Energy-Efficient Anonymous Multicast in Mobile Ad-Hoc Networks*

Jung-Chun Kao and Radu Marculescu
Carnegie Mellon University
Pittsburgh, PA 15213-3890, USA
{jckao, radum}@cmu.edu

Abstract

*Protecting personal privacy and energy efficiency are two primary concerns for mobile ad hoc networks. However, no energy-efficient multicast algorithm designed for preserving anonymity has been proposed to date. At the same time, existing approaches cannot be applied to anonymous routing due to their incapability of preserving anonymity. To solve this critical issue, we propose an energy-efficient anonymous multicast algorithm (*EEAMA*), which relies only on the statistical properties of the wireless network. This not only makes EEAMA suitable to preserving anonymity, but also reduces its execution time significantly. The complexity of EEAMA increases polynomially with the size of the multicast group,* as opposed to *the size of the network which determines the complexity of all approaches in the literature. Extensive simulation results show that compared to anonymous unicast, EEAMA offers both better performance (in terms of packet delivery ratio, end-to-end delay and network throughput) and significant energy savings.*

Keywords: *Anonymous routing, energy-efficiency, multicast, mobile ad hoc networks*

1. Introduction and related work

Protecting personal privacy is of primary concern for emerging mobile ad hoc networks (*MANET*) and wireless sensor networks (*WSN*). At the same time, energy efficiency is of primary importance in such networks, especially when network nodes are powered by batteries. At first, energy efficiency and privacy protection seem contradictory goals; this is because privacy protection prevents the existing energy-efficient unicast/multicast algorithms from knowing the critical information (*e.g.,* node location and route details) these algorithms rely on. This apparent contradiction leads to a unique optimization problem, which is the main objective of this paper. Important privacy information includes identity, location and communication contents. As an important component of privacy, *user/location anonymity* can improve security significantly by making adversaries unable to identify the potential victims (for instance, based on the information carried in the eavesdropped packets and compromised nodes) and conduct target-specific attacks.

There exists some work on anonymous unicast in mobile ad-hoc networks and little work on anonymous multicast. More precisely, regarding the anonymous unicast, Kong and El-Khatib respectively propose ANODR (*AN*onymous *O*n *D*emand *R*outing) [1] and SDDR (*S*ecure *D*ynamic *D*istributed *R*outing) [2]. In order to provide anonymity in a stronger adversary model, Zhu et al. categorize strict anonymity into *identity privacy*, *location privacy*, and *route anonymity*, and then propose ASR (*A*nonymous *S*ecure *R*outing) in [3]. In order to reduce the huge overhead caused by key pair generation and asymmetric cryptographic operation, Seys and Preneel proposed the ARM protocol in [4]. To further speed up the routing decisions and ultimately be able to route *real-time traffic*[1], Kao and Marculescu propose the ASC protocol in [5] which combines transmission power control as a means to improving both network performance and security [14].

With respect to anonymous multicast, research has been aimed at achieving *loose* anonymity, but none of the proposed solutions can achieve *strict* anonymity (*i.e.,* identity privacy, location privacy and route anonymity as defined by Zhu et al. in [3]). More precisely, the solutions in [7] and [8] provide either sender anonymity or receiver anonymity, but not both. To provide mutual anonymity (*i.e.,* both sender and receiver anonymity), Xiao et al. proposed the MAM protocol in [9]. While being able to provide sender, receiver or mutual anonymity, *none* of the anonymous multicast protocols proposed to date can actually achieve strict anonymity.

Moreover, *none* of the anonymous multicast protocols in the literature is designed for energy efficiency. This is because strict anonymity voids the availability of the

* This work is supported by CMU CyLab Army Research Office (ARO) under grant 9097.74.9 and by Gigascale Systems Research Focus Center, one of the five research centers funded under the FCRP, a SRC program, under grant 10913.2.

[1] According to [6], the maximum one-way end-to-end delay acceptable for real-time traffic is 150ms.

978-1-4244-1890-9/07/$25.00 ©2007 IEEE

information critical to energy savings such as node location and route details. Without such information, it *seems* impossible to design an anonymous multicast algorithm which would also be energy-efficient.

For the reason above, the traditional energy-efficient multicast algorithms designed for MANETs without anonymity concerns (*e.g.,* [10]-[11]) *cannot* be applied to MANETs where strict anonymity is important.[2] Traditional approaches typically save energy by exploiting multihop routing (*i.e.,* packets are sent through several short links rather than one long link) and wireless multicast advantage[3] (*i.e.,* within one transmission multiple receivers can receive packets). Because both techniques require route details or node location in advance, traditional approaches *do not* work with anonymity.

