Energy Maps for Large-scale, Mobile Wireless Networks

Min Kyoung Park and Volkan Rodoplu
Department of Electrical and Computer Engineering
University of California
Santa Barbara, CA 93106-9560
{mkpark, vrodoplu}@ece.ucsb.edu

Abstract—Energy planning and optimization constitutes one of the most significant challenges for high-mobility networks. This paper proposes a novel framework to share, retain and refine end-to-end energy metrics in the joint memory of the nodes, over time scales over which this information can be spread to the network and utilized for energy planning decisions. We construct maps of end-to-end energy metrics that enable energy optimization in high-mobility networks. We show how to (1) compute the spatial derivatives of energy potentials in high-mobility networks, (2) construct energy maps on-demand via path integration methods, (3) distribute, share, fuse, and refine energy maps over time by information exchange during encounters, (4) allow the nodes to use energy maps for energy planning and optimization in delay-tolerant, high-mobility networks.

I. INTRODUCTION

Energy planning and optimization constitutes one of the most significant challenges for high-mobility networks. This problem has been deemed infeasible due to the high amount of node mobility that was believed to limit the performance of protocols. A key problem in high-mobility, large-scale networks is that the paths quickly become obsolete, rendering end-to-end energy-based routing decisions very difficult. Hence, our main goal is two-fold: First, we establish a novel framework and methodology to make end-to-end information on energy metrics available in high-mobility networks. Second, we show that with the information on energy metrics, the energy optimization and planning problem becomes feasible in high-mobility networks.

We propose “energy maps”, which are maps of the end-to-end energy metrics between physical locations in space. The underlying assumption of traditional mobile routing protocols [1][2][3] is that a path is constituted by a fixed sequence of nodes, and if the nodes change, then the path is no longer there (and a path update is required). However, our insight is that even though the network might be highly mobile, and the nodes between two physical locations in space might change at a very high rate, the end-to-end energy metric between two physical points in space might still be roughly constant. (For commonly-used simulation models such as the Random Waypoint (RWP) Model [4][5] and others [6][7], due to the fact that there is a stationary node density function, the averaged end-to-end transmit energy consumption settles down to an invariant function over space.) The reason is that if the main goal is to transmit information within a certain total energy consumption, the relay nodes in between that carry the information are interchangeable. Hence, in this paper, paths are conceived of not as fixed sequences of nodes, but rather as “paths through space”. This view has already been utilized in trajectory-based routing [8], but no claims have been made there as to whether the end-to-end metrics may stay roughly constant over time between two physical locations, and this is the contribution that our work makes in that regard.

It is commonly believed that the larger the individual node velocities, the less tractable these networks become [9][10][11][12]. However, this is not the case because there are two fundamental aspects of variation that result from mobility. The first aspect is the variation that results from individual node motion, which we shall exploit to spread the energy maps. The second aspect is the rate at which the end-to-end energy metrics themselves are changing. Hence, one can separate the two components of mobility with respect to a given energy metric of interest: the individual node mobility determines how fast control information can be spread, and a separate “coherence time” of the average energy metric over space determines how fast the energy metric is changing. If the coherence time of the average energy metric is much larger than the spreading period, we have a high-mobility network for which energy maps are useful.

The energy maps are very valuable in designing energy optimization and planning protocols in high-mobility networks. In actual mobile networks, there would be good or bad positions with respect to the distance to the destination for each node as illustrated in Fig. 1. (In the figure, $t$ denotes time; $k$ denotes a cell number; and $e$ denotes the estimated end-to-end energy metric from an energy map.) Intuitively, if mobile sensors know their future trajectories, then they can reduce energy consumption by leveraging their movements around the network, since they are able to estimate the expected energy consumption for the future locations by using an energy map.

In this paper, we compare the total energy consumption of energy planning based on our energy maps, with that of the energy optimization based on global end-to-end energy metrics (which are not available in practice but constitute an upper bound on the performance).

