen efficiency measure. Then, starting from the most efficient node, we add one node at a time to the set of transmitting nodes, but only, if this does not violate any of the energy constraints at relay nodes. If an energy constraint is violated, the node is added to the set of local computing nodes. The critical point of the algorithm is the choice of the efficiency measure. Our framework uses an efficiency measure taking into account how much energy is saved when node i is chosen for transmission and which portions of energy provided by relay nodes it requires [15]:

$$\mathrm{eff}_i := \frac{E_{\mathrm{C},i} - E_{\mathrm{T},i}}{\sum_{j \in R_i} \frac{E^j_{\mathrm{T},i}}{E_{\mathrm{prov},j} - E_{\mathrm{C},j}}}. \tag{12}$$

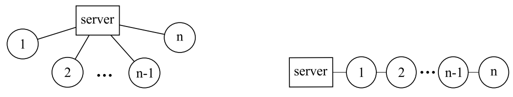
The primal greedy heuristic for problem (10) is given in Figure 2. Since problem (10) is decomposable to subtrees, cf. Subsection IV-B, the algorithm can be performed with local knowledge on each of these subtrees separately.

VI. Analytical Results for Special Topologies

Next, we investigate the energy minimization problem (10) for two special topologies, a star and a line topology. The star topology establishes the connection to computation offloading in single-hop networks. The line topology reveals the impact of a non-decomposable topology with maximum number of relay nodes on the benefit of computation offloading.

```
procedure GREEDY HEURISTIC
    Input: Problem (10)
    Pre-process (10) according to Rules 1 and 2
    Let $\alpha_1, ..., \alpha_{\tilde{n}}$ be the variables not fixed in pre-processing
    for $i = 1, ..., \tilde{n}$ do
        Initialize $\alpha_i := 0$
        $\mathrm{eff}_i := \frac{E_{\mathrm{C},i} - E_{\mathrm{T},i}}{\sum_{j \in R_i} \frac{E^j_{\mathrm{T},i}}{E_{\mathrm{prov},j} - E_{\mathrm{C},j}}}$
    end for
    Sort efficiencies $\mathrm{eff}_i$ into decreasing order and save into vector ord := indices of ordered efficiencies
    for $j = 1, ..., \tilde{n}$ do
        $\alpha_{\mathrm{ord}(j)} := 1$
        if decision vector $\alpha$ not feasible for (10) then
            $\alpha_{\mathrm{ord}(j)} := 0$
        end if
    end for
end procedure
```

Figure 2. Greedy Heuristic



(a) Star Topology (b) Line Topology

Figure 3. Special Topologies

A. Star Topology

A *star* topology is a tree with a root and n leaves, see Figure 3(a), i.e., a single-hop network. In this case, problem (10) has no constraints ($m = 0$). It can thus be decomposed to each individual node and optimally solved node-wise. The globally optimal decisions are given by

$$\alpha_i = \begin{cases} 1, & \text{if } E_{\mathrm{T},i} < E_{\mathrm{C},i} \\ 0, & \text{if } E_{\mathrm{T},i} \geq E_{\mathrm{C},i}. \end{cases} \tag{13}$$

The primal greedy heuristic always yields this global optimum, since all nodes with $E_{\mathrm{T},i} < E_{\mathrm{C},i}$ have $\mathrm{eff}_i = +\infty$.

B. Line Topology

A *line* topology is a single rooted branch whose nodes can be labeled according to their hop-distance to the server from 1 to n, see Figure 3(b). The number of relay nodes is $m = n - 1$, which gives the highest possible number of energy constraints. However, since the tree consists of one single branch, the energy constraints have a special structure, which allows to give an analytical result of problem (10) in case of a homogeneous network, where the energies per node are equal.

Proposition 2. *Consider problem* (10) *in the case of a line topology in a homogeneous network, i.e., there are constants* $E_{\mathrm{C}}, E_{\mathrm{link}}, E_{\mathrm{prov}}$ *with*

- $E_{\mathrm{C},i} = E_{\mathrm{C}}$ *for all* $i = 1, ..., n$

- $E^j_{\mathrm{T},i} = E_{\mathrm{link}}$ *for all* $i, j = 1, ..., n$ *with* $i \geq j$
- $E_{\mathrm{prov},j} = E_{\mathrm{prov}}$ *for all* $j = 1, ..., n-1$.