A possible way for anonymous multicast to improve the energy efficiency is to rely on *predicted* route details, rather than using the actual information. Towards this end, in this paper, we propose the *E*nergy-*E*fficient *A*nonymous *M*ulticast *A*lgorithm (EEAMA). Unlike existing energy-efficient multicast algorithms, EEAMA's capability of saving energy comes from exploiting the *statistical properties* of the network. More specifically, instead of exchanging the *actual* route details (*e.g.,* intermediate nodes, hop count, link lengths and path energy), the route details are *predicted* based on the statistical properties of the network. To have accurate predictions, we derive *lower and upper bounds* for the path energy and so the predicted path energy fed into EEAMA is a weighted sum of these bounds. Based on this prediction, EEAMA decides how to send packets to destination nodes. Relying on prediction instead of the actual route details is the key to energy efficiency when enforcing anonymity; this makes EEAMA unique.

EEAMA consists of several iterations. During each iteration, EEAMA decides *where* to send packets—a subset of destination nodes—and *how* to send packets—via a few multihop unicasts, a one-hop multicast, or a combination thereof. The complexity of the EEAMA algorithm is $O(g^3)$, where g is the multicast group size. As such, EEAMA complexity is independent of the network size, and so it is extremely scalable.

EEAMA can run on top of *any* anonymous unicast protocol (*e.g.,* ANODR, ASR, ASC, etc.) as an extension meant to support energy-efficient multicasting. This is because EEAMA either sends packets through the routes established (and maintained) by the underlying unicast protocol or further relays packets (meant to reach a subset of group member nodes) in a one-hop manner. We note that EEAMA does not establish extra connections. Therefore, as long as the underlying unicast protocol satisfies the anonymity requirements, EEAMA is able to save energy without sacrificing anonymity.

The remaining of this paper is organized as follows. Section 2 introduces the energy model. The restrictions on multicast trees under anonymity requirements are presented in Section 3. Section 4 presents the theoretical analysis and the newly derived theoretical bounds. Section 5 describes the energy-efficient multicast algorithm we propose, while the simulation results are given in Section 6. Finally, Section 7 presents the concluding remarks.

2. Energy model

In this paper, we use the energy model in [10]. Basically, we consider omni-directional antennas and uniform propagation conditions. If the sender sends a signal at transmission power P_0, then the received signal power is $P_0x^{-\alpha}$, where α is the propagation loss exponent and x is the distance between sender and receiver. Typically, the propagation loss exponent takes a value between 2 to 4, depending on the wireless environment characteristics.

Assuming that the noise power level is N and the threshold of the signal-to-noise ratio (*SNR*) required for a successful signal reception is β, the minimum transmission power required for a successful link communication is $N\beta x^{\alpha}$. Without loss of generality, we normalize to 1 the actual transmission energy required for a successful transmission of a unit-size packet while traversing a unit distance. Therefore, the minimum transmission energy required for a unit-size packet to traverse a distance of x units, denoted by $E_L(x)$, can be normalized as $E_L(x) = x^{\alpha}$.

This normalized minimum transmission energy is referred to as *link energy*. The *path energy* of a given path is defined as the sum of *all* link energy values along the path. Given a source-destination pair, the *minimum path energy* is defined as the smallest path energy over *all* possible paths which connect the given source and destination pair.

3. Anonymity concerns and multicast support

Next, we discuss the restrictions on constructing a multicast tree due to anonymity concerns. Our multicast algorithm (EEAMA) that complies with these restrictions will be detailed later in Section 5.

3.1 Available information with anonymity concerns

In EEAMA, a node (called *sink*) can join a group by initiating an *on-demand* join request to a group leader (called *source*) based on the underlying anonymous unicast protocol. From the perspective of the underlying unicast protocol, the join process is no different from establishing a new connection between the source and the sink (except that a field of sink location should be attached in an encrypted form). Similarly, a sink can leave a group via a

[2] In this paper, "anonymity" means "strict anonymity" unless otherwise specified.

[3] The term "wireless multicast advantage" is from [10].

leave process. After the join process, a (unicast) route is established and refreshed *dynamically* by the underlying unicast protocol. This route is maintained until the sink leaves the group. We note that due to the strict anonymity requirements [3], the route details such as intermediate nodes, hop count, link lengths and path energy are *not* available. Because route details are hidden from the source and sink, besides statistical properties of the network (*e.g.,* average node density of the network[4]), a source is only aware of:

- The existence of routes established during join processes (route details are unknown though)
- The location of the sinks in its multicast group