The rest of this paper is organized as follows: in Section II, we describe our assumptions and the system model. In Section III, we discuss how to construct the energy maps via local link energy cost measurements. In Section IV, we
develop new methods for the mobile nodes to disseminate the localized measurements efficiently among themselves for fast construction of the energy potential. In Section V, we apply the energy maps to an energy optimization problem for delay tolerant, highly mobile networks. In Section VI, we discuss the implications and possible extensions of this work for the design of future wireless networks.

II. SYSTEM DESCRIPTION

In this paper, we use a specialized scenario where there is a single destination (which we refer to as a “base station”) in the middle of the deployment region. We define the “energy potential” as the average amount of total energy incurred for a bit to be sent from a fixed point in space to the base station, through a mobile network.

We make the following assumptions: (1) the network transfers small amounts of critical data over energy-limited, mobile devices. Hence, the network operates in the energy-limited regime with no data-rate-intensive traffic, (2) the forwarding operation at each node is fast enough that from the time that a packet arrives at a node to the time that the forwarded packet is put out on an outgoing port, the node moves negligibly. This assumption allows us to construct paths without modelling the physical transport of the packet with the node at which it is waiting to be forwarded, (3) for simplicity, only the transmit energy consumption is modelled, and it depends only on the path loss model (the path loss exponent, $q = 2$ in our simulations).

Throughout this paper, we shall refer to results to illustrate the main ideas that we develop. The results we refer to were all generated using the following simulation set-up: 100 mobile nodes are deployed on a square deployment region that is 1000 meters on each side. We divide the deployment region into a uniform, $20 \times 20$ grid of square 400 “cells”; hence, each cell is 50 meters on each side. We assume that each node knows its own position via a position device (such as a GPS receiver), but not the instantaneous positions of the other nodes in the network (except when explicitly communicated).

In our simulations, we use two mobility modes: the Random Waypoint model only, and the combination of the RWP and the modified contraction model [13]. We use the RWP only mode to generate a time-invariant energy potential (this is called, a “time-invariant mode”), and the combination of the RWP and the modified contraction models in order to generate a time-varying energy potential (called, a “time-varying mode”). Each node moves with a fixed velocity $v = 10$ m/s, and the pause times are zero.

For the time-varying mode, we use the contraction of node density in a network. The contraction is very useful to simulate networks having several attractors such as famous visiting points, the market areas in a city, and the lecture halls on campus. In the deployment region, we place four attractors at positions (0.275, 0.275), (0.275, 0.725), (0.725, 0.275) and (0.725, 0.725) km with attraction regions of 0.0225 km² around each. The base station is located at (0.5, 0.5) km. Outside the attraction areas, a node follows the RWP model. When a node enters into the attraction areas (that is, when a node’s destination is inside the attraction area), it moves according to the modified contraction model. In the modified contraction model which was introduced in [13], a node chooses a destination by keeping itself directed to the center of attraction area. A node strictly follows this contraction model until it arrives at the cell where the attractor is located.

We define the “sampling interval” as the time that passes from one measurement of the energy potential to the next measurement. Every node uses the same sampling interval in our design, and in our simulations, the sampling interval is 5 seconds. During this interval, since $v = 10$ m/s, every node moves 50 meters, which is equal in length to one side of a cell in our design. Hence, a node will typically have moved to a new cell.

III. CONSTRUCTION OF AN ENERGY MAP VIA LOCAL “FIELD” MEASUREMENTS

We collect the end-to-end measurements via Dijkstra’s algorithm in order to use them as a benchmark against which to measure the performance of the algorithms we shall develop. The important point is that for the Random Waypoint Model, the average for each cell converges to a cup-shaped energy potential function, which remains invariant over time. This time invariance of the energy potential is due to the fact that there is an underlying stationary node position density of the Random Waypoint Model. However, in actual networks, the energy metrics between physical locations do not remain invariant over time.

In order to explore both the time-invariant and the time-varying energy potentials over space, we use the following scenario: we set the time-varying mode for a duration that lasts only from 1000 seconds to 1500 seconds, and we use the time-invariant mode for the other durations over the whole simulation time of 2500 seconds starting from 0 seconds. During a time-varying mode, if a node arrives at the cell with an attractor (in the deployment region shown in Fig. 1), it stays there until the time-varying mode ends.

