Optimal Tree Pruning for Location Update in Machine-to-Machine Communications

Rung-Hung Gau, *Member, IEEE*, and Ching-Pei Cheng, *Member, IEEE*

***Abstract*—In this paper, we propose a novel energy-and-memory efficient location update scheme for wireless M2M communications. According to information theory, in comparison with periodically registering to the Machine-Type Communications (MTC) server, it is more efficient for a MTC device to perform location updates only when new nodes are added into the corresponding parsing tree. Based on the theory of random walks over trees, we propose optimally pruning the parsing tree to minimize the expected value of the sum of the memory cost and the energy cost for a MTC device. We use both analytical results and simulation results to justify the usage of the proposed scheme.**

***Index Terms*—Machine-to-machine communications, mobility management, random walks over trees.**

I. Introduction

MACHINE-TO-MACHINE (M2M) communications represents a future where billions of everyday objects and the surrounding environment are connected and managed through a variety of devices, communications networks, and cloud-based servers [1]. Representative M2M usage models include (but are not limited to) smart electric grids, connected cars that react in real time to prevent accidents, and body area networks that track vital signs. M2M communications is a key enabling technology for Internet of Things [2] in which the large number involved leads to a number of research challenges. In 3GPP standards, M2M communications is also called Machine-Type Communications (MTC) [3] [4] [5].

In this paper, we propose an energy-and-memory efficient location update scheme for wireless M2M communications. In particular, we study the case in which the MTC server has to track the location of each MTC device but does not know the mobility patterns of MTC devices in advance. Since it is widely expected that a variety of wireless access networks will coexist in the near future, the proposed location update scheme is designed to be independent of the wireless access networks. In addition to the identity of the closest base station, a MTC device could inform the MTC server its geographical location if the MTC device is equipped with a GPS receiver. Since the total number of MTC devices is expected to exceed the human population, it is desired to optimize the location update cost for each MTC device.

Manuscript received November 22, 2011; revised July 6 and December 5, 2012; accepted March 5, 2013. The associate editor coordinating the review of this paper and approving it for publication was A. Abouzeid.

The authors are with the Institute of Communications Engineering, National Chiao Tung University, Hsinchu, Taiwan (e-mail: runghung@mail.nctu.edu.tw; setpeggy@gmail.com).

This research is supported in part by the National Science Council, Taiwan, R.O.C. under grant numbers NSC 100-2628-E-009-017-MY2 and NSC 101-2221-E-009-005-MY3.

Digital Object Identifier 10.1109/TWC.2013.040413.112086

Novel information-theoretic approaches for location update in wireless cellular networks can be found in [6] [7] [8]. In particular, instead of performing location updates periodically, information theory (source coding theory) [9] is used to decide the optimal time instances for a mobile device to perform location updates. The realizations of the information-theoretic approaches are based on parsing trees. Our major technical contributions include the following. First, based on the theory of random walks over trees [10], we propose a novel algorithm to optimally prune the parsing tree whenever appropriate. Note that the size of a parsing tree grows with time but the memory size of a MTC device is finite and may not be as large as a smart phone. The proposed algorithm is designed to minimize the weighted sum of the memory cost and the energy cost. In addition, we formally prove that when the weight for the energy cost is smaller than the weight for the memory cost, whenever the parsing tree has to be pruned, it is optimal to remove all nodes except the root node. In contrast, when the ratio of the weight for the energy cost to the weight for the memory cost is large enough, it might not be optimal to remove all nodes except the root node. We use both analytical results and simulation results to justify the usage of the proposed approach.

The rest of the paper is organized as follows. Related works are covered in Section II. In Section III, we present the system models and briefly introduce the parsing tree for the benefits of the readers. In Section IV, we propose a novel approach to optimally prune the parsing tree based on the theory of random walks over trees. In Section V, we prove related mathematical results. In Section VI, we analyze the computational complexity of an algorithm that is used to estimate the location transition probabilities. In Section VII, we analytically derive the worst-case computational complexity for the proposed location update scheme. In Section VIII, we show simulation results that reveal the advantages of using the proposed scheme. Our conclusions are included in Section IX. Related mathematical proofs are included in the Appendix.

II. Related Works

Wireless M2M communications is a young research field. Lopez, Moura, Moreno, and Almeida [11] proposed a M2M-based platform that enables saving energy by remotely monitoring, controlling, and coordinating power generation and consumption in electric grids. Chen and Wang [12] considered optimizations for M2M communications in LTE-A systems. Lien and Chen [13] proposed a massive access management scheme to provide guarantees for MTC devices. In [14],

1536-1276/13$31.00 © 2013 IEEE

a non-cooperative game-theoretic approach is proposed for distributed rate and admission control in home M2M networks.

In cellular networks, by sensing the signal strengths and listening to broadcast messages, a mobile device could know the identity of the closest base station. Location update and paging are key components for idle-mode mobility management. On the other hand, handover/handoff is essential for connected-mode mobility management. A lot of prior works on paging could be found in the reference of [15] [16]. We focus on location update for a large number of MTC devices in this paper. Location update schemes could be movement-based [17], timer-based [18], distance-based [19] [20], profile-based [21], state-based [22], or velocity-based [23]. It was shown in [24] [25] that distance-based schemes achieve better performance compared to movement-based schemes and timer-based schemes. Hybrid location update schemes can be found in [26] and reference therein. Liang and Haas [27] proposed a predictive distance-based user-tracking scheme based on the Markov-Gaussian random process. Cayirci and Akyildiz [28] proposed the user mobility pattern scheme for location update and paging. Wu, Mukherjee, and Bhargava [29] examined a location update/paging scheme in hierarchical cellular networks. Ng and Chan [30] proposed an enhanced distance-based location management scheme using cell coordinates.

Some researchers assumed that a mobile user sends a location update message to the system whenever it enters a new location area and concentrated on the design of an optimal location area [31] [32] [33] [34] [35]. Other researchers assumed that location areas are given and focused on the decision problem of whether a mobile user should send a location update message when it enters a new location area [36]. Some proposed location update schemes [37] [38] suggested that a mobile user should register its location only when it enters some predefined cells, referred to as reporting centers. In [39], overlapping location areas were proposed to reduce the location update cost. The problem of joint optimization of location update and paging has been investigated in [40] [41]. Jeon and Jeong [42] proposed changing the location update probability of a mobile user based on its call arrival rate and location area change rate. Mao and Douligeris [43] proposed buffering location update messages until a call arrives. In this paper, we explicitly take the memory constraints of MTC devices into consideration. Instead of random walks over a graph with cycles, our work is based on the theory of random walks over trees. To the best of our knowledge, the latter is not used in the previous works on location update.

III. System Models and The Parsing Tree

A. System Models

In Figure 1, we show our network model. In particular, there is a MTC server and M MTC devices, which are indexed by $1, 2, 3, .., M$. The MTC server is connected to the Internet. On the other hand, a MTC device connects to the Internet through 3G/LTE wireless networks or WiFi access points. The service region of M2M communications is partitioned into $\alpha \geq 2$ M2M location areas, indexed by $1, 2, 3, .., \alpha$. When each MTC device has a GPS receiver, a M2M location area corresponds to a square with width equal to L meters on

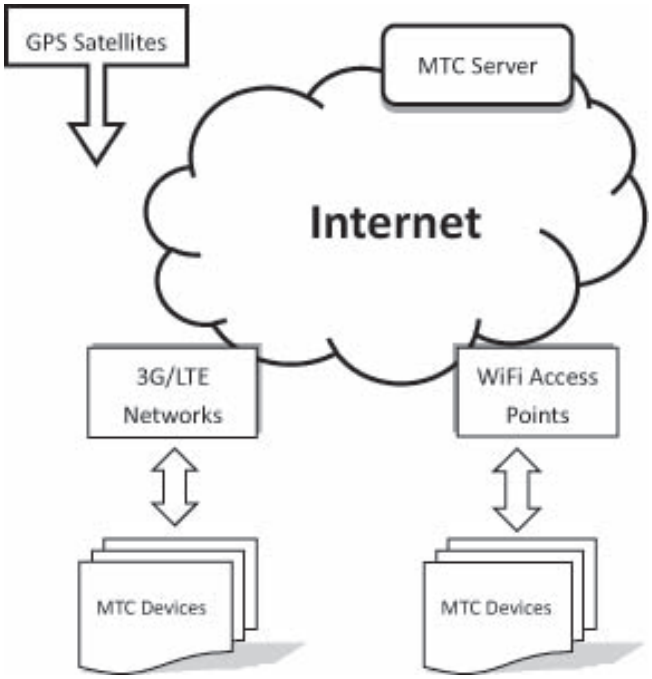


Fig. 1. The system model for wireless M2M communications.

Earth's surface. Otherwise, a M2M location area corresponds to the area covered by a 3G/LTE base station or a WiFi access point. To simplify the descriptions, in this paper, it is assumed that a MTC device is equipped with a GPS receiver. With minor revisions, the proposed algorithm could be used when a MTC device is not equipped with GPS receivers. It is assumed that whenever a MTC device performs sensing, the MTC device knows the index of the M2M location area in which it currently resides.

For modeling the mobility of MTC devices, we first introduce a very general continuous-time model. Next, we create a discrete-time model that is based on the continuous-time model and is sufficient for studying location updates. Let $(x_m(t), y_m(t))$ be the position of the mth MTC device at time t, where t is a non-negative real number. It is assumed that for each integer m, where $1 \leq m \leq M$, $x_m(t)$ and $y_m(t)$ are both continuous and differentiable functions. Let S_{max} be a positive real number. The maximum speed of a MTC device is S_{max} meters per second. Recall that location update is a mechanism for idle-mode mobility management. In particular, to reduce energy consumption, a mobile device only performs location update occasionally. Thus, we propose the following discrete-time mobility model. Let T be a positive real number. Time is partition into time slots of equal length. The length of a time slot equals T seconds. The time interval $[(k-1)T, kT]$ is called the kth time slot, $\forall k \geq 1$. Based on the above continuous-time mobility model, for the ith MTC device in time slot $k+1$, where $1 \leq i \leq M$ and $k \geq 1$, we have $-S_{max}T \leq x_i((k+1)T) - x_i(kT) \leq S_{max}T$ and $-S_{max}T \leq y_i((k+1)T) - y_i(kT) \leq S_{max}T$. Let $X_k^{(m)}$ be a random variable that represents the index of the M2M location area in which the mth MTC device resides at time kT, $\forall k \geq 0$. In order to obtain the value of $X_k^{(m)}$, at time $k \times T$, the mth MTC device performs sensing. A

graph $G = (V, E)$, where V is the vertex set and E is the edge set [44], is used to represent the relations between $X_{k+1}^{(m)}$ and $X_k^{(m)}$. In particular, vertex i corresponds to the ith M2M location area, $\forall i$. Note that (i, j) is the edge between vertex i and vertex j in the graph G. In addition, $(i, j) \in E$ if and only if there exists a pair of integers (m, k) such that $P\{X_{k+1}^{(m)} = j | X_k^{(m)} = i\} > 0$. Namely, $(i, j) \in E$ if and only if there exists a MTC device that might move from the ith M2M location area to the jth M2M location area in one time slot. For each $i \in V$, define $N_1(i) = \{j | (i, j) \in E\}$. Denote the cardinality of a set S by $|S|$. Since the maximum distance of movement in a time slot for a MTC device is $S_{max}T$ and the width of a M2M location area is L, $|N_1(i)| \leq (2\lceil \frac{S_{max}T}{L} \rceil + 1)^2$. In addition, $|N_1(i)| \leq |V| = \alpha$. Thus, $|N_1(i)| \leq \min\{(2\lceil \frac{S_{max}T}{L} \rceil + 1)^2, \alpha\}$. Namely, given that a MTC device is in the ith M2M location area at time kT, we can conclude that at time $(k + 1)T$ the MTC device has to reside in a M2M location area indexed by j, where $j \in N_1(i)$. While a MTC device might continuously move, for studying location updates, it is sufficient to use the above discrete-time model. Define $\Delta = \min\{(2\lceil \frac{S_{max}T}{L} \rceil + 1)^2, \alpha\}$. Define $d(i, j)$ as the distance between vertex i and vertex j in the graph G. In particular, $d(i, j)$ is the total number of edges that belong to the shortest path between vertex i and vertex j in the graph G.