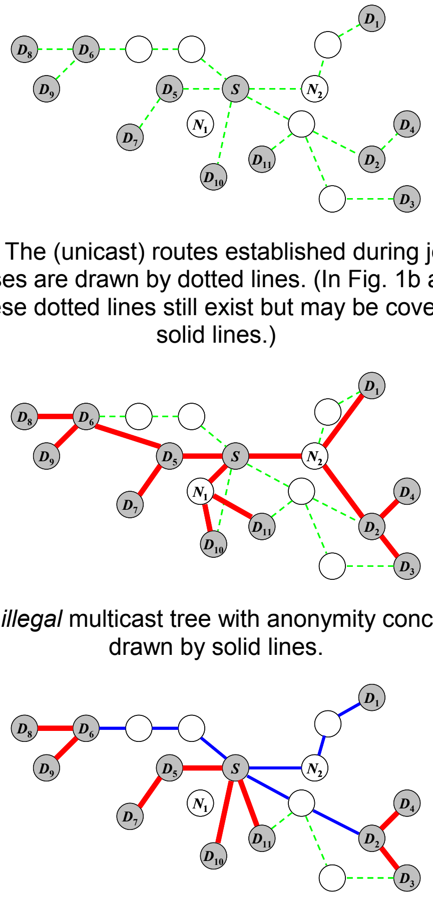
3.2 Multicast tree under anonymity concerns

Since the *only* information available to the source node is that about its *member nodes* (*i.e.,* sinks and itself), the anonymity concerns restrict the construction of legal multicast trees. In addition, an anonymous multicast algorithm itself should *not* initiate any new connection. Thus, a *legal* multicast tree with anonymity concerns must satisfy the following three restrictions:[5]

- Cannot contain any node which none of the existing routes passes through (for example, the multicast tree shown in Fig. 1b is *not* legal because all routes established during join processes, drawn by dotted lines, do not pass through node N_1)
- Cannot contain any non-member node as a branch node (for example, node N_2 in Fig. 1b)
- Cannot contain any sink that has grandchildren (for example, the multicast tree in Fig. 1b is illegal because node D_5 has two grandchildren D_8 and D_9)

In other words, packets are either sent through the routes established during join processes (*i.e.,* the blue solid lines in Fig. 1c) or broadcasted by the source and/or sinks in a one-hop manner after the source and/or sinks have received the packets (for example, red solid lines in Fig. 1c). In the latter case, the multicast algorithm decides the transmission power to relay the packets. Such power information is attached to packets in an encrypted form; only the intended sink can decrypt it and relay the packets accordingly.

While complying with the above three restrictions, EEAMA constructs an *energy-efficient* multicast tree. As detailed later in Section 5, the energy-efficient multicast tree is composed of *i*) *unicast* routes that have been established during the join processes and *ii*) *multicast* links that originate from group members and relay received packets in a one-hop manner. As illustrated in Fig. 1c, such unicasts are illustrated in blue thin lines, while such multicasts are represented in red thick lines. Note that since EEAMA relies only on nodes and routes associated with join processes of the underlying anonymous unicast protocol, the anonymity preservation is guaranteed.



(a) The (unicast) routes established during join processes are drawn by dotted lines. (In Fig. 1b and Fig. 1c, these dotted lines still exist but may be covered by solid lines.)

(b) An *illegal* multicast tree with anonymity concerns is drawn by solid lines.

(c) A *legal* multicast tree with anonymity concerns is drawn by (blue and red) solid lines.

Fig. 1. Example of legal and illegal multicast trees with anonymity concerns. The group consists of 12 member nodes: 1 source (node *S*) and 11 sinks (D_1, D_2, …, D_{11}). Other nodes are non-members. Red solid lines show that multiple receivers receive packets within one transmission. Blue lines represent unicast links. Note that non-members (drawn as white circles) are hidden from member nodes (grey circles).

3.3 Multicast tree maintenance

Maintaining such a multicast tree is extremely simple. First, its unicast routes (*i.e.,* blue lines in Fig. 1c) are maintained by the underlying anonymous unicast protocol; no further care from EEAMA is required. Second, maintaining the multicast links (*i.e.,* red lines in Fig. 1c) is not necessary at all; such links are actually broadcasts

[4] The node density can be given or estimated by monitoring the packets traversing the network. The estimation does not need to be very accurate. As shown in Section 6, EEAMA is robust to variations in node density (*e.g.,* ±25% differences from the estimated value).

[5] To better understand the following three restrictions, readers are suggested to compare Fig. 1a, Fig. 1b and Fig. 1c.

determined *dynamically* by EEAMA each time the source attempts to send packets to sinks.

Note that the multicast tree is uni-directional: The source sends packets to sinks through the multicast tree but, sinks send acknowledgements separately (back to the source) via their own unicast routes (dotted lines in Fig. 1a). Retransmission is triggered if the source node does not receive all intended acknowledgements from sinks. In this case, EEAMA resends packets to the sinks associated with missing acknowledgements through the corresponding unicast routes.

4. Analysis of path energy

Since anonymity requirements make intermediate nodes hidden from the group members, the actual route details are unavailable. As a consequence, it is critical for EEAMA to predict important route details, especially the path energy, so that energy efficiency can be improved. To this end, we present next our derived lower and upper bounds for the *expected* path energy (denoted by $\overline{E}$), which is averaged over the entire sample space. More precisely, we consider an ad-hoc wireless network where nodes are deployed randomly according to a planar Poisson distribution with an average density[6] ρ (as in [12]) and then take the average over all possible paths within a forwarding area.