Then, the optimal decisions α_i*,* $i = 1, ..., n$*, are given by*

$$\alpha_i = \begin{cases} 1, & \textit{if } i < \frac{E_{\mathrm{C}}}{E_{\mathrm{link}}} \textit{ and } i \leq \frac{E_{\mathrm{prov}} - E_{\mathrm{C}}}{E_{\mathrm{link}}} + 1 \\ 0, & \textit{else.} \end{cases} \tag{14}$$

Proof: In the line topology, predecessors of node i are all nodes with smaller hop-distance, i.e., nodes 1 to $i-1$. Hence, in the homogeneous setting of Proposition 2, the energy $E_{\mathrm{T},i}$ for transmission from (6) reduces to

$$E_{\mathrm{T},i} = E^i_{\mathrm{T},i} + \sum_{j=1}^{i-1} E^j_{\mathrm{T},i} = E_{\mathrm{link}} + \sum_{j=1}^{i-1} E_{\mathrm{link}} = i \cdot E_{\mathrm{link}}. \tag{15}$$

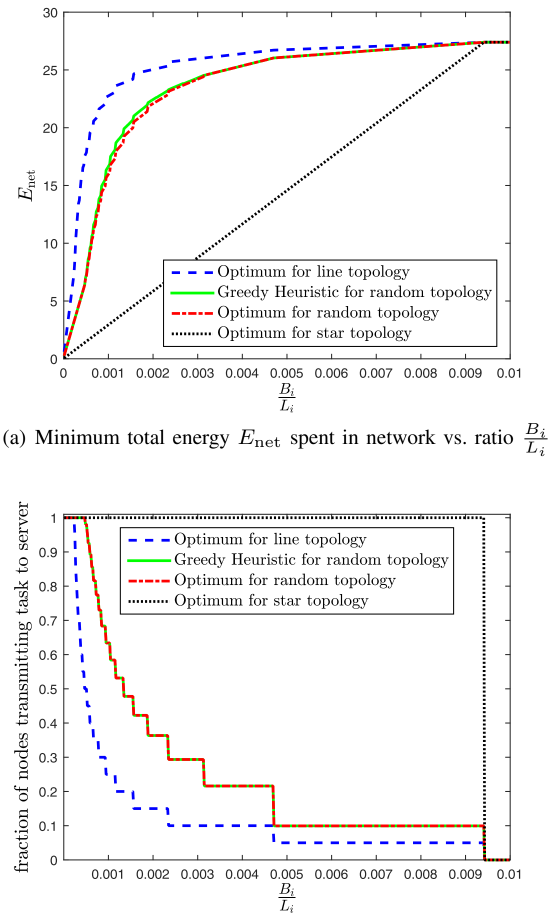
Using (15) and the fact that successor nodes of relay node j, $j = 1, ..., n-1$, are all nodes with higher hop-distance, i.e., nodes $j+1$ to n, problem (10) becomes

$$\begin{aligned} \min \ & \sum_{i=1}^{n} \alpha_i (i E_{\mathrm{link}} - E_{\mathrm{C}}) \\ \text{s.t.} \ & \sum_{i=j+1}^{n} \alpha_i E_{\mathrm{link}} \leq E_{\mathrm{prov}} - E_{\mathrm{C}} \text{ for } j = 1, ..., n-1 \\ & \alpha_i \in \{0, 1\} \text{ for } i = 1, ..., n. \end{aligned}$$

As discussed in pre-processing Rule 1, in the optimal solution only nodes with $E_{\mathrm{T},i} < E_{\mathrm{C}}$ transmit to the server. By (15), this is equivalent to $i \cdot E_{\mathrm{link}} < E_{\mathrm{C}}$ or $i < \frac{E_{\mathrm{C}}}{E_{\mathrm{link}}}$. The concept of dominance, see [15], is now used to prove which variables α_i are nonzero in the optimal solution. Translated to our context, node i dominates node k if a) node i provides at least as much energy savings when transmitting to the server as node k and b) node i needs at most as many energy resources from any relay node in the network as node k. Both conditions are satisfied if $i \leq k$ because a) then the objective values satisfy $iE_{\mathrm{link}} - E_{\mathrm{C}} \leq kE_{\mathrm{link}} - E_{\mathrm{C}}$ and b) nodes i and k need the same amount of energy from common relay nodes by homogeneity, but node i has lower hop-distance to the server than node k and thus needs energy of fewer relay nodes. Hence, each node i dominates all nodes k with $k \geq i$. This gives a dominance ordering of the nodes according to their hop-distance to the server. Starting from node 1, we set $\alpha_i = 1$ for one node after the other as long as this does not violate any of the constraints. The tightest constraint is the one of node 1 because all relay nodes have the same energy available, but node 1 has the highest number of successor nodes. Suppose nodes 1 to $i-1$ were already chosen to transmit, i.e., $\alpha_k = 1$ for $k = 1, ..., i-1$. Then, enough energy is available for node i to transmit if

$$\sum_{k=2}^{i} 1 \cdot E_{\mathrm{link}} \leq E_{\mathrm{prov}} - E_{\mathrm{C}},$$

or equivalently $i - 1 \leq \frac{E_{\mathrm{prov}} - E_{\mathrm{C}}}{E_{\mathrm{link}}}$. To sum up, this yields the optimal decision vector given in (14). ■