It is assumed that for each fixed m, the discrete-time stochastic process $\{X_k^{(m)}\}_{k=0}^{\infty}$ is a (time-homogeneous) discrete-time Markov chain (DTMC) [10] with state space $\{1, 2, 3, .., \alpha\}$. To the best of our knowledge, DTMC is widely used for modeling mobility. Note that if $(i, j) \notin E$, $P\{X_{k+1}^{(m)} = j | X_k^{(m)} = i\} = 0$, $\forall m, k$. A MTC device is either static or mobile. If the mth MTC device is static, $P\{X_{k+1}^{(m)} = i | X_k^{(m)} = i\} = 1$ and $X_k^{(m)} = X_0^{(m)}$, $\forall k, i$. On the other hand, if the mth MTC device is mobile, $P\{X_{k+1}^{(m)} = i | X_k^{(m)} = i\} < 1$, $\forall i$. We focus on the case in which $\lim_{k\to\infty} P\{X_k^{(m)} = j\}$ exists, $\forall m, j$. Define $v_j^{(m)} = \lim_{k\to\infty} P\{X_k^{(m)} = j\}$, $\forall m, j$. The above DTMC model is very general and covers a number of mobility patterns. For example, when the movements of the mth MTC device form a random walk or a random walk with drift, $\{X_k^{(m)}\}_{k=0}^{\infty}$ is a DTMC.

All MTC devices adopt the same location update algorithm and each MTC device runs the location update algorithm independently. Therefore, it is sufficient to consider a MTC device. In the rest of the paper, when it is clear from the context, we abbreviate $X_k^{(m)}$, $P\{X_{k+1}^{(m)} = j | X_k^{(m)} = i\}$, and $v_j^{(m)}$ by X_k, $P\{X_{k+1} = j | X_k = i\}$, and v_j, respectively. Define $p_{i,j} = P\{X_{k+1} = j | X_k = i\}$, $\forall i, j$.

B. Parsing Trees for Information-Theoretic Location Update

A string is composed of a number of symbols. The maximal proper prefix of a string is obtained by removing the last symbol from the string. Among the prefixes of a string, we focus only on the maximal proper prefix of the string. In this paper, prefix is simply an abbreviation for maximum proper prefix. Note that for a string containing a unique symbol, the prefix is $\emptyset$. Denote the concatenation of two strings Y_1 and Y_2 by $Y_1 \oplus Y_2$, which is abbreviated by Y_1Y_2. For example, when $Y_1 = 12$ and $Y_2 = 345$, $Y_1Y_2 = Y_1 \oplus Y_2 = 12345$. Define $s_0 = \emptyset$. For each string s, there exists an integer $k \geq 1$, k substrings $s_1, s_2, .., s_k$, and a function f such that $s = s_1 \oplus s_2 \oplus .. \oplus s_k$, $f(i) \in \{0, 1, 2, .., i-1\}$, $\forall i \in \{1, 2, 3, .., k\}$, and the prefix of s_i equals $s_{f(i)}$, $\forall i \in \{1, 2, 3, .., k\}$ [46] [6]. Denote the last symbol of s_i by w_i. Denote the prefix of s_i by v_i. Note that $s_{f(i)} = v_i$ and $s_i = v_i \oplus w_i$. Given a string s, we can use a parsing tree to find the value of k, the function f, and the substrings $s_1, s_2, .., s_k$.

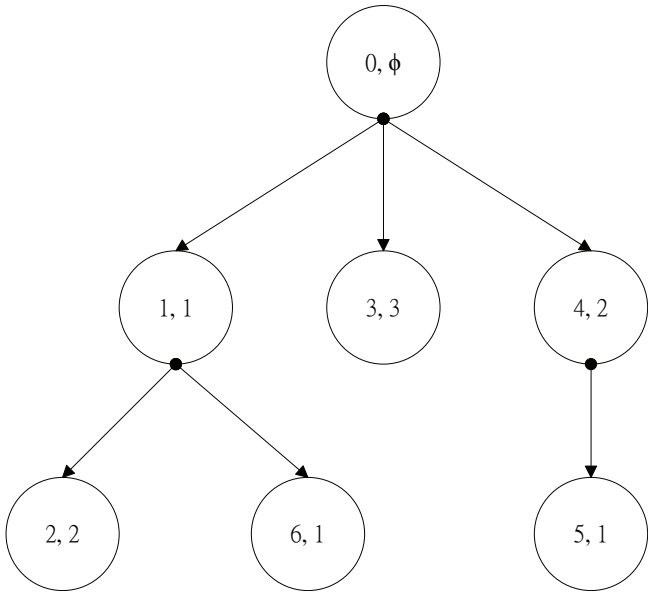


Fig. 2. The parsing tree Γ at time t_9.

The MTC device maintains a parsing tree and a queue, which is initially empty. A parsing tree is a well-known data structure in the field of source coding [45] [6]. Let γ be the maximum number of nodes in a parsing tree due to memory constraints. Each node in the parsing tree has an index and a label. In particular, the index of a node is defined to be the order that the node is added into the parsing tree. Note that node i is the node with index equal to i. Let L_i be the label of node i. In particular, L_i is the index of the M2M location area in which the MTC device resides when node i is added into the parsing tree. In addition, each node in a parsing tree contains an array of α pointers [45]. Each pointer either points to a child of the node or is a null pointer. Let Φ_i be the set composed of the indexes of the children of node i in the parsing tree.

Initially, the parsing tree is composed of the root node with index zero and label $\emptyset$. Let $N(t)$ be the total number of nodes in the parsing tree at time t. Let $N(t^-)$ be the total number of nodes in the parsing tree just before time t. If a new node is added into the parsing tree at time t, the index of the node is set to be $N(t^-)$. A MTC device maintains two pointers, which are called root and current state, respectively. The root pointer always points to the root node of the parsing tree. At time zero, the current state pointer also points to the root node of the parsing tree. Define $t_k = kT$, $\forall k \geq 1$. Let S_k be the index of the node that is pointed by the current state pointer just before time t_k, $\forall k \geq 1$. Note that $S_1 = 0$. Define

TABLE I
THE DICTIONARY FOR ENCODING LOCATION UPDATE MESSAGES

i	record creation time	the ith substring, s_i	$v_i, w_i, f(i)$ ($s_i = v_i \oplus w_i$, $s_{f(i)} = v_i$)	integer to be encoded for location update, $m_i = f(i)\alpha + w_i - 1$	number of sent bits for location update, $\lceil \log_2(\alpha i) \rceil$	bits sent for location update
0	0	$\emptyset$				
1	t_1	$1(X_1)$	$\emptyset$,1,0	$0 \times 4 + (1-1) = 0$	2	00
2	t_3	$12(X_2 \oplus X_3)$	1,2,1	$1 \times 4 + (2-1) = 5$	3	101
3	t_4	$3(X_4)$	$\emptyset$,3,0	$0 \times 4 + (3-1) = 2$	4	0010
4	t_5	$2(X_5)$	$\emptyset$,2,0	$0 \times 4 + (2-1) = 1$	4	0001
5	t_7	$21(X_6 \oplus X_7)$	2,1,4	$4 \times 4 + (1-1) = 16$	5	10000
6	t_9	$11(X_8 \oplus X_9)$	1,1,1	$1 \times 4 + (1-1) = 4$	5	00100

TABLE II
THE DICTIONARY FOR DECODING LOCATION UPDATE MESSAGES

i	record creation time	expected number of bits for location update, $\lceil \log_2(\alpha i) \rceil$	received bits for location update	received integer for location update, m_i	q_i, the quotient of dividing m_i by α	r_i, the remainder of dividing m_i by α	s_i, the ith substring $s_i = s_{q_i} \oplus (r_i + 1)$
0	0						
1	t_1	2	00	0	0	0	$\emptyset \oplus (0+1) = 1$
2	t_3	3	101	5	1	1	$1 \oplus (1+1) = 12$
3	t_4	4	0010	2	0	2	$\emptyset \oplus (2+1) = 3$
4	t_5	4	0001	1	0	1	$\emptyset \oplus (1+1) = 2$
5	t_7	5	10000	16	4	0	$2 \oplus (0+1) = 21$
6	t_9	5	00100	4	1	0	$1 \oplus (0+1) = 11$

$\Theta_k = \{L_j : j \in \Phi(S_k)\}$. Namely, the set Θ_k is composed of the labels of the children of node S_k in the parsing tree. At time t_k, the value of X_k is inserted into the queue. If $X_k \in \Theta_k$, the MTC device does not perform location updates. In this case, S_{k+1} is set to be the integer j such that $j \in \Phi(S_k)$ and $L_j = X_k$. On the other hand, if $X_k \notin \Theta_k$, a new node with index $N(t_k^-)$ and label X_k is added into the parsing tree as a child of node S_k and S_{k+1} is set to be 0. In addition, based on the LZ78 data compression algorithm [46], the MTC device encodes the messages in the queue (into bits) and then clears the queue. In particular, when the kth node is added into the parsing tree, $\lceil \log_2(\alpha \cdot k) \rceil$ bits are used to encode all messages in the buffer. In this case, the MTC device performs a location update and sends the bits to the MTC server.