In Fig. 2, we display the node density corresponding to four different snapshots. In the beginning, for the time-invariant mode, the node density is an increasing function toward the center of the grid and there is a peak around the center, which is similar to the analytical node density of the RWP model. After 1000 seconds, the system goes into the time-varying
At time = 1000 seconds

At time = 1050 seconds

At time = 1500 seconds

At time = 1600 seconds

Fig. 2. Time-varying node densities at different time sequences.

Fig. 3. Time-varying energy metrics at different time sequences.

mode. Some nodes start converging to the cells where the attractors are located. As time passes under the time-varying mode, more nodes reach the attractors, and only a few nodes wander around outside the attraction areas and follow the RWP model. (In the simulations, we observed that 50 to 60 out of 100 nodes are gathered near the four attractors, and the rest are distributed outside.) At 1500 seconds, the time-varying mode is over, and the time-invariant mode is re-activated. At 1600 seconds, the corresponding node density is then going back to that of the RWP model only.

The energy maps highly depend on the node density of the deployment region. The energy metric in a cell is highly variable depending on the existence of intermediate nodes between the cell and the base station. The results of this dependence are displayed in Fig. 3. We see that the energy metrics have been scaled up when applying the time-varying mode. At 1500 seconds, it has the highest energy cost due to the scarcity of intermediate nodes in the network. It should be noted that although the energy metric has increased, the shape of energy potential over space still resembles a cup, albeit with some distortion. This is a property that we shall exploit in Section V.

This energy potential has been obtained by averaging global end-to-end metrics of energy consumption, and for real networks, we propose a method for how to estimate this potential using only local measurements (of the “field” associated with this potential). The link energy costs can be obtained by measuring the transmit energy (per bit) to transmit useful data to one’s neighbor. The choice of a neighbor must be done in a completely localized fashion, since the network is assumed to be highly mobile. We base our decisions on the minimum Joules per bit-meter that a node can achieve at every hop instead of using global information passing in a network. That is, the next-hop neighbor in a mobile network is chosen as the node that minimizes the Joules per bit-meter towards the destination. Note that the “meter” in this metric is the number of useful meters towards the destination, not the number of meters in a hop (hence, one must project the distance onto the line towards the base station). For example, in Fig. 4, the “Joules per bit-meter” metric $M_1$ toward node 1 is as follows: $M_1 = \frac{\|d_1\|^2}{\|q\| \|d_1\|}$, where $d_1$ denotes the projected vector. ($q = 4$, in the figure.) We say that a point $(x, y)$ in space is dominated by $(x', y')$ if the Joules per bit-meter metric to transmit to $(x, y)$ is more than that to $(x', y')$. Hence, we can exclude the dominated region (the shaded region in the figure) from consideration when searching for optimal neighbors at each hop. This minimization of Joules per bit-meter at every hop can be shown to be a suboptimal scheme, and counterexamples are easy to construct. However, in randomly deployed mobile networks with a single base station as the destination, most of the true minimum energy routes to the destination are indeed along these greedy Joules per bit-meter neighbors at each step.

The link energy costs are measured using the minimum Joules per bit-meter neighbor and serve as the spatial derivative of the energy potential function. Hence, this method does not rely on any cost distribution from any physical location. Any node that wishes to get the energy metric required to reach the base station, will be able to compute the value of the energy potential for its current location, on demand, by using the local measurements made available to it via exchanges with other nodes (which we shall describe in Section IV), and using a “path integration method”, which we develop next.
REPORT:
Report the raw measurements with
generating node ID and time-tag
\( t_0 \), \( t_1 \), \( t_2 \), \( t_3 \), \( t_4 \)

RECEPTION:
Receive the raw measurements
reported by its neighbors
\( t_0 \), \( t_1 \), \( t_2 \), \( t_3 \), \( t_4 \)