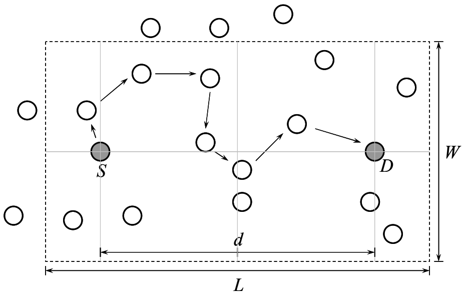


Fig. 2. An example of a rectangular forwarding area of length *L* and width *W*. Packets are forwarded from node *S* to node *D* within the forwarding area. Any node outside the forwarding area simply drops the received packets. The arrows depict the minimum energy path from source to destination.

Similar to [13] for a communication session, we restrict packets to be routed within a *forwarding area* (rather than the entire network). The packets within the forwarding area can be routed *arbitrarily*, but any node outside the forwarding area simply discards the received packets. The forwarding area is a rectangular region of length L and width W, encompassing the source and destination nodes as shown in Fig. 2. L is set to a number greater than d. Note that our analysis is *not* limited to any specific routing protocol with a rectangular forwarding area. We present next formulae for the lower and upper bounds for the path energy. (Due to page limitation, the proof is omitted.)

Theorem 1: Given the average node density ρ, a source-destination pair at distance d, and a forwarding area of length $L \geq d$ and width W, the expected minimum path energy between the source-destination pair ($\overline{E}$) is greater than the following lower bound:

$$\overline{E} \geq d^{\alpha} \sum_{n=0}^{\infty} \frac{e^{-\lambda} \lambda^{n} (n+1)}{(\alpha+1)(\alpha+2)\cdots(\alpha+n)}$$

where $\lambda = \rho dW$ and $\alpha > 1$ is the propagation loss exponent.

If the propagation loss exponent α is an integer greater than 1, then the above formula can be simplified to:

$$\overline{E} \geq d^{\alpha} \left[\frac{\alpha!}{\lambda^{\alpha}} (\lambda - \alpha + 1) - \frac{\alpha! e^{-\lambda}}{\lambda^{\alpha}} \left((\lambda - \alpha + 1) \sum_{i=0}^{\alpha-2} \frac{\lambda^{i}}{i!} - \frac{\lambda^{\alpha-1}}{(\alpha-2)!} \right) \right]$$

The lower bound derived in Theorem 1 can be applied to all practical cases. This is because α can be any real number greater than 1, while a typical propagation loss exponent ranges between 2 and 4. Note that this lower bound is computationally light. It has few terms when α is an integer. Even when α is not an integer, the lower bound converges *quickly* because the number of nodes in the forwarding area is usually not large.

Theorem 2: Given an average node density ρ, a source-destination pair at distance d, and the forwarding area of length L and width W, we have:

$$\overline{E} \leq 2^{\alpha/2-1} \left(\left[\mathrm{LB} + 2 \left(\frac{L-d}{2} \right)^{\alpha} \right] + \left[\frac{2(\rho LW - 1)}{(\alpha+1)(\alpha+2)} + \frac{2}{\alpha+1} \right] W^{\alpha} \right)$$

where α is the propagation loss exponent and **LB** is the lower bound given in Theorem 1 with $\lambda = \rho LW$ (rather than $\lambda = \rho dW$).

Note that the bounds above are normalized; that is, the communication energy consumed on transmitting a unit-size packet over a unit distance is normalized to 1. The actual energy can be reversely de-normalized.

The following section describes in detail EEAMA that makes energy-efficient routing decisions based on the above lower and upper bounds.

5. Newly proposed algorithm

In this section, we describe the *E*nergy-*E*fficient *A*nonymous *M*ulticast Algorithm (EEAMA). For anonymity reasons, only three types of information are provided to EEAMA, namely, the average node density in the network, the location of group members, and the routes from the source node to sinks established during the join processes. (However, the route details such as intermediate

[6] The node density ρ can be given or estimated by monitoring the headers of packets traversing the network. The estimation does not need to be very accurate. As shown in Section 6.3, EEAMA performs well for almost all network instances, even if the actual node density is quite different from the estimated value.

nodes, hop count, link lengths and path energy values are unavailable due to anonymity requirements.)