We use the following example to illustrate the above notations. Suppose $\alpha = 4$, $X_1 = 1$, $X_2 = 1$, $X_3 = 2$, $X_4 = 3$, $X_5 = 2$, $X_6 = 2$, $X_7 = 1$, $X_8 = 1$, and $X_9 = 1$. At time t_1, since $S_1 = 0$, $\Phi(S_1) = \emptyset$, $\Theta_1 = \emptyset$, and $X_1 \notin \Theta_1$, node $N(t_1^-) = 1$ is added into the parsing tree as a child of node 0 and S_2 is set to be 0. In addition, according to the LZ78 data compression algorithm, X_1 is encoded into $\log_2(\alpha \cdot 1) = 2$ bits 00, since $X_1 - 1 = 0$. Furthermore, the MTC device performs a location update and sends the two bits 00 to the MTC server. When the MTC server receives the first location update message from the MTC device, based on the LZ78 data compression algorithm, it decodes the two bits 00 and finds that $X_1 = 1$. At time t_2, since $S_2 = 0$, $\Phi(S_2) = \{1\}$, $\Theta_2 = \{1\}$, and $X_2 \in \Theta_2$, no nodes are added into the parsing tree and the MTC device does not perform location updates. Since $L_1 = X_2$, S_3 is set to be 1. At time t_3, since $S_3 = 1$, $\Phi(S_3) = \emptyset$, $\Theta_3 = \emptyset$, and $X_3 \notin \Theta_3$, node $N(t_3^-) = 2$ is added into the parsing tree as a child of node 1 and S_4 is set to be 0. In addition, according to the LZ78 data compression algorithm, the string $X_2 \oplus X_3 = 1 \oplus 2 = 12$ is encoded into $\log_2(\alpha \cdot 2) = 3$ bits 101. Furthermore, the MTC device performs a location update and sends the three bits 101 to the MTC server. When the MTC server receives the second location update message from the MTC device, it decodes the three bits 101 and concludes that $(X_2, X_3) = (1, 2)$. In Figure 2, we show the corresponding parsing tree at time t_9. For a node in Figure 2, the first symbol is the index of the node, while the second symbol is the label of the node. In

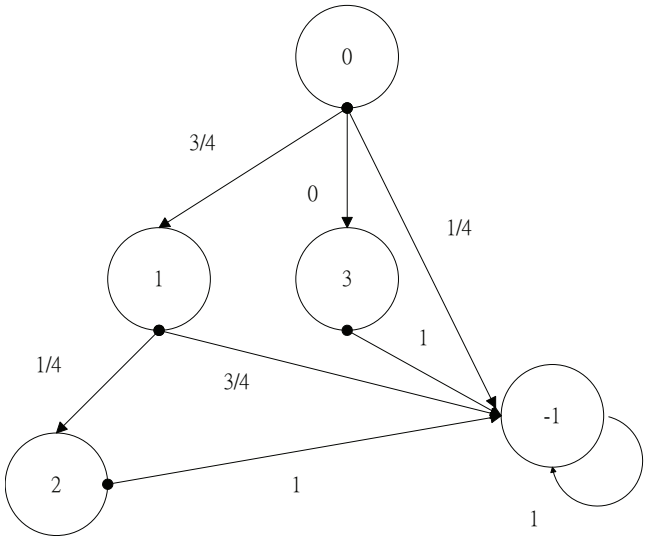


Fig. 3. The state transition diagram for the DTMC associated with Γ_{-3} at time t_9.

Table I, we show the dictionary used by the MTC device for compressing location update messages into bits at time t_9, when the LZ78 compression algorithm is used. In Table II, we show the corresponding dictionary used by the MTC server for decompressing received bits at time t_9, when the LZ78 compression algorithm is used.

Note that a MTC device performs location updates only when new nodes are added into the associated parsing tree. Therefore, the location update cost is reduced. In addition, since the above approach of location update corresponds to a lossless data compression scheme [46], the MTC server knows all values of X_k's [6].

IV. Optimally Pruning the Parsing Tree

In this section, based on the theory of random walks over trees, we propose a novel approach to optimally prune the parsing tree. For each MTC device, the memory size is finite and therefore the parsing tree cannot keep growing forever. Whenever it is necessary to prune the parsing tree, it is desired to prune the parsing tree in an optimal way so that the cost is minimized. In addition, both the memory cost and the energy cost should be taken into consideration. Typically, memory cost is an increasing function of the total number of nodes in the parsing tree. On the other hand, the energy cost is a decreasing function of the expected value of the time difference between two consecutive location updates. We will explicitly define the cost function later in the section.

Suppose now we have to prune the parsing tree Γ. Namely, there are γ nodes in the parsing tree and we have to remove at least one node from the tree Γ. To maintain the structure of the parsing tree for decompression, if node i is removed and $i < j$, node j also has to be removed. Let Γ_{-k} be the tree obtained by removing each node with index greater than k from the tree Γ. Once a parsing tree is pruned, the current state pointer points to the root node. Let $\Phi_{-k}(i)$ be the set composed of the indexes of the children of node i in the tree Γ_{-k}.

Given that the parsing tree Γ is currently pruned to be Γ_{-k}, we can predict the next location update time based on the movement history of the MTC device. In particular, for the tree Γ_{-k}, we construct the associated discrete-time Markov chain $\{Y_n^{-k}\}_{n=0}^{\infty}$ with state space $\Omega_{-k} = \{-1, 0, 1, 2, .., k\}$ as follows. First, state i of the DTMC corresponds to node i in the tree Γ_{-k}, $\forall 0 \leq i \leq k$. In addition, state -1 is an absorption state. In particular, given that $Y_0^{-k} = 0$, $Y_m^{-k} = -1$ if and only if the MTC device performs a location update at time t_m. The state transition probabilities of the DTMC $\{Y_n^{-k}\}_{n=0}^{\infty}$ should depend on the mobility pattern of the MTC device. In order to set the state transition probabilities for the DTMC $\{Y_n^{-k}\}_{n=0}^{\infty}$, we first define a few variables. Let q_i be the estimated value of the probability that the MTC device will appear in the ith M2M location area. Let $q_{i,j}$ be the estimated value of the probability that the MTC device will be in the jth M2M location area at the next sensing time instance given that the MTC device is currently in the ith M2M location area. Later in the paper, we will present an algorithm to derive the values of q_i and $q_{i,j}$. It is desired to set $P\{Y_{n+1} = j | Y_n = i\} = P\{X_{n+1} = L_j | X_n = L_i\}$, $\forall 1 \leq i, j \leq k$. However, the value of $P\{X_{n+1} = L_j | X_n = L_i\}$ is not always known in advance. Therefore, whenever the value of $P\{X_{n+1} = L_j | X_n = L_i\}$ is required, we use the value of q_{L_i,L_j} instead. Let τ be the index of the M2M location area in which the MTC device currently resides. We propose setting the state transition probabilities for the DTMC $\{Y_n^{-k}\}_{n=0}^{\infty}$ as follows:

$$P\{Y_{n+1}^{-k} = j | Y_n^{-k} = i\}$$

$$= \begin{cases} q_{\tau,L_j}, \ if \ i = 0 \ and \ j \in \Phi_{-k}(i) \\ 1 - \sum_{j: j \in \Phi_{-k}(i)} q_{\tau,L_j}, \ if \ i = 0 \ and \ j = -1 \\ q_{L_i,L_j}, \ if \ 1 \leq i \leq k \ and \ j \in \Phi_{-k}(i) \\ 1 - \sum_{j: j \in \Phi_{-k}(i)} q_{L_i,L_j}, if \ 1 \leq i \leq k \ and \ j = -1 \\ 1, \ if \ i = j = -1 \\ 0, \ otherwise. \end{cases}$$

We now elaborate on the above equation. Suppose the current state pointer points to the node with index i. Denote the next sensing result by s. If $i = 0$, the MTC device resides in the τth M2M location area. In this case, if $s \in \{L_j | j \in \Phi_{-k}(i)\}$, the MTC device will not perform a location update when it will get the next sensing result. Instead, the current state pointer will point to the node with index equals j, where j is an integer such that $L_j = s$. Given that the MTC device is in the τth M2M location area, the probability that the next sensing result will be s equals $p_{\tau,s}$. Since the value of $p_{\tau,s}$ is not known in advance, if $i = 0$ and $j \in \Phi_{-k}(i)$, we set $P\{Y_{n+1}^{-k} = j | Y_n^{-k} = i\}$ to be q_{τ,L_j}. If $i = 0$ and $s \notin \{L_j | j \in \Phi_{-k}(i)\}$, the MTC device will perform a location update when it will get the next sensing result. Therefore, we set $P\{Y_{n+1}^{-k} = -1 | Y_n^{-k} = 0\} = 1 - \sum_{j: j \in \Phi_{-k}(i)} q_{\tau,L_j}$. If $i \geq 1$, the MTC device resides in the L_ith M2M location area. Similar to the case in which $i = 0$, if $i \geq 1$ and $j \in \Phi_{-k}(i)$, we set $P\{Y_{n+1}^{-k} = j | Y_n^{-k} = i\}$ to be q_{L_i,L_j}. In addition, we set $P\{Y_{n+1}^{-k} = -1 | Y_n^{-k} = i\}$ to be $1 - \sum_{j: j \in \Phi_{-k}(i)} q_{L_i,L_j}$. The state -1 is designed to be an absorption state. Therefore, $P\{Y_{n+1}^{-k} = -1 | Y_n^{-k} = -1\} = 1$.

We use the following example to illustrate the above notations. Suppose $\alpha = 4$, $X_1 = 1$, $X_2 = 1$, $X_3 = 2$, $X_4 = 3$, $X_5 = 2$, $X_6 = 2$, $X_7 = 1$,$X_8 = 1$, and $X_9 = 1$. Suppose we want to prune the tree at time t_9. Then, $\tau = X_9 = 1$. Based on the estimation algorithm in Section VI, at time t_9, we have $q_1 = \frac{5}{9}$, $q_2 = \frac{3}{9}$, $q_3 = \frac{1}{9}$, and $q_4 = \frac{0}{9}$. In addition, $q_{1,1} = \frac{3}{4}$, $q_{1,2} = \frac{1}{4}$, and $q_{2,1} = \frac{1}{3}$. In Figure 2, we show the parsing tree Γ at time t_9. In Figure 3, we show the state transition diagram for the DTMC corresponding to Γ_{-3} (at time t_9). Except for the state -1, a state of the DTMC $\{Y_n^{-3}\}_{n=0}^{\infty}$ in Figure 3 corresponds to the index of a node of the tree in Figure 2. For example, since the node with index 1 and label 1 is a child of node 0 and $q_{1,1} = \frac{3}{4}$, we have $P\{Y_{n+1}^{-3} = 1|Y_n^{-3} = 0\} = \frac{3}{4}$. In addition, $P\{Y_{n+1}^{-3} = -1|Y_n^{-3} = 0\} = 1 - q_{1,1} - q_{1,3} = \frac{1}{4}$. Similarly, since the node with index 2 and label 2 is a child of node 1 and $q_{1,2} = \frac{1}{4}$, we have $P\{Y_{n+1}^{-3} = 2|Y_n^{-3} = 1\} = \frac{1}{4}$. Furthermore, $P\{Y_{n+1}^{-3} = -1|Y_n^{-3} = 1\} = 1 - q_{1,2} = \frac{3}{4}$. Moreover, due to that node 3 is a leaf node in Γ_{-3}, $P\{Y_{n+1}^{-3} = -1|Y_n^{-3} = 3\} = 1 - 0 = 1$.