FILTERING:
Check the generating node ID and time-tag of the
measurements and remove the replica
From node \( j \), \( \{ m(1, t_0), m(4, t_1), \ldots, m(9, t_3) \} \)
From node \( k \), \( \{ m(3, t_0), m(4, t_1), \ldots, m(9, t_3) \} \)

UPDATE NEXT REPORT:
Generate next reporting data by
shifting the timing window and
including new link cost measured at \( t_1 \)
\( t_0 \), \( t_1 \), \( t_2 \)

UPDATE FIELD INFORMATION:
Another filtering out the replica
already used for the previous update
up to \( t_1 \), and update the field
information at \( t_0 \)

Fig. 5. Algorithm for energy metric dissemination and filtering.

In the path integration method, if the link cost measurement was based on a transmission from \((x, y)\) to some point \((x', y')\), we record this as a “transmission vector”. Assume, for now, that the data set of these local link costs measurements and their transmission vectors are globally available to us. Starting at any point \((x, y)\) in space, we take the vector average over the transmission vectors which are recorded for the cell in which \((x, y)\) falls. Here, we limit the number of transmission vector measurements to be averaged (which is parameterized by a window size of \(S\)). The reason is that if we do not limit the number of measurements \((S)\) to be averaged, measurements might become obsolete in a fast varying network, and we avoid the use of these obsolete data which would reduce the accuracy of an energy map. We also average separately the energy consumption of the measured link energy costs recorded for the transmission vectors from that cell, and associate this average energy with that useful displacement. After we take the useful average displacement using the average transmission vector, we apply the same procedure to the new point at which we have arrived. Hence, this procedure is repeated for subsequent points and a “projected” path is virtually constructed from the point \((x, y)\) to the base station. The sum of the averaged per-hop energy metrics on this path from the point \((x, y)\) to the base station, constitutes the end-to-end energy potential at the point \((x, y)\).

IV. COLLECTING AND DISTRIBUTING THE SPATIAL ENERGY MEASUREMENTS

In this section, we show how the nodes can exchange the local link cost measurements so that they can form the energy maps.

In our design, the nodes measure the magnitude and direction of their transmission vectors and the energy per bit that it takes to transmit along a link. Each such measurement is stamped with the position, the time (the absolute time is provided by the same GPS receiver), and node ID. It appears expensive to share raw data of the measurements. However, as the new measurements spread throughout the network, it soon becomes no longer necessary to exchange those measurements since almost every node will have received them. Hence, we define a parameter \(W\) as a window size, to limit the number of measurements by the generation times of data. The nodes report only those measurements that were made with generation stamps that are within the window of \(W\) discrete-time steps of the current time. This \(W\) is different from \(S\) because \(W\) is not related to the coherence time of the energy potential, but rather to the speed of data dissemination.

Fig. 5 displays the algorithm that every node uses. Every node makes one measurement at every discrete-time increment and keeps a table of the measurements it has heard or generated with the generation times that are at most \(W\) discrete-time increments into the past. At each sampling interval (5 seconds in our simulations), nodes exchange only this data with their neighbors. The neighbors are selected via the Joule per bit-meter metrics. The received data is checked against the data reported, and any received data that is a replica of what has been reported is eliminated in the “Filtering” block shown in the figure. The figure shows an example where two blocks of measurement data have been received, one from neighbor node \(j\), and another from neighbor node \(k\). In the middle, there is a replica where both the node ID (4 in the figure) and the generation times (\(t_{-3}\) in the figure) match; hence, this replica is eliminated. After filtering, the remaining data is used in two ways: first, these measurements are used to update the average link cost vector assigned to each cell (shown in the lower right hand corner of Fig. 5). Second, the next report is created by incorporating the new data into the current table, and also shifting the window of size \(W\) by one interval.