Notation

R: the remaining set
R_i, Q_i: the i^{th} element of the sets R_i and Q_i, respectively
$E_L(u, v)$: the link energy between node u and node v
H: the index of the transfer hub (*i.e.,* R_H is the transfer hub)
$M(i)$: the set of selected multicast receivers assuming $H = i$
$E_M(i)$: the multicast energy assuming $H = i$
$E_U(i)$: the total unicast energy assuming $H = i$
$\mathbf{LB}(u, v)$, $\mathbf{UB}(u, v)$: the lower and upper bounds (given in theorems 1 and 2) for the path energy between node u and node v
w: the assigned weight

Pseudo code

```
1   $R$ := the set of all the group members including the source node $S$
2   While $R$ is not empty
3       cost := $\infty$
4       For $i$ := 1 to $|R|$
5           $Q$ := the ordered sequence after sorting $R$ in order of the
              distance from node $R_i$
6           $M(i) := Q$
7           $E_M(i) := E_L(R_i, Q_{|Q|})$
8           $E_U(i) := 0$
9           $E_{total} := E_M(i)$
10          For $j$ := $|R|$-1 to 1 by -1
11              $E_U(i) := E_U(i) + w\,\mathbf{LB}(R_i, Q_j) + (1-w)\,\mathbf{UB}(R_i, Q_j)$
12              If $E_L(R_i, Q_j) + E_U(i) \le E_{total}$
13                  $M(i) := \{Q_1, Q_2, \ldots, Q_j\}$
14                  $E_M(i) := E_L(R_i, Q_j)$
15                  $E_{total} := E_M(i)+E_U(i)$
16              End If
17          End For
18          If $E_M(i) / |M(i)|$ < cost
19              cost := $E_M(i) / |M(i)|$
20              $H := i$
21          End If
22      End For
23      If $M(i)$ is not empty
24          Node $S$ unicasts packet(s) via the given route to node $R_H$.
25          Node $H$ multicasts the received packet(s) once, at energy
              $E_M(H)$, to all the nodes in $M(H)$.
26          $R := R \setminus \{M(H), R_H\}$
27      Else
28          Node S unicasts packets individually via the given routes to
              all the nodes in $R$.
29          $R := \phi$
30      End If
31  End While
```

Fig. 3. The pseudo code of EEAMA. For clarity, $M(i)$, $E_M(i)$ and $E_U(i)$ are indexed by *i*. However, such indexing is not necessary in practical implementations; $M(i)$, $E_M(i)$ and $E_U(i)$ can be replaced by un-indexed variables (*i.e., M*, E_M and E_U) to save memory usage.

5.1 Description of EEAMA

EEAMA makes decisions based on the predicted path energy values. Given a source-destination pair, the predicted path energy value is the *weighted average* of the lower bound in Theorem 1, **LB**, and the upper bound in Theorem 2, **UB,** namely, the predicted path energy value is $w\mathbf{LB} + (1 - w)\mathbf{UB}$, where $w \in [0, 1]$ is the assigned weight.

EEAMA takes a variable number of iterations before it completes. Initially, the *remaining set* contains all the g group members (including the source node). After one iteration, EEAMA determines a subset of group members (called *receivers* for simplicity) and the precise ways of sending packets to these receivers. Then these receivers are removed from the remaining set. EEAMA continues the next iteration until the remaining set becomes empty. Since each iteration removes at least one receiver from the remaining set, the number of iterations EEAMA actually takes is bounded by g.

During each iteration (*i.e.,* lines 3-30 in Fig. 3), EEAMA considers any node in the remaining set to be the *transfer hub*. For each possible transfer hub, EEAMA determines the set of receivers to which multicasting packets in a one-time, single-hop transmission from the transfer hub is more energy-efficient than unicasting packets individually. Then, among all possible transfer hubs in this iteration, EEAMA chooses the *best transfer hub* (denoted by H in Fig. 3) which minimizes the multicast energy per receiver (denoted by *cost* in Fig. 3).

If the set of receivers associated with the best transfer hub is empty (*i.e.,* condition in line 23 in Fig. 3 is false since multicasting cannot save energy), then EEAMA completes after unicasting packets *individually* to all nodes in the remaining set from the source node (lines 28-29 in Fig. 3). Otherwise, the source sends packets to the best transfer hub which then forwards the received packets in a one-hop multicast manner to all the corresponding receivers (lines 24-25 in Fig. 3). After that, the best transfer hub and all its receivers are removed from the remaining set (line 26 in Fig. 3); this completes one iteration. EEAMA starts the next iteration if the remaining set is not empty.

5.2 Complexity analysis of EEAMA

The pseudo code of EEAMA is presented in Fig. 3. The *worst-case* complexity can be analyzed as follows. An iteration corresponds to an execution *inside* the while loop (*i.e.,* lines 3-30) of Fig. 3. Initially, $|R| = g$ where g is the group size. Since each iteration decreases the size of R by at least one, the number of iterations cannot exceed g. The algorithm has two for loops. Because the execution time of each for loop depends on $|R|$, the worst-case complexity of an iteration is $O(g \cdot g) = O(g^2)$.

