Topology control adopting optimal topology over update interval in mobile ad hoc networks

Kouichi Genda
Department of Computer Science, College of Engineering, Nihon University
1 Nakagawara, Tokusada, Tamura, Koriyama-shi, Fukushima 963-8642, Japan
gen@cs.ce.nihon-u.ac.jp

Abstract: A topology control method for mobile ad hoc networks is proposed, in which the optimal topology of each topology update interval is adopted by considering both the dynamic positions of one-hop neighboring nodes and cumulative energy consumption to send data over the update interval. This approach of adopting the topology optimized over the update interval differs from an existing approach in which the topology update frequency is increased. Through performance simulation, conditions under which the proposed method is applicable are revealed by comparing the energy consumption of the proposed method to that of the existing method. Additionally, the proposed method is practical because it takes only several tens of milliseconds to determine the optimal topology even when using a CPU widely used for embedded devices.

Keywords: Mobile ad hoc network, topology control, optimization, energy consumption, connectivity

Classification: Wireless Communication Technologies

References

[1] Y. Kawamoto, H. Nishiyama, N. Kato, "MA-LTRT: A novel method to improve network connectivity and power consumption in mobile ad-hoc based cyber-physical systems", IEEE Trans. Emerging Top. Comput., vol. 1, no. 2, pp. 366-374, Aug. 2013.

[2] J. Wu, F. Dai, "Mobility-sensitive topology control in mobile ad hoc networks", IEEE Trans. Parallel Distrib. Syst., vol. 17, no. 6, pp. 522-535, May 2006.

[3] N. Li, J. Hou, C. Sha, L. Sha, "Design and analysis of an MST-based topology control algorithm", IEEE Trans. Wireless Commun., vol. 4, no. 3, pp. 1195-1206, May 2005.

[4] N. Li, J. C. Hou, "Localized fault-tolerant topology control in wireless ad hoc networks", IEEE Trans. Parallel Distrib. Syst., vol. 17, no. 4, pp. 307-320, Mar. 2006.

[5] K. Miyao, H. Nakayama, N. Ansari, N. Kato, "LTRT: An efficient and reliable topology control algorithm for ad-hoc networks", IEEE Trans. Wireless Commun., vol. 8, no. 12, pp. 6050-6058, Dec. 2009.
1 Introduction

In mobile ad hoc networks (MANETs) constructed among various kinds of mobile devices, such as smartphones, robots, drones, and vehicles, topology control is essential to reduce energy consumption while preserving network connectivity. This control is achieved by controlling the transmission radius and the next hop device of each mobile device (node) for every topology update interval.

When each node moves with a different speed and direction, the node positions change dynamically until the next topology update. This implies that the appropriate topology based on the initial positions of the nodes at the current update time may become completely different from the topology based on the positions of the nodes immediately before the next update time. Consequently, to maintain network connectivity, a large transmission radius will be required and, as a result, the energy consumption for data transmission will increase.

To rectify this situation, an existing approach [1, 2] increases the topology update frequency by shortening the update interval. This method reduces the transmission radius required for data transmission, but frequent updates increase the cost of topology reconstruction because each node frequently transmits a hello message with the maximum transmission radius, resulting in high energy consumption. A method to set the topology reconstruction interval dynamically has been proposed to reduce energy consumption [1]. A different approach is to adopt the topology optimized over the update interval using information about the dynamic positions of one-hop neighbors. This method is expected to prevent increases in the number of topology reconstructions. However, a major challenge is determination of the optimal topology over the update interval.

Here, a topology control method based on the latter approach is proposed. In the proposed method, the optimal topology is determined by considering both the dynamic positions of one-hop neighboring nodes and cumulative energy consumption to send data over the update interval.

2 Proposed method

2.1 Realization of proposed method

In the proposed method, the optimal topology of each update interval is adopted to reduce energy consumption. To determine the optimal topology, we focus on the cumulative energy consumption, which is defined as the total energy consumption to send data during the update interval when adopting a topology. The topology with the minimum cumulative energy consumption is determined as the optimal topology. In contrast, in existing methods [1, 2], the topology of each interval is determined using only the initial positions of the one-hop neighbors at update time.

The optimal topology is constructed in three phases: a) grouped node configuration, b) optimal topology construction, and c) transmission radius control.

a) Grouped node configuration
Each node configures its grouped nodes among one-hop neighbors that remain within the distance of the maximum transmission radius, \( R_{\text{max}} \), over the update interval, and constructs its topology among the grouped nodes.

b) Optimal topology construction

The optimal topology with the minimum cumulative energy consumption is approximately determined using two steps: first, determine the candidates for the optimal topology (called topology candidates); and second, determine the optimal topology. These two steps allow the computation of the optimal topology as simply as possible and can be adopted for various types of mobile devices with high to low processing abilities.

During the first step, topology candidates are calculated by considering the dynamic positions of nodes in the update interval. The simplest method to determine topology candidates is to calculate \( m \) topologies using the predicted positions of grouped nodes at time \( iT/m \), where \( T \) is the update interval, \( m \) is the number of divisions of \( T \), and \( i \) is an integer from zero to \( (m - 1) \). Various types of useful topology algorithms, such as those described in Refs. [3-5], can be applied to calculate a topology candidate.

During the second step, to determine the optimal topology, the cumulative energy consumption is calculated for each topology candidate. The optimal topology with the minimum cumulative energy consumption can be determined using Eqs. (1) and (2) under the worst condition where data are continuously generated or relayed during the update interval in each node. As the metric of energy consumption of a topology, here we apply a simple formula, the sum of the square of the distance of the link in a topology, to ascertain if our approach is effective.