Let H_{-k} be a random variable that represents the time instance when the discrete-time Markov chain $\{Y_n^{-k}\}_{n=0}^{\infty}$ visits state -1 for the first time, given that $Y_0^{-k} = 0$. In particular,

$$H_{-k} = \min\{n : n \geq 1, Y_n^{-k} = -1|Y_0^{-k} = 0\}. \tag{1}$$

Note that the DTMC $\{Y_n^{-k}\}_{n=0}^{\infty}$ corresponds to a random walk over the tree Γ_{-k}. In addition, the random variable H_{-k} corresponds to the length of the time interval elapsed from now to the time instance when the MTC device performs the next location update, given that the current parsing tree is Γ_{-k} and the current state pointer points to the root node. The larger the value of H_{-k} is, the smaller the energy cost for location update is.

We now derive the value of $\mathbf{E}[H_{-k}]$. Consider a fixed value of k. To simplify the notations, we abbreviate Y_n^{-k} by Y_n. Define $z_i = \mathbf{E}[\min\{n : n \geq 1, Y_n = -1\}|Y_0 = i]$. Namely, z_i is the average number of steps it takes for the DTMC $\{Y_n\}_{n=0}^{\infty}$ to visit state -1 for the first time, given that $Y_0 = i$. In particular, $z_0 = \mathbf{E}[H_{-k}]$. Then, based on [10], $\forall i \in \{0, 1, 2, .., k\}$,

$$z_i \quad = \quad 1 + \sum_{j=0}^{k} P\{Y_{n+1} = j|Y_n = i\} \times z_j. \tag{2}$$

We now elaborate on the above equation. First, it takes at least one step for the DTMC $\{Y_n\}_{n=0}^{\infty}$ to move from state i to state -1, $\forall i \neq -1$. Given that $Y_0 = i$, with probability $P\{Y_{n+1} = j|Y_n = i\}$, $Y_1 = j$. For each $j \neq -1$, given that $Y_1 = j$, on average, it takes z_j additional steps for the DTMC $\{Y_n\}_{n=0}^{\infty}$ to move from state j to state -1. Therefore, we have the above equation.

Let $\mathbf{z} = (z_0, z_1, .., z_k)^T$ and $\mathbf{1}_{k+1} = (1, 1, .., 1)^T$ be two $(k+1)$-dimensional column vectors. Define a $(k+1) \times (k+1)$ matrix $\mathbf{U}$ such that $[\mathbf{U}]_{i+1,j+1} = P\{Y_{n+1} = j|Y_n = i\}$, $\forall 0 \leq i, j \leq k$. For the above example, which is associated with Figure 3, we have

$$\mathbf{U} = \begin{bmatrix} 0 & \frac{3}{4} & 0 & \frac{1}{4} \\ 0 & 0 & \frac{1}{4} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

Recall that for a parsing tree, the index of a node corresponds to the order that the node is inserted into the parsing tree. Thus, that node j is a child of node i implies $j > i$. Therefore, $\mathbf{U}$ is an upper triangular matrix and $[\mathbf{U}]_{i,i} = 0$, $\forall i$. Then, based on Equation (2), we have the following matrix equation.

$$\mathbf{z} = \mathbf{1}_{k+1} + \mathbf{U} \times \mathbf{z}. \tag{3}$$

Let $\mathbf{I}_{k+1}$ be the $(k+1)$th order identity matrix. Then,

$$(\mathbf{I}_{k+1} - \mathbf{U}) \times \mathbf{z} \quad = \quad \mathbf{1}_{k+1}. \tag{4}$$

Based on the Gauss elimination method, the above equation can be solved in $O(k^3)$ time. We now show that the above matrix equation can be solved in $O(k)$ time. Since $\mathbf{U}$ is an upper triangular matrix, $(\mathbf{I}_{k+1} - \mathbf{U})$ is also an upper-triangular matrix. In addition, $[\mathbf{I}_{k+1} - \mathbf{U}]_{i,i} = 1$, $\forall i$. Thus, the matrix $\mathbf{I}_{k+1} - \mathbf{U}$ is full-ranked. Since the matrix $(\mathbf{I}_{k+1} - \mathbf{U})$ is both upper-triangular and full-ranked, Equation (4) can be solved by backward substitution without explicitly calculating the inverse matrix $(\mathbf{I}_{k+1} - \mathbf{U})^{-1}$. Recall that when the graph $G = (V, E)$ represents a tree, $|E| = |V| - 1$ [44]. Since there are $k + 1$ nodes and k edges in the tree Γ_{-k}, there are k nonzero elements in the matrix $\mathbf{U}$ and $k + (k + 1) = 2k + 1$ non-zero elements in the matrix $\mathbf{I}_{k+1} - \mathbf{U}$. Therefore, Equation (4) can be solved by backward substitution in $O(k)$ time.

Let $c_1, c_2 > 0$ be two real numbers. Denote the optimal value of k by k^*. We propose using the following approach to find the value of k^*.

$$k^* \quad = \quad \arg \min_{k:k \in \{0,1,2,..,\gamma-2\}} c_1 \times (k + 1) - c_2 \times \mathbf{E}[H_{-k}]. \tag{5}$$

Note that $c_1 \times (k + 1)$ corresponds to the memory cost, since there are $(k + 1)$ nodes in the parsing tree Γ_{-k}. On the other hand, $-c_2 \times \mathbf{E}[H_{-k}]$ corresponds to the energy cost. The larger the average length of the interval between two time instances of performing location updates is, the smaller the energy cost is.

Once the value of k^* is obtained, the MTC device prunes the parsing tree Γ such that nodes with indexes greater than k^* are removed from the parsing tree. In addition, the current state pointer points to the root node of the parsing tree. Furthermore, the MTC device informs the MTC server the latest value of k^*.

The proposed tree pruning algorithm and the encoding algorithm are executed in each MTC device in a distributed manner, while the decoding algorithm is executed in the MTC server (or Gateway) in a centralized manner. For location update, there are no control message exchanges between MTC devices. When the total number of MTC devices is very large, a cluster of MTC servers could be used for load balancing.

V. Analytical Results

Consider a MTC device. Let V_n be a random variable that represents the nth time when the parsing tree has γ nodes, $\forall n \geq 1$. We first prove that the random variable V_1 is finite for sure (with probability one). From the time instance when the current state pointer points to the root node to the time instance when a new node is added into the parsing tree, each of the old nodes in the parsing tree is pointed by the current state pointer in at most one sensing period. Therefore, it takes at most k sensing periods for the total number of nodes in the parsing tree to increase from k to $k+1$. Hence, $V_1 \leq \sum_{k=1}^{\gamma-1} k = \frac{\gamma(\gamma-1)}{2}$ for sure. Similarly, $V_{n+1} - V_n \leq \frac{\gamma(\gamma-1)}{2}$ for sure, $\forall n \geq 1$.

Consider a parsing tree Γ that contains γ nodes. We now derive properties for the random variables H_{-k}'s. First, $H_{-0} = 1$ for sure. Since Γ_{-k} is a tree that contains no cycles, starting from state 0, the DTMC $\{Y_n^{-k}\}_{n=0}^{\infty}$ never visits state i twice before it visits the state -1, $\forall 1 \leq i \leq k$. Thus, the DTMC $\{Y_n^{-k}\}_{n=0}^{\infty}$ visits up to k states before it visits the state -1. Therefore, $H_{-k} \leq k+1$ for sure. Furthermore, it can be proved that $1 < \mathbf{E}[H_{-1}] < \mathbf{E}[H_{-2}] < .. < \mathbf{E}[H_{-(r-2)}]$.

Let Z_n be a random variable that represents the state of the parsing tree at time V_n, $\forall n \geq 1$. Let $W(t)$ be the total number of nodes in the parsing tree at time t. By definition of V_n, $W(V_n) = \gamma$, $\forall n \geq 1$. Let Ω_Z be the state space of the random variables Z_n's. Define $\zeta = |\Omega_Z|$. We now prove that $\zeta \leq [(\alpha+1) \times (\gamma!\alpha!)]^{\gamma} < \infty$. First, there are γ nodes in the parsing tree at time V_n, $\forall n \geq 1$. For a node in the parsing tree, the total number of children ranges from 0 to α. When a node has $m \geq 1$ children, the total number of combinations for the indexes of the m children is at most $\gamma \times (\gamma-1) \times .. \times (\gamma-m+1)$, which is no greater than $\gamma!$. When a node has $m = 0$ children, the total number of combinations for the indexes of the m children is one, which is also smaller than $\gamma!$. Similarly, when a node has $m \geq 1$ children, the total number of combinations for the labels of the m children is at most $\alpha \times (\alpha-1) \times .. \times (\alpha-m+1)$, which is no greater than $\alpha!$. When a node has $m = 0$ children, the total number of combinations for the labels of the m children is one, which is also smaller than $\alpha!$. Therefore, $\zeta \leq [(\alpha+1) \times (\gamma!\alpha!)]^{\gamma} < \infty$. Since $\zeta < \infty$, for each realization of the stochastic process $\{Z_k\}_{k=1}^{\zeta+1}$, denoted by $\{Z_k(\omega)\}_{k=1}^{\zeta+1}$, there exists two integers $G_1(\omega)$ and $G_2(\omega)$ such that $1 \leq G_1(\omega) < G_2(\omega) \leq \zeta+1$ and $Z_{G_1(\omega)} = Z_{G_2(\omega)}$.

If $\frac{c_2}{c_1}$ is small enough, whenever the parsing tree has to be pruned, it is optimal to remove all nodes except the root node.

Lemma 1: If $\frac{c_2}{c_1} < 1$, $k^* = 0$.

Proof: See Appendix.

In contrast, if $\frac{c_2}{c_1}$ is large enough, it might not be optimal to remove all nodes except the root node.

Lemma 2: If there exists an integer j, where $j \in \Phi_0$ and $j \neq \gamma - 1$, such that $c_2 \cdot q_{\tau,L_j} - c_1 \cdot j > 0$, $k^* \geq 1$.

Proof: See Appendix.

The following theorem provides a sufficient condition for the DTMC $\{Z_n\}_{n=1}^{\infty}$ to be irreducible, aperiodic, and positive-recurrent.

Theorem 1: If $q_i = v_i$, $\forall i$, $q_{i,j} = P\{X_{k+1} = j | X_k = i\}$, $\forall i, j$, $\frac{c_2}{c_1} < 1$, $\gamma > \alpha$, the DTMC $\{X_k\}_{k=1}^{\infty}$ is irreducible, and $P\{X_{k+1} = i | X_k = i\} > 0$, $\forall 1 \leq i \leq \alpha$, the DTMC $\{Z_n\}_{n=1}^{\infty}$ is irreducible, aperiodic, and positive-recurrent.

Proof: See Appendix.

If $q_i = v_i$, $\forall i$, $q_{i,j} = P\{X_{k+1} = j | X_k = i\}$, $\forall i, j$, $\frac{c_2}{c_1} < 1$, $\gamma > \alpha$, the DTMC $\{X_k\}_{k=1}^{\infty}$ is irreducible, and $P\{X_{k+1} = i | X_k = i\} > 0$, $\forall 1 \leq i \leq \alpha$, based on the above theorem, $\lim_{n\to\infty} P\{Z_n = \xi | Z_1 = \xi_1\}$ exists and is independent of ξ_1, $\forall \xi, \xi_1 \in \Omega_Z$. In this case, the DTMC $\{Z_n\}_{n=1}^{\infty}$ has a unique steady-state probability distribution.