Note that the sorting in line 5 does *not* increase the complexity because advanced sorting algorithms have a complexity of $O(g \log g)$; even the simplest sorting algorithms like bubble sort would result in a complexity of only $O(g^2)$. As such, the worst-case complexity of EEAMA is $O(g \cdot g^2) = O(g^3)$.

6. Evaluation of the algorithm

While Section 3.2 explains why EEAMA preserves anonymity, the goal of this section is to evaluate the performance of EEAMA by simulation. As such, we present performance improvements due to EEAMA, compared to the cases which solely run an anonymous *unicast* protocol. The performance metrics of interest are energy consumption, packet delivery ratio and end-to-end delay.

6.1 Simulation setup

The simulation setup is as follows. The network covers an area of 1000m×1000m. Unless otherwise specified, 100 nodes are placed randomly over the network region. Nodes are either static (sections 6.2 and 6.3) or mobile at a variable speed of 0 to 10m/s (Section 6.4). Data packets of size 512 bytes are injected into each short-lived group at a rate of 4 packets per second. All multicast groups live for 1 minute. After the 1-minute lifetime is up, a new multicast group is created to replace the old group. Unless otherwise specified, each multicast group consists of 1 source and 9 sinks. Each data packet is multicasted to group members from the group leader (*i.e.,* the source node) using EEAMA. The data rate in a wireless channel is 1 Mb/sec. We consider that ASC is the underlying anonymous unicast protocol and AES is the default encryption algorithm. The execution time of encrypting and decrypting an AES block is taken from [1].

Instead of customizing the transmission ranges (which would improve performance drastically), the transmission ranges are fixed to several levels, namely, 30.48m, 91.44m, 365.76m, and 6437.376m. The first three values are typical for 802.11b/g transmission range in offices, 802.11b/g range outdoors, and 802.11a range outdoors, respectively, while the last value is the typical 802.16 transmission range. The propagation loss exponent is set to 2.

Similar to [5], two *media access control* mechanisms are used by this routing protocol to improve the packet delivery ratio, namely, IEEE 802.11 DCF and CSMA/CA. Packets are sent using the IEEE 802.11 DCF if packets are sent through routes maintained by the underlying anonymous unicast protocol. Otherwise, packets are broadcasted using CSMA/CA.

A *retransmission* mechanism with a predefined timer is also implemented. If a source node cannot receive acknowledgements sent from all sinks within a predefined interval (*i.e.,* one second in this simulation setup), the source node will resend the packet to the sink(s) associated with missing acknowledge(s) through the routes maintained by the underlying anonymous unicast protocol. In our simulations, the maximum number of end-to-end retransmissions is set to 4. After 4 retransmissions, the sink is assumed to *leave* the group. Therefore, the source stops further packet transmissions to that sink.

6.2 Main simulation results

We first present the energy savings due to EEAMA, in the form of "*normalized* energy per packet". The energy per packet is defined as the total energy consumption over the entire network divided by the total number of packets that reach sinks if no packet is dropped or corrupted. We *normalize* to 1 the energy per packet value when there is exactly one group.

Fig. 4 shows that *EEAMA* consistently achieves significant energy savings, compared to the underlying anonymous *unicast* protocol. This is because, based on prediction of path energy values, EEAMA is able to route packets more efficiently by exploiting both wireless multicast advantage and multihop routing.

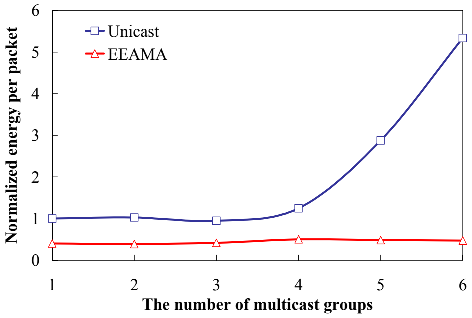


Fig. 4. This figure shows the normalized energy per packet when the number of multicast groups varies. When there are few groups, the network is not congested and therefore, the normalized energy per packet value is fairly constant. Without EEAMA, once the number of multicast groups goes beyond 4, a large number of (hop-by-hop and end-to-end) packet retransmissions take place, which increases the energy consumption per successful transmission.

Moreover, since packets are transmitted in an efficient way, EEAMA can support more multicast groups without causing network congestion. As shown in Fig. 4, the normalized energy per packet value of EEAMA is fairly *constant*; while the normalized energy per packet value of the unicast protocol increases *sharply* beyond 4 groups (due to network congestion and also because a large number of retransmissions is taking place).

From the perspective of packet delivery ratio, the same phenomenon can be observed in Fig. 5. We compute *packet delivery ratio* as the ratio of the total number of packets received by destination nodes to the sum of the number of packets transmitted times the number of intended destination nodes. The number of intended destination nodes is 1 during the join process and it is g - 1 during packet multicasting, where g is the group size.