The performance of this energy metric dissemination algorithm can be measured by the percentage difference between the energy potential that is obtained via data dissemination (and path integral construction) and the actual end-to-end measurement that is obtained via Dijkstra’s algorithm. Fig. 6 shows this percentage error for the simulations using the RWP model only. The graph reports the percentage error averaged over all the nodes in the network; further, it is parameterized by the window size \(W\). Whenever a node cannot construct a path, an error of 100% is assigned to the error percentage in that occurrence. We see that a window size of \(W = 5\) produces an error percentage of roughly 8—9% in the limit. This error has two components. First, an irreducible error floor component...
comes from the fact that data dissemination methods have used the path integration method that causes an average of 5% error on the deployment region. The difference of 3 – 4% arises due to the effects of a finite $W$ (where $W = 5$ in our discussion). Because $W$ places limits on what can be disseminated before it is considered obsolete, it is as expected that this causes errors when compared to a scheme that has global access to all of the link cost measurements when computing the average link cost vector for each cell. However, the difference is small.

Based on Fig. 6, we define the “spreading period” as the time that it takes to reach the point where a node can construct the energy potential with 80% accuracy, that is, with a target of 20% percentage error. (This definition can be modified to suit the application, by varying the target percentage of error.) We see that $W = 0$ leads to a spreading period that is roughly 1400 seconds, and it is down to 65 seconds for $W = 5$. (This has been calculated based on the exact data set.) Note that it takes roughly 50 seconds (in our simulation set-up for the RWP-only mode) for a node to reach from a starting point to an end-point (where it picks a new destination to travel to). Hence, for $W = 5$, after a typical node has travelled roughly 1.3 such lengths, we arrive at a point where a node can, on average, construct the end-to-end energies with 80% of accuracy, for all the positions it needs.

Now, we show the results for the simulations that includes the time-varying mode in Fig. 7 for $W = 5$. As a performance metric, we now compute an instantaneous percentage error of the constructed energy values with respect to the instantaneous, global energy measurements (not the averaged, global energy measurements over the entire time horizon, which we used in the previous evaluation). The conclusion of this graph is two-fold: first, a path from any point over the deployment region to the base station can be constructed with a percentage error of less than 17% in the time-invariant mode, and of between 20 – 27% in the time-varying mode. The reason of this worse performance for the time-varying mode is that the network is extremely fast changing and its performance is severely affected by the scarcity of up-to-date energy measurements. Despite this performance degradation, overall, a node is able to keep track of the variance at the cost of percentage errors roughly of only 3 – 10% more. Furthermore, note that if we can detect which mode of the network is currently in, such a large percentage error for the time-invariant mode can be reduced by expanding the window $S$ (namely, by using more relevant measurements) to obtain the reliable average energy metric.

Second, we explain the effect of the window size $S$ on the average energy potential. From Fig. 7, we see that the case of $S = 50$, which is the largest $S$ in our simulations, performs better for the time-invariant mode, while for the time-varying mode the case of $S = 1$ does better. This is because the coherence time of the energy potential in time-invariant mode is infinite (i.e. the optimal $S = \infty$), while the coherence time of the every potential in the time-varying mode in this simulation is only 5 seconds (equal to the sampling interval), because the mobility models we used induces an extremely fast varying energy potential. The implication of this result for the actual system design is that the coherence time $S$ could be obtained by measurements from the network conditions. One might perform a field study over the deployment regions such as campuses, malls and tourist spots. In the study, the collected energy maps will play the role of the indicators of the energy potential characteristics. Then, a look-up table of $S$ corresponding to the different network conditions can be built, and each node can use this table for its path integration to construct the energy maps.