VI. Estimating The Location Transition Probabilities

For the completeness of the paper, we now present an algorithm for estimating the location transition probabilities of a MTC device. The algorithm is quite simple and its variants have been widely used. Let $q_{m,i}$ be the estimated value of the probability that the MTC device will appear in the ith M2M location area at the next sensing time based on the values of $X_1, X_2, .., X_m$. Let $q_{m,i,j}$ be the estimated value of the probability that the MTC device will appear in the jth M2M location area at the next sensing time given that the MTC device is currently in the ith M2M location area, based on the values of $X_1, X_2, .., X_m$. Let $I(condition)$ be the indicator function with value one if the condition is true or with value zero otherwise. To reduce the memory requirement, during the time interval $[t_m, t_{m+1})$, the MTC device only keeps the values of X_m, $q_{m-1,i}$'s, $q_{m,i}$'s, and $q_{m,i,j}$'s. Thus, the space requirement of the estimation algorithm is $O(1+\alpha+\alpha+\alpha^2) = O(\alpha^2)$. At time t_{m+1}, the MTC device obtains the value of X_{m+1} and calculates the values of $q_{m+1,i}$'s according to the following equation.

$$\begin{aligned} q_{m+1,i} &= \frac{\sum_{k=1}^{m+1} I(X_k = i)}{m+1} \\ &= \frac{\sum_{k=1}^{m} I(X_k = i) + I(X_{m+1} = i)}{m+1} \\ &= \frac{m \cdot q_{m,i}}{m+1} + \frac{I(X_{m+1} = i)}{m+1}, \forall i. \end{aligned} \tag{6}$$

In addition, at time t_{m+1}, the MTC device calculates the values of $q_{m+1,i,j}$'s as follows. First, if $\sum_{k=1}^{m} I(X_k = i) = 0$, $q_{m+1,i,j} = 0$, $\forall j$. On the other hand, if $\sum_{k=1}^{m} I(X_k = i) \geq 1$,

$$\begin{aligned} q_{m+1,i,j} &= \frac{\sum_{k=1}^{m} I(X_k = i, X_{k+1} = j)}{\sum_{k=1}^{m} I(X_k = i)} \\ &= \frac{1}{\sum_{k=1}^{m} I(X_k = i)} \cdot [\sum_{k=1}^{m-1} I(X_k = i, X_{k+1} = j) + \\ &\quad I(X_m = i, X_{m+1} = j)] \\ &= \frac{1}{m \cdot q_{m,i}} \cdot [(m-1) \cdot q_{m-1,i} \cdot q_{m,i,j} + \\ &\quad I(X_m = i, X_{m+1} = j)], \forall j. \end{aligned} \tag{7}$$

The computational complexity of Equation (6) for calculating the values of $q_{m,i}$'s is $O(\alpha)$. In addition, the computational complexity of Equation (7) for calculating the values of $q_{m,i,j}$'s is $O(\alpha^2)$. Therefore, the computational complexity of the estimation algorithm is $O(\alpha + \alpha^2) = O(\alpha^2)$.

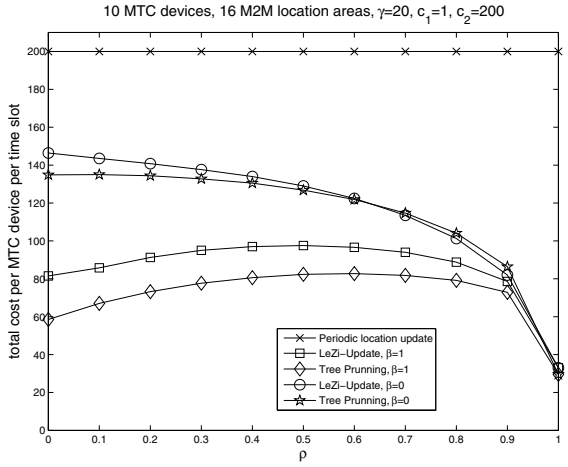


Fig. 4. The total cost of location update schemes, when there are 10 MTC devices.

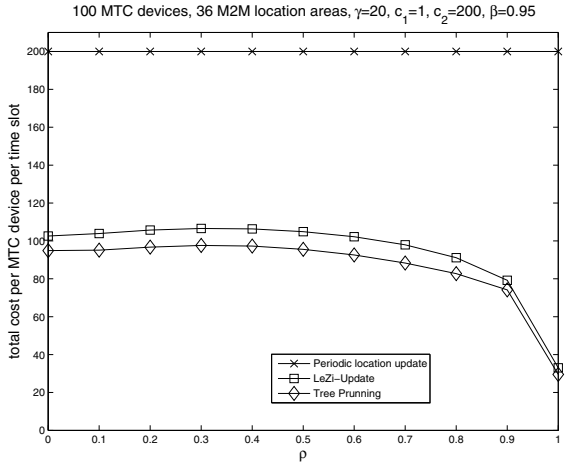


Fig. 5. The total cost of location update schemes, when there are 100 MTC devices.

If at time t_m, the MTC device has to prune the parsing tree, the value of q_i in Section IV is set to be the value of $q_{m,i}$. In addition, the value of $q_{i,j}$ in Section IV is set to be the value of $q_{m,i,j}$.

VII. Complexity Analysis

In this section, for the proposed location update algorithm, we analyze the worst-case computational complexity in time and space. Recall that $\Delta = \min\{(2\lceil \frac{S_{max}T}{L} \rceil + 1)^2, \alpha\}$ and $|N_1(i)| \leq \Delta$, $\forall i \in V$. Consider a MTC device in a time slot.

We now analyze the time complexity. First, it takes at most $O(\Delta)$ time to check if a new node should be added into the parsing tree in the current time slot. If a new node should be added, it takes $O(1)$ time to add the new node. Otherwise, it takes $O(1)$ time to update the current state pointer. Second, as shown in Section IV, it takes $O(k)$ time to calculate the value of $\mathbf{E}[H_{-k}]$, $\forall 0 \leq k \leq \gamma - 2$. Thus, it takes at most $O(\sum_{k=0}^{\gamma-2} k) = O(\gamma^2)$ time to obtain the value of k^*. Recall that $\Delta \leq \alpha$. Recall that the time complexity for estimating the location transition probabilities is $O(\alpha^2)$. Therefore, the time complexity of the proposed location update scheme is upper bounded by $O(\Delta + \gamma^2 + \alpha^2) = O(\gamma^2 + \alpha^2)$.

We now analyze the space/memory complexity. First, there are at most γ nodes in the parsing tree. Thus, it takes $\lceil \log_2(\gamma) \rceil$ bits to represent the index of a node in the parsing tree. Second, since there are α M2M location areas in the network, it takes $\lceil \log_2(\alpha) \rceil$ bits to represent the label of a node in the parsing tree. Third, a node in the parsing tree has at most Δ children. Recall that the space complexity for estimating the location transition probabilities is $O(\alpha^2)$. Therefore, the space complexity of the proposed location update algorithm is upper bounded by $O(\gamma(\lceil \log_2(\gamma) \rceil + \lceil \log_2(\alpha) \rceil + \Delta) + \alpha^2) = O(\gamma(\log_2(\gamma) + \log_2(\alpha) + \Delta) + \alpha^2)$.

VIII. Simulation Results

We wrote C programs in Windows 7 to obtain discrete event-driven simulation results [47] [48]. Each simulation instance consists of 100000 time slots. We also used ns-2 [49] in Fedora 17 to obtain additional simulation results. We

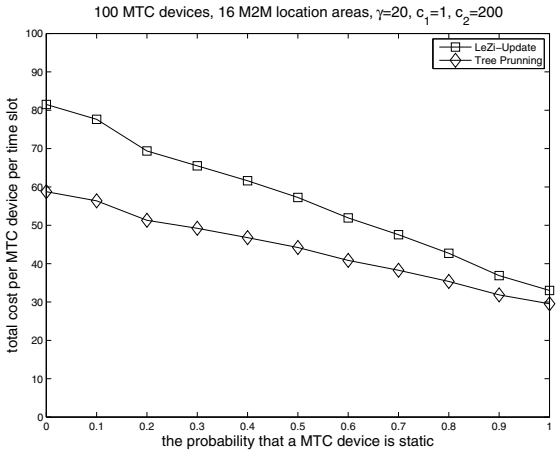


Fig. 6. The total cost of location update schemes in heterogeneous networks.

study the case in which the service region is a square and is partitioned into $\alpha = w^2$ M2M location areas of equal size. Let $\Lambda(i)$ be the set composed of the indexes of M2M location areas that are adjacent to the ith M2M location area. Note that $|\Lambda(i)| \in \{2, 3, 4\}$. For a MTC device in a time slot, the memory cost equals the total number of nodes in the parsing tree multiplied by the value of c_1. On the other hand, for a MTC device, the total energy cost equals the total number of location updates performed by the MTC device during the simulated time interval multiplied by the value of c_2. For a MTC device in a simulation instance, the total cost is defined to be the sum of the total memory cost and the total energy cost. We evaluate the proposed tree pruning algorithm, the LeZi-Update algorithm [6], and the naive periodic location update algorithm. When the LeZi-Update is used and it is necessary to prune the parsing tree, except for the root node, all nodes of the parsing tree are deleted. For a MTC device, when the naive periodic location update scheme is used, the memory cost per time slot is zero and the energy cost per location update equals c_2, regardless of the mobility pattern. As in many previous works on location update, it is assumed

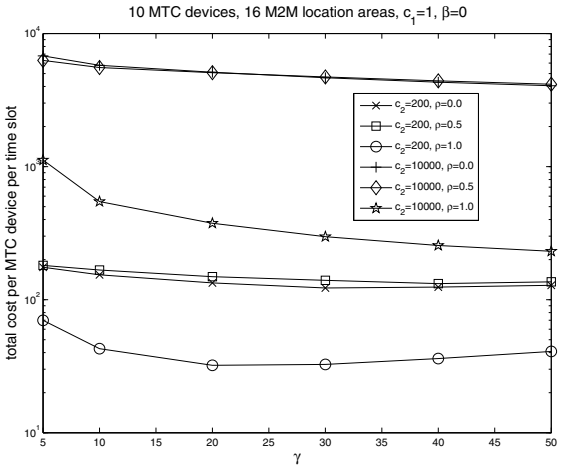


Fig. 7. The impacts of the maximum number of nodes in a parsing tree.

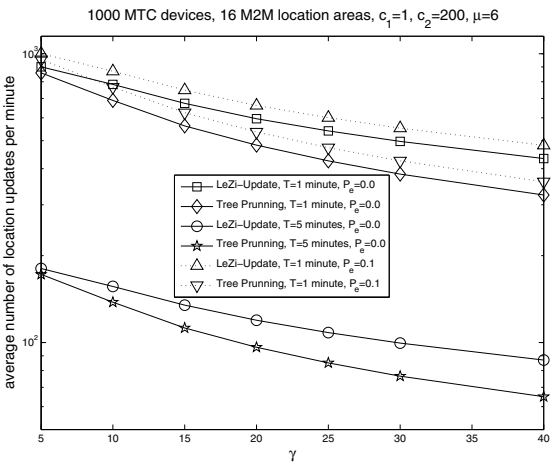


Fig. 9. The average number of location updates per minute.