We observe in Fig. 5 that the packet delivery ratio of EEAMA decreases slowly as the number of multicast groups increases; while the packet delivery ratio of the unicast protocol drops quickly beyond 4 multicast groups. This is because without EEAMA, the network becomes severely congested beyond 4 groups. On the contrary, EEAMA can multicast packets efficiently, thus lowering network congestion and sustaining packet delivery ratio.

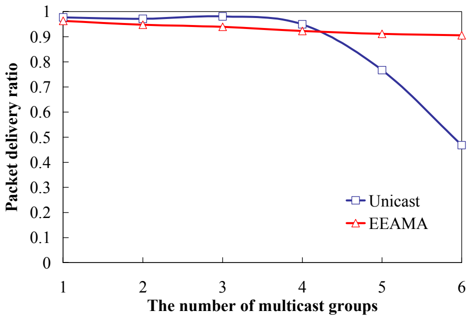


Fig. 5. In a 100-node network where each multicast group has 10 member nodes, as the number of multicast groups increases, the network becomes more congested and the packet delivery ratio thus decreases. However, compared to the anonymous unicast protocol, EEAMA is able to lower network congestion due to its capability of efficient multicasting. This advantage is particularly obvious beyond 4 multicast groups.

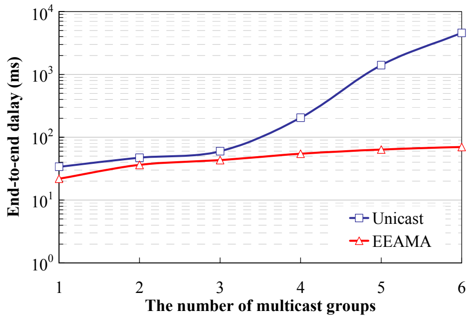


Fig. 6. As shown, the end-to-end delay of EEAMA is always smaller than the end-to-end delay of the anonymous unicast protocol. Beyond 4 multicast groups, the difference becomes significant. This shows that compared to the anonymous unicast protocol, EEAMA can support more active multicast groups without causing network congestion (or an unacceptably long delay for real-time traffic).

Fig. 6 shows another advantage of our proposed algorithm. In terms of end-to-end delay, EEAMA beats the unicast protocol over the entire simulation range. This is because of EEAMA's capability of reducing network congestion and the numbers of packets queued at network nodes. Moreover, because of the small end-to-end delay, EEAMA can support more groups to multicast their *real-time traffic* in a timely fashion.

6.3 EEAMA applicability with inaccurate node density

In reality, it is possible that the exact number of nodes in the network may be unavailable and the estimation of node density may be thus inaccurate. The goal of this section is to investigate the impact on the performance of EEAMA due to such inaccurate information.

To this end, instead of using the default setup where the network consists of a fixed number of nodes (*i.e.,* 100 nodes), a random number of nodes (according to a Poisson distribution with an average of 100) are placed over the network. EEAMA *does not* know the actual node density; instead, a fake node density of $10^{-4}/m^2$ is fed into EEAMA.

We generate 100 network instances and simulate their individual energy consumptions. As shown in Fig. 7, although the node density is not accurate, EEAMA performs better than the unicast protocol in all instances.

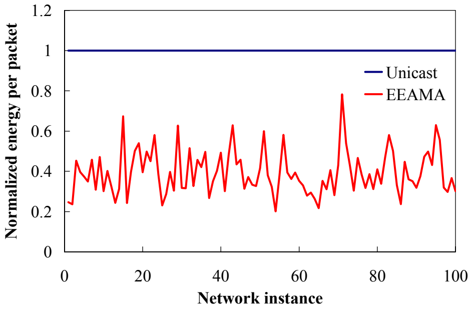


Fig. 7. The normalized energy per packet for 100-node network instances with one 10-member group. The figure shows that although EEAMA *does not* know the accurate node density, it still outperforms the unicast protocol in all instances.

6.4 EEAMA applicability to mobile environments

The goal of this section is to study the applicability of EEAMA to *mobile* networks. In particular, the most important metric reflecting the applicability of EEAMA in mobile ad hoc networks is packet delivery ratio.

Nodes mobility is modeled as a random waypoint process, which is often used in ad hoc network simulations. The node mobility speed varies between 0 to 10m/s. The pause time is fixed to 30 seconds. Results are averaged over multiple runs with different seeds for the random number generator.

As the node mobility speed varies from 0 to 10m/s, Fig. 8 shows the packet delivery ratio of EEAMA. This figure confirms the applicability of EEAMA in such mobile networks because the packet delivery ratio is high and increasing node mobility speed does not affect the packet delivery ratio significantly.