The average amount of measurement data broadcast by a node indicates the amount of bandwidth consumed by measurement exchanges. We counted the average number of measurements broadcast per node at each sampling interval, as a function of continuous time. We found that even though there is an initial sharp rise is in the beginning, the average number of measurements broadcast per node per sampling interval settles down to a steady-steady value. The steady-state values allow us to quantify the overhead involved in disseminating QoS information. For example, when $W = 5$, we found that the average number of measurements that a node broadcasts in steady-state is 75 per broadcast. Since a node ID, the position and the time are also transmitted with each measurement, the following calculation gives an idea of the overhead involved: there are 7 bits required to address 100 nodes uniquely, and 9 bits required for the 400 cells. Using a round-robin sequencing method for the times, and assuming that we can wrap around after 4 spreading periods, 7 bits are required to stamp the times for $W = 5$. Each measurement reports a transmission vector, which is a displacement from the center of the cell to an angular quantization of 10 degrees, leading to 18 such useful displacements in the relevant half-plane, hence representable by 5 bits. Finally, the measured value of the link cost can be quantized to 8 bits, representing 256 such values. Therefore, each measurement entails a total of 36 bits; hence, 75 measurements per discrete-time unit entails 338 bytes of information. For $W = 3$, we found that this is reduced to 112 bytes (25 measurements), and the percentage error performance from Fig. 6 is still within reasonable range. If the link speed is 100 kbps (whenever a transmission occurs), 901 bits for the $W = 3$ design would be transmitted in roughly 0.01 seconds, and this would not consume a large fraction of the overall bandwidth. However, the Doppler shift due to mobility will also place bounds on the link speeds that can be achieved. A real network design must carefully examine these trade-offs.
VI. CONCLUSION

We developed the spatial energy maps framework aimed at enabling energy planning in high-mobility networks, where the networks have time-varying energy potentials as well as time-invariant energy potentials. Overall, our energy planning method is based on the constructed energy maps via the path integration and the dissemination methods has a percentage error of less than 15% compared with the same planning algorithm but using the optimal, globally available energy metrics. There are some aspects of this framework for further study: first, our results were confined to only energy potentials. The extension to other QoS metrics such as end-to-end delay and bit error rate (BER) may be considered as well. The dissemination method developed should be applicable to other QoS metrics, but their construction and integration methods will be different, depending on the metric. Second, overhead reduction methods must be pursued to reduce the amount of energy metric data broadcast in every iteration. We showed that when the link speeds are on the order of 100 kbps, this overhead can be handled by the methods in this work. But we expect drastic reductions to occur with a better understanding of these redundancies. Third, it would be useful to model a network where the underlying averages at each cell vary at a slow or a moderate rate over time. It might be possible to use the real network statistics obtained from some field studies. Using these real network statistics, we can see how our new design paradigm could work in future, large-scale wireless networks.

REFERENCES


Fig. 8. Performance comparison of energy planning with between the constructed energy maps and the actual energy maps.

V. ENERGY PLANNING UTILIZING AN ENERGY MAP

In this section, we allow a node to allocate its traffic to save its energy by utilizing the energy maps constructed.

As a problem description, in Fig. 1, a node moves around the deployment region, and it needs to send its traffic to the collection site (base station) with a time deadline of $4T_s$, where $T_s$ denotes the sampling interval. An energy optimization, which is based on Dynamic Programming (DP), dynamically allocates the optimal amount of bits at each stage. (Due to the limited space, we shall not explain how dynamic program works in detail but show only the results.) Assuming that the node can predict its trajectory in the near future, end-to-end energy metrics from the cells where the node will be located can be known via its current energy map of the network. By applying the same traffic allocation algorithm, we compare the total energy consumption of our energy maps and that of an optimal method in which the total energy consumption is computed by using Dijkstra’s algorithm with global information on the network.

We display the results in Fig. 8. Part (a) shows the normalized energy expenditures of the two methods, and Part (b) shows the percentage error of our energy planning over the optimal energy expenditure. The percentage error is roughly between $+15\%$ and $-15\%$. The positive errors come from the over-estimation of the energy value of our path integration, while the negative ones come from the under-estimations. This error is caused by all the errors induced by the path integration and the local dissemination. Even under the fast varying mode, our energy planning based on the energy maps can perform with such accuracies. This is due to the property that the ordering of energy potentials over the geometric energy potential surface has remained roughly the same (as shown in Fig. 2 and Fig. 3 and noted in Section IV). The implication of this result for an actual system design is that we can use an energy map to do the planning of data transmission with an accuracy of more than $85\%$ without worrying about the time-variace of the network. Hence, the energy planning and optimization problem for high-mobility networks is solvable.