(a) Minimum total energy E_{net} spent in network vs. ratio $\frac{B_i}{L_i}$

(b) Fraction of nodes transmitting task to server vs. ratio $\frac{B_i}{L_i}$

Figure 4. Optimal and heuristic results of problem (10) for homogeneous line, random and star topologies

This result illustrates that in a homogeneous line topology, if not enough energy is available for all nodes to transmit to the server, at first nodes closer to the server should be selected for transmission. Note that for topologies consisting of several parallel lines to the server, problem (10) is decomposable to each of the lines by Subsection IV-B. Hence, Proposition 2 is applicable in this case as well. Moreover, one can prove that the primal greedy heuristic always yields the global optimum for homogeneous line topologies.

VII. Numerical Results

In this section, we investigate the effect of different topologies on the optimal result of problem (10). We also evaluate the greedy heuristic by comparing its solutions to the optimal results obtained using the solver Gurobi [17]. Our numerical study includes networks of $n = 20$ nodes with star, line as well as random topologies. The results for random topologies are obtained by simulating 100 random trees. We fix the following parameters homogeneously for each node i. For task sizes, as in [10] the number of CPU cycles is set to $L_i = 1000\,$Mcycles and the number of bits B_i is variable.

For energy consumption, we take the best values experimentally obtained in [5], for local computing $e_{\mathrm{C},i} = \frac{1}{730} \frac{\mathrm{J}}{\mathrm{Mcycle}}$ and for transmission $e_{\mathrm{T},i} = \frac{1}{860} \frac{\mathrm{J}}{\mathrm{kbyte}}$. This corresponds to an energy ratio of $\frac{e_{\mathrm{C},i}}{e_{\mathrm{T},i}} \approx 0.0094 \frac{\mathrm{bits}}{\mathrm{cycle}}$. Concerning energy resources, we arbitrarily assume that each node spends an additional 100% of its own required energy for relaying, i.e., $E_{\mathrm{prov},i} = 2 \max(E_{\mathrm{C},i}, E^{i}_{\mathrm{T},i})$.

Figure 4 shows optimal and heuristic results for line, star and random topology under varying ratio $\frac{B_i}{L_i}$. Results of the greedy heuristic are only shown for random topologies since the heuristic produces optimal results for star and homogeneous line topology. While Figure 4(a) gives the minimum total energy spent in the network, Figure 4(b) shows the corresponding fraction of nodes transmitting their task to the server. For all topologies, when $\frac{B_i}{L_i}$ increases, the energy spent in the system increases until it reaches a stable point at which $\frac{B_i}{L_i} = \frac{e_{\mathrm{C},i}}{e_{\mathrm{T},i}} \approx 0.0094 \frac{\mathrm{bits}}{\mathrm{cycle}}$. At the same time, the fraction of transmitting nodes decreases, until it reaches a stable point at which no node is transmitting. This results from the fact that when transmitting gets more expensive, more and more nodes will instead compute locally. The stable point is reached when for all nodes the networks' energy costs for transmission to the server are as expensive as for local computing. At this point, no node transmits its task to the server. For both random and line topologies, the steps in the graphs of Figure 4(a) reflect the impact of the relay energy $E_{\mathrm{prov},i}$. When not enough relay energy is left for one node to further perform computation offloading, it is forced to compute locally, leading to an abrupt increase of the energy spent.

Our numerical results allow three conclusions. Firstly, the results on the star topology reflect that in single-hop networks, computation offloading noticeably pays off as soon as $\frac{B_i}{L_i} < \frac{e_{\mathrm{C},i}}{e_{\mathrm{T},i}}$, compare [5]. This is because, if transmitting to the server is cheaper, all nodes perform computation offloading, cf. Subsection VI-A. For multi-hop networks, the effect of computation offloading becomes particularly apparent if $\frac{B_i}{L_i} \ll \frac{e_{\mathrm{C},i}}{e_{\mathrm{T},i}}$, especially in case of a line topology due to the high number of hops, cf. Subsection VI-B. Secondly, the effect of computation offloading strongly depends on the provided energy $E_{\mathrm{prov},i}$, as for higher $E_{\mathrm{prov},i}$, the curves for line and random topology would be flatter. Thirdly, even though the greedy heuristic for knapsack problems has no performance guarantee, it shows a very good overall performance in our scenarios, with a maximal deviation of less than 5% from the optimal results.