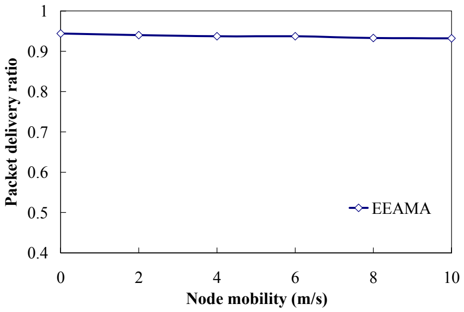


Fig. 8. The packet delivery ratio for 100-node network with one 25-node group. As the speed of mobile nodes increases, the packet delivery ratio under EEAMA decreases slowly. This can be attributed to EEAMA's small end-to-end latency (as shown in Fig. 6).

The simulation results can be summarized as follows:

- EEAMA performs better than the anonymous unicast protocol in terms of energy efficiency, packet delivery ratio, and end-to-end delay.
- EEAMA works well even when the information about node density is of limited accuracy.
- In MANETs where nodes move at a reasonable speed (*e.g.,* 0 to 10m/s), the packet delivery ratio of EEAMA does not degrade significantly as the node speed increases.

7. Conclusion

In this paper, we have addressed the issue of energy-efficient anonymous multicast and have proposed an Energy-Efficient Anonymous Multicast Algorithm (EEAMA) to solve this important issue.

EEAMA has three unique features. First, it does not rely on route details which are not easily available in anonymous environments. Instead, it predicts path energy by exploiting statistical properties and our derived bounds. Second, EEAMA is fast and lightweight. Indeed, its complexity depends on multicast group size, *not* network size. Third, EEAMA is dynamic which makes it a perfect solution for mobile networks. Indeed, *each time* a source node attempts to multicast packets to its sink nodes, EEAMA constructs an energy-efficient multicast tree. These three features make EEAMA an excellent solution to energy-efficient anonymous multicast in MANETs.

Our simulation results show that compared to the sole use of an anonymous unicast protocol, EEAMA achieves significant performance improvements in terms of energy efficiency, packet delivery ratio and end-to-end delay. Simulation results also show that EEAMA allows higher injection rates of multicast traffic before causing network congested.

Future work may include evaluation of EEAMA when running on top of other anonymous unicast protocols, as well as its worst-case analysis (for example, using the lower bound as the predicted path energy value instead a weighted sum of lower and upper bounds).

References

[1] J. Kong and X. Hong, "ANODR: Anonymous on demand routing protocol with untraceable routes for mobile ad-hoc networks," in Proc. ACM MobiHoc, Annapolis, June 2003.

[2] K. El-Khatib, L. Korba, R. Song, and G. Yee, "Secure dynamic distributed routing algorithm for ad hoc wireless networks," in Proc. ICPP Workshops, Kaohsiung, Taiwan, Oct. 2003.

[3] B. Zhu, Z. Wan, M. S. Kankanhalli, F. Bao, and R. H. Deng, "Anonymous secure routing in mobile ad-hoc networks," in Proc. IEEE LCN, Tampa, Nov. 2004.

[4] S. Seys and B. Preneel, "ARM: Anonymous Routing Protocol for Mobile Ad hoc Networks," in Proc. IEEE AINA, Vienna, Austria, Apr. 2006.

[5] J.-C. Kao and R. Marculescu, "Real-Time Anonymous Routing for Mobile Ad Hoc Networks," in Proc. IEEE WCNC, Hong Kong, Mar. 2007.

[6] ITU-T Recommendation G.114, "One-way transmission time," Mar. 1993.

[7] C. Grosch, "Framework for Anonymity in IP-Multicast Environment," in Proc. IEEE GLOBECOM, San Francisco, Nov. 2000.

[8] N. Weiler, "Secure Anonymous Group Infrastructure for Common and Future Internet Application," in Proc. ACSAC, New Orleans, Dec. 2001.

[9] Li Xiao, X. Liu, W. Gu, D. Xuan, and Y. Liu, "A Design of Overlay Anonymous Multicast Protocol," in Proc. IEEE IPDPS, Rhodes Island, Greece, Apr., 2006.

[10] J. E. Wieselthier, G. D. Nguyen and A. Ephremides, "On the Construction of Energy-Efficient Broadcast and Multicast Trees in Wireless Networks," in Proc. IEEE INFOCOM, Tel-Aviv, Israel, Mar. 2000.

[11] S. Guo and O. W. Yang, "Minimum-Energy Multicast Routing in Static Wireless Ad Hoc Networks," in Proc. IEEE VTC, Los Angeles, Sep. 2004.

[12] P. Hall, Introduction to the Theory of Coverage Processes. John Wiley & Sons Inc., 1988.

[13] Y. Ko and N. H. Vaidya, "Geocasting in Mobile Ad Hoc Networks: Location-Based Multicast Algorithms," in Proc. IEEE WMCSA, New Orleans, Feb. 1999.

[14] J.-C. Kao and R. Marculescu, "Minimizing Eavesdropping Risk by Transmission Power Control in Multihop Wireless Networks," IEEE Trans. on Computers, Vol.56, No.8, pp. 1009-1023, Aug. 2007.