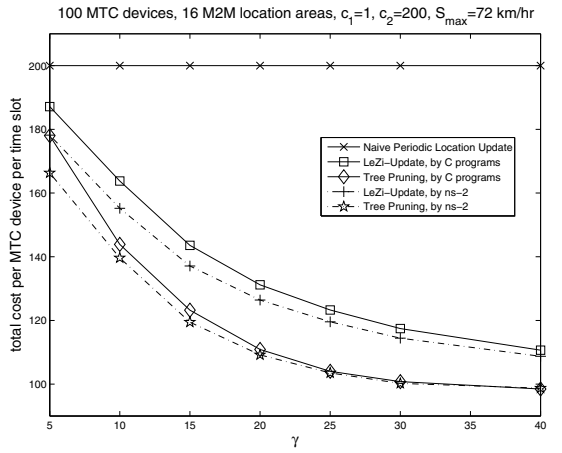


Fig. 8. The total cost when the maximum speed of a MTC device is 72 km/hr.

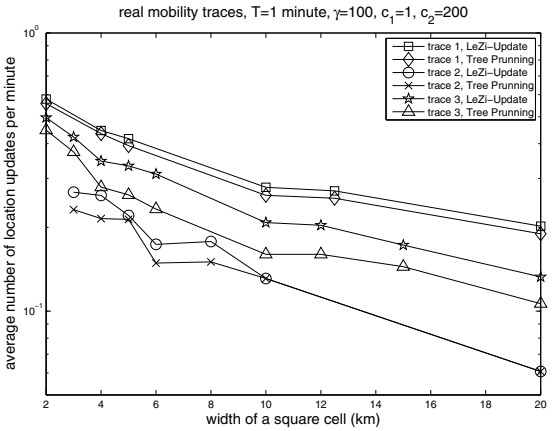


Fig. 10. The impacts of cell size for real mobility traces.

that the wireless channel is characterized by P_e, the probability that a location update message is not successfully received by the access point/base station. Unless explicitly stated, $P_e = 0$ in this section.

We use the following DTMC-based mobility model in the C programs. A MTC device is either static or mobile. The movements of two mobile MTC devices are statistically independent. It is assumed that for each m, $\{X_k^{(m)}\}_{k=0}^{\infty}$ is a DTMC and $X_0^{(m)}$ is an integer-valued random variable that is uniformly distributed in $[1, \alpha]$. After one time slot, a mobile MTC device either stays in the current M2M location area or moves to one of the adjacent M2M location areas. To model a variety of mobility patterns, the DTMC-based mobility model has three parameters, ρ_m, β_m, and h. ρ_m is the probability that the mth mobile device will stay in the current M2M location area after one time slot. β_m is the probability that the mth MTC device moves along a predetermined path/cycle. The function h is used to create the predetermined path/cycle. In particular, h is a one-to-one mapping from $\{1, 2, .., \alpha\}$ to $\{1, 2, .., \alpha\}$ such that $h(i) \in \Lambda(i)$, $\forall i$. In addition, given that the mth mobile device always moves along the predetermined path and it is currently in the ith M2M location area, it will move to the $h(i)$th M2M location area after one time slot. For the DTMC $\{X_k^{(m)}\}_{k=0}^{\infty}$, the corresponding state transition probabilities are set as follows.

$$P\{X_{k+1}^{(m)} = j | X_k^{(m)} = i\}$$

$$= \begin{cases} \rho_m, \ if \ j = i \\ (1-\rho_m)(\beta_m + \frac{1-\beta_m}{|\Lambda(i)|}), \ if \ j = h(i) \\ (1-\rho_m)(\frac{1-\beta_m}{|\Lambda(i)|}), \ if \ j \in \Lambda(i) - \{h(i)\} \\ 0, \ otherwise. \end{cases}$$

When $\rho_m = 0$ and $\beta_m = 1$, the mobility pattern of the mth MTC device is cyclic. Furthermore, when $\rho_m < 1$ and $\beta_m = 0$, $\{X_k^{(m)}\}_{k=0}^{\infty}$ corresponds to a symmetric random walk over the graph G.

Let $\rho, \beta \in [0, 1]$ be two real numbers. We study homogeneous networks in which $(\rho_m, \beta_m) = (\rho, \beta)$, $\forall m$. In this case the mobility patterns of all MTC devices are identical. In Figure 4, we show the total cost per MTC device per time slot, when $M = 10$, $\alpha = 16$, and $\beta \in \{0, 1\}$. In comparison with the naive periodic location update scheme, the proposed

location update scheme could reduce the total cost by at least 32%. When the movement pattern of each MTC device is cyclic, regardless of the value of ρ, the proposed location update scheme outperforms the LeZi-Update algorithm. In particular, the reduction in the total cost ranges from 7% to 28%. When the mobility pattern of each mobile device is cyclic, the proposed location update algorithm could use a small parsing tree to correctly predict the movements of a mobile device and reduce the frequency of location update. Therefore, in terms of the overall cost, the proposed location update algorithm outperforms the LeZi-Update algorithm. When the movements of each MTC device form a symmetric random walk and $\rho < 0.6$, the proposed location update algorithm is superior to the LeZi-Update algorithm. When the movements of each MTC device form a symmetric random walk and $\rho > 0.6$, the proposed location update algorithm is slightly inferior to the LeZi-Update algorithm. In Figure 5, we show the total cost per MTC device per time slot, when $M = 100$, $\alpha = 36$, and $\beta = 0.95$. In this case, regardless of the value of ρ, the proposed location update algorithm outperforms the LeZi-Update algorithm. In comparison with the periodic location update scheme, the proposed location update algorithm could reduce the total cost by at least 50%. In comparison with the LeZi-Update algorithm, the proposed location update algorithm could reduce the total cost by up to 10%.

We also study heterogeneous networks in which there are static MTC devices and mobile MTC devices in the network. In particular, if the mth MTC device is mobile, $(\rho_m, \beta_m) = (0, 1)$. On the other hand, if the mth MTC device is static $\rho_m = 1$. In Figure 6, we represent the total cost per MTC device per time slot as a function of the probability that a MTC device never moves. When either the proposed location update algorithm or the LeZi-Update algorithm is used, as the fraction of static MTC devices in the network increases, the cost of location update decreases. In comparison with the LeZi-Update algorithm, the proposed location update algorithm could reduce the total cost by up to 27%.

In Figure 7, we show the impacts of the maximum number of nodes in the parsing tree on the total cost, when $c_2 \in \{200, 10000\}$ and $(\rho_m, \beta_m) = (\rho, 0)$, $\forall m$. We find that the optimal value of γ depends on the value of ρ and c_2. For example, when $c_2 = 200$, $\rho = 0.0$, and $\gamma \in \{5, 10, 20, 30, 40, 50\}$, the total cost is minimized when $\gamma = 30$. In contrast, when $c_2 = 10000$, regardless of the value of ρ, the total cost is a decreasing function of γ. In this case, as the value of γ increases from 40 to 50, the total cost only slightly decreases. Thus, in practice, γ does not have to be a very large number.

We also use ns-2 with the following mobility model to evaluate the location update schemes. A MTC device moves along a circle in the service region. Let C_i be the circle associated with the ith MTC device. Let R_i be the radius of C_i in units of meters. Recall that L is the width of a M2M location area. Let (A_i, B_i) be the position of the center of C_i. In particular, R_i is a random variable that is uniformly distributed in $[R_{min}, R_{max}]$, where $R_{min} \leq R_{max} \leq \frac{wL}{2}$. In addition, A_i and B_i are random variables that are uniformly distributed in $[R_i, wL - R_i]$. Recall that S_{max} is the maximum speed of a MTC device. Let S_i be the speed of a MTC device in units of meters per second. S_i is a random variable that is uniformly distributed in $[0, S_{max}]$. In Figure 8, we show the impacts of γ on the total cost, when $L = 100$, $\alpha = 16$, $M = 100$, $(R_{min}, R_{max}) = (100, 200)$, $S_{max} = 20$, and $T = 60$. It should be emphasized that the proposed algorithm does not know the values of the variables in advance. Note that 20 meters per second equals 72 kilometers per hour. Regardless of the value of γ, the proposed location update algorithm outperforms the LeZi-Update algorithm and the naive periodic location update algorithm. Simulation results obtained by our C programs are consistent with simulation results obtained by ns-2. Based on ns-2 simulation results, in comparison with the LeZi-Update algorithm, the proposed location update algorithm could reduce the total cost by up to 13%.

In Figure 9, we show the average number of location updates per minute, when there are 1000 MTC devices in the network and the sensing period is either one minute or five minutes. Note that Figure 8 and Figure 9 use the same mobility model. Let μ be the maximum number of times that a location update message is retransmitted whenever necessary. Regardless of the value of (γ, P_e), the proposed location update algorithm outperforms the LeZi-Update algorithm and the naive periodic location update algorithm. When $P_e = 0$, in comparison with the LeZi-Update algorithm, the proposed location update algorithm could reduce the average number of location updates per time unit by up to 25%.

In Figure 10, we show the performance of the location update algorithms, when real mobility traces obtained from [50] are used. For the first mobility trace, in comparison with the naive periodic location update algorithm, the proposed location update algorithm could reduce the average number of location updates per time unit by up to 81%. In comparison with the LeZi-Update algorithm, the proposed location update algorithm could reduce the average number of location updates per time unit by up to 6.39%. For the third mobility trace, in comparison with the LeZi-Update algorithm, the proposed location update algorithm could reduce the average number of location updates per time unit by up to 25.15%.

IX. Conclusion

In this paper, we have proposed a novel energy-and-memory efficient location update scheme for wireless M2M communications. In comparison with periodically registering to the MTC server, it is more efficient for a MTC device to perform location updates only when new nodes are added into the corresponding parsing tree. Based on the theory of random walks over trees, we have proposed optimally pruning the parsing tree to minimize the expected value of the sum of the memory cost and the energy cost for a MTC device. We have found that the proposed scheme could significantly reduce the location update cost. Future works include jointly optimizing the location update scheme and the paging scheme for wireless M2M communications. Another direction of future research is to explore and exploit the statistical correlations among the movements of MTC devices for mobility management. Utilizing cloud computing technologies for large-scale implementation of the decoding part of the proposed location update algorithm is a promising direction of future research.

Future works also include collaborating with industry to create prototype to test the performance of the proposed algorithm. In our opinion, dynamically adjusting the sensing period is a good direction for future research.

APPENDIX

A. Key Variables

M: the total number of MTC devices.

α: the total number of M2M location areas.

L: the width of a M2M location area.

$(x_m(t), y_m(t))$: the position of the mth MTC device at time t.

S_{max}: the maximum speed of a MTC device.

T: a positive real number that represents the length of time slot.

$X_k^{(m)}$: a random variable that represents the index of the M2M location area in which the mth MTC device resides at time $k \cdot T$.

$G = (V, E)$: a graph such that each vertex corresponds to a M2M location area and $(i, j) \in E$ if and only if there exists a MTC device that might move from the ith M2M location area to the jth M2M location area in one time slot.