VIII. Conclusion

In this paper, we introduced a novel theoretical framework for the problem of energy minimization of computation offloading taking into account challenges of multi-hop networks. We showed its equivalence to a multi-dimensional knapsack problem and applied the well-known primal greedy heuristic for knapsack problems to our problem. The heuristic showed very good performance, with a maximal deviation of less than 5% from the optimal results. Our numerical as well as analytical results revealed that multi-hop networks benefit noticeably from computation offloading for highly computation-intensive applications with small amount of data to transmit. Additionally, the outcome is strongly affected by the amount of energy which nodes provide to relay for others.

While this paper focused on a centralized solution, in future work, a decentralized algorithm will be developed to be applicable to ad-hoc networks.

Acknowledgment

This work has been funded by the German Research Foundation (DFG) as part of project B03 within the Collaborative Research Center (CRC) 1053 – MAKI. The authors would like to thank Alexander Kühne (project C01), Mousie Fasil (project C01) and Alexander Frömmgen (project A02) for valuable discussions.

References

[1] M. Satyanarayanan, "Fundamental challenges in mobile computing," in *Proc. 15th Annual ACM Symposium on Principles of Distributed Computing*, 1996, pp. 1–7.

[2] K. Kumar and Y.-H. Lu, "Cloud computing for mobile users: Can offloading computation save energy?" *Computer*, vol. 43, no. 4, pp. 51–56, 2010.

[3] K. Kumar, J. Liu, Y.-H. Lu, and B. Bhargava, "A survey of computation offloading for mobile systems," *Mobile Networks and Applications*, vol. 18, no. 1, pp. 129–140, 2013.

[4] Y.-D. Lin and Y.-C. Hsu, "Multihop cellular: a new architecture for wireless communications," in *Proc. 19th IEEE INFOCOM*, vol. 3, 2000, pp. 1273–1282.

[5] A. P. Miettinen and J. K. Nurminen, "Energy efficiency of mobile clients in cloud computing," in *Proc. 2nd USENIX Conference on Hot Topics in Cloud Computing*, 2010, pp. 4–4.

[6] C. Xian, Y.-H. Lu, and Z. Li, "Adaptive computation offloading for energy conservation on battery-powered systems," in *Proc. 13th International Conference on Parallel and Distributed Systems*, 2007, pp. 1–8.

[7] Y. Wen, W. Zhang, and H. Luo, "Energy-optimal mobile application execution: Taming resource-poor mobile devices with cloud clones," in *Proc. 31st IEEE INFOCOM*, 2012, pp. 2716–2720.

[8] D. Huang, P. Wang, and D. Niyato, "A dynamic offloading algorithm for mobile computing," *IEEE Transactions on Wireless Communications*, vol. 11, no. 6, pp. 1991–1995, 2012.

[9] P. Rong and M. Pedram, "Extending the lifetime of a network of battery-powered mobile devices by remote processing: A markovian decision-based approach," in *Proc. 40th ACM Annual Design Automation Conference*, 2003, pp. 906–911.

[10] X. Chen, "Decentralized computation offloading game for mobile cloud computing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 4, pp. 974–983, 2015.

[11] Y. Ge, Y. Zhang, Q. Qiu, and Y.-H. Lu, "A game theoretic resource allocation for overall energy minimization in mobile cloud computing system," in *Proc. ACM/IEEE International Symposium on Low Power Electronics and Design*, 2012, pp. 279–284.

[12] M. Tahir and R. Farrell, "Optimal communication-computation tradeoff for wireless multimedia sensor network lifetime maximization," in *IEEE Wireless Communications and Networking Conference*, 2009, pp. 1–6.

[13] D. Maniezzo, K. Yao, and G. Mazzini, "Energetic trade-off between computing and communication resource in multimedia surveillance sensor network," in *4th International Workshop on Mobile and Wireless Communications Network*, 2002, pp. 373–376.

[14] J.-C. Fournier, *Graph Theory and Applications*. ISTE, 2010.

[15] H. Kellerer, U. Pferschy, and D. Pisinger, *Knapsack Problems*. Springer, 2004.

[16] A. Fréville, "The multidimensional 0–1 knapsack problem: An overview," *European Journal of Operational Research*, vol. 155, no. 1, pp. 1 – 21, 2004.

[17] I. Gurobi Optimization, "Gurobi optimizer reference manual," 2015. [Online]. Available: http://www.gurobi.com