$N_1(i) = \{j | (i, j) \in E\}$: the adjacent vertices of vertex i.

$\Delta = \min\{(2\lceil \frac{S_{max}T}{L} \rceil + 1)^2, \alpha\}$.

$d(i, j)$: the distance between vertex i and vertex j in the graph G.

X_k: an abbreviation of $X_k^{(m)}$, when only one MTC device is taken into consideration.

s_i: the ith substring in the encoding dictionary for the proposed location update algorithm.

v_i: the (maximal proper) prefix of s_i.

w_i: the last symbol of s_i.

$f(i)$: an integer such that $s_{f(i)} = v_i$.

γ: the maximum number of nodes in the parsing tree.

L_i: the label of node i in the parsing tree.

Φ_i: the set composed of the indexes of the children of node i in the parsing tree.

$N(t)$: the total number of nodes in the parsing tree at time t.

$t_k = k \times T$: the time instance when a MTC device performs sensing for the kth time.

S_k: the index of the node that is pointed by the current state pointer just before time t_k.

Θ_k: a set that is composed of the labels of the children of node (with index) S_k in the parsing tree.

Γ: a parsing tree.

Γ_{-k}: the tree obtained by removing each node with index greater than k from the tree Γ.

$\Phi_{-k}(i)$: a set that is composed of the indexes of the children of node i in the tree Γ_{-k}.

$\{Y_n^{-k}\}_{n=0}^{\infty}$: a discrete-time Markov chain that is associated with the tree Γ_{-k}.

q_i: the estimated value of the probability that the MTC device will appear in the ith M2M location area.

$q_{i,j}$: the estimated value of the probability that the MTC device will be in the jth M2M location area at the next sensing time instance given that the MTC device is currently in the ith M2M location area.

τ: the index of the M2M location area in which the MTC device currently resides.

$H_{-k} = \min\{n : n \geq 1, Y_n^{-k} = -1 | Y_0^{-k} = 0\}$.

Y_n: an abbreviation for Y_n^{-k}, when the value of k is fixed.

$z_i = \mathbf{E}[\min\{n : n \geq 1, Y_n = -1\} | Y_0 = i]$. In addition, $\mathbf{z} = (z_0, z_1, ..z_k)^T$.

$\mathbf{U}$: a $(k+1) \times (k+1)$ matrix such that $[\mathbf{U}]_{i+1,j+1} = P\{Y_{n+1} = j | Y_n = i\}$, $\forall 0 \leq i, j \leq k$.

c_1, c_2: two positive real numbers that correspond to the unitary memory cost and unitary energy cost, respectively.

$k^* = \arg\min_{k:k\in\{0,1,2,..,\gamma-2\}} c_1 \times (k+1) - c_2 \times \mathbf{E}[H_{-k}]$.

$\Lambda(i)$: a set that is composed of the indexes of M2M location areas that are adjacent to the ith M2M location area.

ρ_m: the probability that the mth MTC device will stay in the current M2M location area after one time slot.

β_m: the probability that the mobility pattern of the mth MTC device is cyclic.

B. Pseudo Codes for Maintaining The Parsing Tree

For the convenience of readers, we include pseudo codes for maintaining the parsing tree in Algorithm 1. In particular, some of the pseudo codes use the syntax of the C programming language.

C. Proofs

Lemma 1: If $\frac{c_2}{c_1} < 1$, $k^* = 0$.

Proof:

1. Define $f(k) = c_1 \cdot (k+1) - c_2 \cdot \mathbf{E}[H_{-k}]$, $\forall k \in \{0, 1, 2, .., \gamma - 2\}$. Then, $f(0) = c_1 - c_2$.

2. When $k \geq 1$,

$$\begin{aligned} f(k) &\geq c_1(k+1) - c_2(k+1) \\ &= (c_1 - c_2)(k+1) \\ &> c_1 - c_2 \\ &= f(0). \end{aligned}$$

Since $H_{-k} \leq k+1$ for sure, $\mathbf{E}[H_{-k}] \leq k+1$. Thus, we have the first inequality. Since $c_1, c_2 > 0$ and $\frac{c_2}{c_1} < 1$, $c_1 - c_2 > 0$. In addition, $k + 1 > 1$, $\forall k \geq 1$. Thus, we have the third inequality.

3. Since $f(0) < f(k)$, $\forall k \geq 1$, $k^* = \arg\min_{0\leq k\leq\gamma-2} f(k) = 0$.

QED.

Lemma 2: If there exists an integer j, where $j \in \Phi_0$ and $j \neq \gamma - 1$, such that $c_2 \cdot q_{\tau,L_j} - c_1 \cdot j > 0$, $k^* \geq 1$.

Proof:

1. Define $f(k) = c_1 \cdot (k+1) - c_2 \cdot \mathbf{E}[H_{-k}]$, $\forall k \in \{0, 1, 2, .., \gamma - 2\}$. Then, $f(0) = c_1 - c_2$.

2. $\mathbf{E}[H_{-j}] \geq 1 + q_{\tau,L_j}$. Thus, $f(j) \leq c_1(j+1) - c_2(1 + q_{\tau,L_j})$.

4. $f(0) - f(j) \geq c_2 \cdot q_{\tau,L_j} - c_1 \cdot j > 0$.

5. Thus, $k^* \neq 0$ and therefore $k^* \geq 1$.

QED.

Theorem 1: If $q_i = v_i$, $\forall i$, $q_{i,j} = p_{i,j}$, $\forall i, j$, $\frac{c_2}{c_1} < 1$, $\alpha < \gamma - 1$, the DTMC $\{X_k\}_{k=1}^{\infty}$ is irreducible, and $P\{X_{k+1} = i | X_k = i\} > 0$, $\forall 1 \leq i \leq \alpha$, the DTMC $\{Z_n\}_{n=1}^{\infty}$ is irreducible, aperiodic, and positive-recurrent.

Proof:

Algorithm 1 The Parsing Tree Formation Algorithm

```
// initialize the parsing tree
if initialization then
    parsingTree.root = (struct treeNode *) malloc(sizeof(struct treeNode));
    parsingTree.root→index = 0; // the index of the root node is 0
    parsingTree.root→label = -1; // the label of the root node is ∅
    parsingTree.root→totalChildren = 0; // the total number of children of the root node
    parsingTree.root→childNode = (struct treeNode **) malloc(totalBaseStation*sizeof(struct treeNode *));
    parsingTree.currentNode = parsingTree.root; // current state pointer
    parsingTree.nextId = 1; // index for the next node added into the parsing tree
end if
// update the parsing tree when obtaining the kth sensing result, X_k
if getNewSensingResult(k) then
    add X_k into the tail of the queue;
    if X_k ∈ Θ_k then
        // does not perform location updates or add new nodes into the parsing tree
        j = findChildIndex(X_k,Φ(S_k)); // find j such that j ∈ Φ(S_k) and L_j = X_k;
        parsingTree.currentNode = node j in the parsing tree; // update the current state pointer
    else
        // add a new node into the parsing tree
        totalChildren = parsingTree.currentNode.totalChildren;
        parsingTree.currentNode.child[totalChildren] = (struct treeNode *) malloc(sizeof(struct treeNode));
        parsingTree.currentNode.child[totalChildren].index = parsingTree.nextId;
        parsingTree.currentNode.child[totalChildren].label = X_k;
        parsingTree.currentNode.totalChildren++;
        parsingTree.currentNode = parsingTree.root;
        parsingTree.nextId++;
        // perform a location update
        clear the queue;
    end if
end if
```

1. We first prove that the DTMC is irreducible. Let $\xi_1, \xi_2 \in \Omega_Z$ be two distinct states. Consider the case in which $Z_1 = \xi_1$. Since $\xi_1 \in \Omega_Z$, there exists an integer $\theta_1 \in \{1, 2, 3, .., \alpha\}$ such that given that $X_0 = \theta_1$, with probability greater than zero, $Z_1 = \xi_1$. Similarly, there exists an integer $\theta_2 \in \{1, 2, 3, .., \alpha\}$ such that given that $X_0 = \theta_2$, with probability greater than zero, $Z_1 = \xi_2$.

2. At time V_1, the parsing tree is pruned and only the root node is kept based on Lemma 1. Recall that $d(i, j)$ is the total number of edges on the shortest path from vertex i to vertex j in the graph G. Define $N_1 = d(X_{V_1}, \theta_2)$. Since the DTMC $\{X_k\}_{k=1}^{\infty}$ is irreducible, with probability greater than zero, $X_{V_1+N_1} = \theta_2$. Since $d(X_{V_1}, \theta_2) < \alpha$ for sure and $\alpha < \gamma - 1$, $N_1 < \gamma - 1$ for sure. Due to that $N_1 < \gamma - 1$ and at most one node is added into the parsing tree in a sensing period, $V_2 > V_1 + N_1$ for sure. Since $P\{X_{k+1} = \theta_2 | X_k = \theta_2\} > 0$, given that $X_{V_1+N_1} = \theta_2$, with probability greater than zero, $X_{V_1+N_1} = X_{V_1+N_1+1} = .. = X_{V_2} = \theta_2$.

3. At time V_2, the parsing tree is pruned and only the root node is kept based on Lemma 1. In addition, based on the definition of θ_2, given that $X_{V_2} = \theta_2$, with probability greater than zero, $Z_3 = \xi_2$.

4. Based on 2 and 3, $P\{Z_3 = \xi_2 | Z_1 = \xi_1\} > 0$ and therefore the DTMC is irreducible.

5. We now prove that the DTMC is aperiodic by showing that $P\{Z_3 = \xi_1 | Z_1 = \xi_1\} > 0$ and $P\{Z_4 = \xi_1 | Z_1 = \xi_1\} > 0$. Consider the case in which $Z_1 = \xi_1$. At time V_1, the parsing tree is pruned and only the root node is kept based on Lemma 1. Similar to 2, it can be proved that with probability greater than zero, $X_{V_2} = \theta_1$.

6. At time V_2, the parsing tree is pruned and only the root node is kept based on Lemma 1. In addition, based on the definition of θ_1, given that $X_{V_2} = \theta_1$, with probability greater than zero, $Z_3 = \xi_1$.

7. Based on 5 and 6, $P\{Z_3 = \xi_1 | Z_1 = \xi_1\} > 0$.

8. We now prove that $P\{Z_4 = \xi_1 | Z_1 = \xi_1\} > 0$. At time V_1, the parsing tree is pruned and only the root node is kept based on Lemma 1. Since $P\{X_{k+1} = i | X_k = i\} > 0$, $\forall i$, with probability greater than zero, $X_{V_1} = X_{V_1+1} = .. = X_{V_2}$.

9. Similar to 5, given that $X_{V_2} = X_{V_1}$, with probability greater than zero, $X_{V_3} = \theta_1$. Similar to 6, given that $X_{V_3} = \theta_1$, with probability greater than zero, $Z_4 = \xi_1$.

10. Based on 8 and 9, $P\{Z_4 = \xi_1 | Z_1 = \xi_1\} > 0$

11. Based on 7 and 10, due to that the greatest common divisor of $3 - 1 = 2$ and $4 - 1 = 3$ is 1, the period of the state ξ_1 is one. Since the DTMC is irreducible, all states have the same period and therefore the DTMC is aperiodic.

12. Since $|\Omega_Z| < \infty$, the DTMC has at least one positive-recurrent state. Since the DTMC is irreducible and has at least one positive-recurrent state, all states are positive-recurrent.

QED.

REFERENCES

[1] G. Wu, S. Talwar, K. Johnsson, N. Himayat, and K. D. Johnson, "M2M: from mobile to embedded internet," *IEEE Commun. Mag.*, pp. 36–43, Apr. 2011.

[2] M. Zorzi, A. Gluhak, S. Lange, and A. Bassi, "From today's intranet of things to a future Internet of things: a wireless- and mobility-related view," *IEEE Wireless Commun.*, pp. 44–51, Dec. 2010.

[3] "Study on facilitating machine to machine communications in 3GPP systems," 3GPP TR 22.868.

[4] "System improvements for machine-type communications," 3GPP TR 23.888, V1.0.0, July 2010.

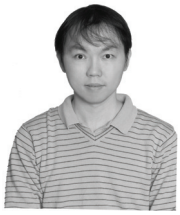
[5] "Service requirements for machine-type communications," 3GPP TS 22.368, V10.2.0, Sept. 2010.

[6] A. Bhattacharya and S. K. Das, "LeZi-update: an information-theoretic approach to track mobile users in PCS networks," in *Proc. 1999 ACM MobiCom*, pp. 1–12.

[7] A. Roy, A. Misra, and S. K. Das, "Location update vs. paging tradeoff in cellular networks: an approach based on vector quantization," *IEEE Trans. Mobile Comput.*, vol. 6, no. 12, pp. 1426–1440, Dec. 2007.

[8] A. Misra, A. Roy, and S. K. Das, "Information-theory based optimal location management schemes for integrated multi-system wireless networks," *IEEE/ACM Trans. Netw.*, vol. 16, no. 3, pp. 525–538, June 2008.

[9] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, 2nd edition. John Wiley and Sons, 2006.
[10] S. M. Ross, *Stochastic Processes*, 2nd edition. John Wiley and Sons, 1996.
[11] G. Lopez, P. Moura, J. I. Moreno, and A. de Almeida, "ENERsip: M2M-based platform to enable energy efficiency with energy-positive neighbors," in *Proc. 2011 IEEE INFOCOM Workshop M2MCN*.
[12] Y. Chen and W. Wang, "Machine-to-machine communications in LTE-A," in *Proc. 2010 IEEE VTC*.
[13] S.-Y. Lien and K.-C. Chen, "Massive access management for QoS guarantees in 3GPP machine-to-machine communications," *IEEE Commun. Lett.*, vol. 15, no. 3, pp. 311–313, Mar. 2011.
[14] R. Yu, Y. Zhang, Y. Chen, C. Huang, Y. Xiao, and M. Guizani, "Distributed rate and admission control in home M2M networks: a non-cooperative game approach," in *Proc. 2011 IEEE INFOCOM Workshop M2MCN*.
[15] R.-H. Gau and Z. J. Haas, "Concurrent search of mobile users in cellular networks," *IEEE/ACM Trans. Netw.*, vol. 12, no. 1, pp. 117–130, Feb. 2004.
[16] R.-H. Gau and C.-W. Lin, "Location management of correlated mobile users in The UMTS," *IEEE Trans. Mobile Comput.*, vol. 4, no. 6, pp. 641–651, Nov. 2005.
[17] I. F. Akyildiz, J. S. M. Ho, and Y.-B. Lin, "Movement-based location update and selective paging for PCS networks," *IEEE/ACM Trans. Netw.*, vol. 4, no. 4, pp. 629–638, Aug. 1996.
[18] C. Rose and R. Yates, "Minimizing the average cost of paging and registration: a timer-based method," *Wireless Netw.*, vol. 2, no. 2, pp. 109–116, June 1996.
[19] J. S. M. Ho and I. F. Akyildiz, "Mobile user location update and paging under delay constraints," *Wireless Netw.*, vol. 1, no. 4, pp. 413–425, Dec. 1995.
[20] I. F. Akyildiz and J. S. M. Ho, "Dynamic mobile user-location update for wireless PCS networks," *Wireless Netw.*, vol. 1, no. 2, pp. 187–196, July 1995.
[21] G. P. Pollini and C.-L. I, "A profile-based location strategy and its performance," *IEEE J. Sel. Areas Commun.*, vol. 15, no. 8, pp. 1415–1424, Oct. 1997.
[22] C. Rose, "State-based paging/registration: a greedy technique," *IEEE Trans. Veh. Technol.*, vol. 48, no. 1, pp. 166–173, Jan. 1999.
[23] G. Wan and E. Lin, "A dynamic paging scheme for wireless communication systems," in *Proc. 1997 ACM MobiCom*, pp. 195–203.
[24] A. Bar-Noy, I. Kessler, and M. Sidi, "Mobile users: to update or not to update?" *Wireless Netw.*, vol. 1, no. 2, pp. 175–185, July 1995.
[25] U. Madhow, M. L. Honig, and K. Steiglitz, "Optimization of wireless resources for personal communications mobility tracking," *IEEE/ACM Trans. Netw.*, vol. 3, no. 6, pp. 698–707, Dec. 1995.
[26] Z. Liu and T. D. Bui, "Dynamic mobile terminal location registration in wireless PCS networks," *IEEE Trans. Mobile Comput.*, vol. 4, no. 6, pp. 630–640, Nov. 2005.
[27] B. Liang and Z. J. Haas, "Predictive distance-based mobility management for multidimensional PCS networks," *IEEE/ACM Trans. Netw.*, vol. 11, no. 5, pp. 718–732, Oct. 2003.
[28] E. Cayirci and I. F. Akyildiz, "User mobility pattern scheme for location update and paging in wireless systems," *IEEE Trans. Mobile Comput.*, vol. 1, no. 3, pp. 236–247, July 2002.
[29] X. Wu, B. Mukherjee, and B. Bhargava, "A crossing-tier location update/paging scheme in hierarchical cellular networks," *IEEE Trans. Wireless Commun.*, vol. 5, no. 4, pp. 839–848, Apr. 2006.
[30] C. K. Ng and H. W. Chan, "Enhanced distance-based location management of mobile communication systems using a cell coordinates approach," *IEEE Trans. Mobile Comput.*, vol. 4, no. 1, pp. 41–55, Jan. 2005.
[31] S. J. Kim and C. Y. Lee, "Modeling and analysis of the dynamic location registration and paging in microcellular systems," *IEEE Trans. Veh. Technol.*, vol. 45, no. 1, pp. 82–90, Feb. 1996.
[32] A. Abutaleb and V. O. K. Li, "Location update optimization in personal communication systems," *Wireless Netw.*, vol. 3, no. 3, pp. 205–216, 1997.
[33] E. Cayirci and I. F. Akyildiz, "Optimal location area design to minimize registration signaling traffic in wireless systems," *IEEE Trans. Mobile Comput.*, vol. 2, no. 1, pp. 76–85, Jan. 2003.
[34] I. Demirkol, C. Ersoy, U. Caglayan, and H. Delic, "Location area planning and cell-to-switch assignment in cellular networks," *IEEE Trans. Wireless Commun.*, vol. 3, no. 3, pp. 880–890, May 2004.
[35] P. S. Bhattacharjee, D. Saha, and A. Mukherjee, "An approach for location area planning in a personal communications services network (PCSN)," *IEEE Trans. Wireless Commun.*, vol. 3, no. 4, pp. 1176–1187, July 2004.
[36] S. K. Das and S. K. Sen, "A new location update strategy for cellular networks and its implementation using a genetic algorithm," in *Proc. 1997 ACM MobiCom*, pp. 185–194.
[37] A. Bar-Noy and I. Kessler, "Tracking mobile users in wireless communication networks," *IEEE Trans. Inf. Theory*, vol. 39, no. 6, pp. 1877–1886, Nov. 1993.
[38] A. Hac and X. Zhou, "Location strategies for personal communication networks: a novel tracking strategy," *IEEE J. Sel. Areas Commun.*, vol. 15, no. 8, pp. 1425–1436, Oct. 1997.
[39] G. Varsamopoulos and S. K. S. Gupta, "Optimal offline and online registration techniques for location management with overlapping registration areas," *IEEE Trans. Mobile Comput.*, vol. 4, no. 5, pp. 474–488, Sept. 2005.
[40] J. Li, H. Kameda, and K. Li, "Optimal dynamic mobility management for PCS networks," *IEEE/ACM Trans. Netw.*, vol. 8, no. 3, pp. 319–327, June 2000.
[41] B. Hajek, K. Mitzel, and S. Yang, "Paging and registration in cellular networks: jointly optimal policies and an iterative algorithm," *IEEE Trans. Inf. Theory*, vol. 54, no. 2, pp. 608–622, Feb. 2008.
[42] W. S. Jeon and D. G. Jeong, "Performance of improved probabilistic location update scheme for cellular mobile networks," *IEEE Trans. Veh. Technol.*, vol. 49, no. 6, pp. 2164–2173, Nov. 2000.
[43] Z. Mao and C. Douligeris, "Group registration with local anchor for location tracking in mobile networks," *IEEE Trans. Mobile Comput.*, vol. 5, no. 5, pp. 583–595, May 2006.
[44] G. Chartrand and L. Lesniak, *Graphs and Digraphs*. Chapman and Hall/CRC, 2005.
[45] E. Horowitz, S. Sahni, and D. P. Mehta, *Fundamentals of Data Structures in C++*, 2nd edition. Silicon Press, 2007.
[46] J. Ziv and A. Lempel, "Compression of individual sequences via variable-rate coding," *IEEE Trans. Inf. Theory*, vol. 24, no. 5, pp. 530–536, Sept. 1978.
[47] S. M. Ross, *Simulation*, 4th edition. Academic Press, 2006.
[48] A. M. Law, *Simulation Modeling and Analysis*, 4th edition. McGraw-Hill, 2007.
[49] ns-2. Available: http://nsnam.isi.edu/nsnam/index.php/Main_Page.
[50] Available: http://www.cise.ufl.edu/~helmy/MobiLib.htm



Rung-Hung Gau received the BS degree in electrical engineering from National Taiwan University, Taipei, Taiwan, in 1994 and the MS degree in electrical engineering from University of California at Los Angeles, Los Angeles, California, USA, in 1997. He received the PhD degree in electrical and computer engineering from Cornell University, Ithaca, New York, USA, in 2001. He is currently a professor in the Institute of Communications Engineering, National Chiao Tung University, Hsinchu, Taiwan. His research interests include cross-layer medium access control in wireless networks, mobility management, M2M communications, network coding, cloud computing, and wireless/mobile ad-hoc networks. His URL is http://cmbsd.cm.nctu.edu.tw/~runghung/.

Ching-Pei Cheng received the BS degree in electrical engineering from National Chiao Tung University, Hsinchu, Taiwan, in 2011. She is currently a graduate student in the Institute of Communications Engineering, National Chiao Tung University, Hsinchu, Taiwan. Her research interests include M2M communications, mobility management, network coding, and smart phone applications.