Minimize\[
Ce_i = \int_0^T \sum_{k=0}^{E_i-1} l_{ik}(t)^2 dt ,
\]
subject to\[
l_{ik}(t) \leq r_k , \quad 0 \leq t \leq T ,
\]
where \( Ce_i \) is the cumulative energy consumption of each node over update interval \( T \) when applying the topology candidate calculated using the predicted positions of grouped nodes at the \( i \)th time \((0 \leq i < m)\); this candidate is called the \( i \)th topology. \( E_i \) is the number of links in the \( i \)th topology. \( l_{ik}(t) \) is the distance of the \( k \)th link \((0 \leq k < E_i)\) in the \( i \)th topology at time \( t \) \((0 \leq t \leq T)\). \( r_k \) is the maximum distance required to maintain the connectivity of \( l_{ik} \). The \( i \)th topology with the minimum \( Ce_i \) is determined as the optimal topology of the update interval.

c) Transmission radius control

Based on the optimal topology in the update interval, the transmission radius of the update interval is controlled. The distances of the links between each node and its grouped nodes are calculated from the beginning to the end of the update interval. The maximum distance is set as the transmission radius \( R_{\text{opt}} \) of the update interval.
2.2 Condition under which proposed method is applicable

Here, we discuss conditions under which the proposed method is applicable by comparing the energy consumption of the proposed method to that of an existing method that increases topology update frequency. We assume that, in the existing method, topology is updated \( n \) times during the update interval of the proposed method \( T \). In the proposed method, the optimal topology is determined among \( m \) topology candidates every update interval.

The energy consumed by each node consists of the energy used to broadcast a hello message for every update interval (\( P_{\text{hello}} \)); energy used to send data during the update interval (\( P_{\text{data}} \)); and energy used to determine the topology of update interval (\( P_{\text{cal}} \)). Here, we compare the energy consumed performing \( P_{\text{hello}} \) and \( P_{\text{data}} \) under the condition of \( m = n \). When \( m = n \), both methods share the same number of computing topologies. Thus, the time complexity of the proposed method, which is proportional to \( P_{\text{cal}} \), is almost equal to that of the existing method.

Based on the above discussion, energy consumption values of the proposed method (\( P_p \)) and that of the existing method (\( P_b \)) over \( T \) are formulated respectively in Eqs. (3) and (4):

\[
P_p = p_d \cdot d_{\text{hello}} + p_d \cdot d_{\text{data}} \cdot \left( \frac{R_{\text{opt}}}{R_{\text{max}}} \right)^4,
\]

\[
P_b = n \cdot p_d \cdot d_{\text{hello}} + p_d \cdot \frac{d_{\text{data}}}{n} \sum_{i=0}^{n-1} \left( \frac{R_{b,\text{opt}}(i)}{R_{\text{max}}} \right)^4.
\]

Here, we compare \( P_p \) and \( P_b \) under the condition of \( m = n \).

The condition that results in less energy consumption in the proposed method than in the existing method is described in Eq. (5):

\[
1 + \frac{d_{\text{data}}}{d_{\text{hello}}} \left( \frac{R_{\text{opt}}}{R_{\text{max}}} \right)^4 < 1.
\]

Because \( \text{ave} R_{b,\text{opt}} \) is equal to or less than \( R_{\text{opt}} \), the applicability of the proposed method depends on the values of \( n \) and \( d_{\text{data}}/d_{\text{hello}} \).

3 Performance evaluation

We evaluate the conditions under which the proposed method is applicable compared to that of an existing method. In addition, we evaluate computation time
of our method for practical usage. Although the proposed method can be applied to various types of topology algorithms, here, we apply it to the local minimum spanning tree (LMST) [4] because of its simplicity. We evaluated the existing method with the LMST topology determined using the initial positions of one-hop neighbors at the update time with a shorter interval $n$ times. Simulations were performed using our custom-made C simulator. 20–80 nodes were initially placed randomly in a $300 \times 300$ area at the center of the $500 \times 500$ simulation area. The maximum transmission radius $R_{\text{max}}$ is 200. Each node moved rectilinearly and the speed and direction of each node were set randomly, at the maximum moving speed of 20–60 per $T$.

Figure 1 shows samples of the topologies constructed by the proposed and existing methods at the update time for 50 nodes at maximum moving speed of 60 per $T$.

![Fig. 1. Samples of topologies constructed by the proposed and existing methods at the update time for 50 nodes at maximum moving speed of 60 per $T$.](image)

In our method, some links exist between two nodes although the distance between them is not minimum. In the existing method, each node is connected to other nodes through the minimum distance path. This is because an appropriate topology over the update interval is adopted in our method. Note that we confirmed that network connectivity is preserved over all the nodes under all the evaluated conditions.

Figure 2 shows an applicability condition of the proposed method, that is sample results of the relationship between $n$ and $P_p/P_b$ for 50 nodes under different values of $d_{\text{data}}/d_{\text{hello}}$. We here define $P_p/P_b$ as the metric of applicable conditions in terms of energy consumption, where the proposed method is more applicable when $P_p/P_b$ is less than 1. The results indicate that the applicability of the proposed method is determined by the balance between the values of $d_{\text{data}}/d_{\text{hello}}$ and $n$, and that the proposed method is more effective when $d_{\text{data}}/d_{\text{hello}}$ is smaller and $n$ is larger. Under this evaluation condition, when $d_{\text{data}}/d_{\text{hello}}$ is less than 400, the proposed method is more effective, and the applicability condition is more extended when $n$ is larger. While the existing method is more effective regardless of the value of $n$ at the $d_{\text{data}}/d_{\text{hello}}$ values of over 400.

Figure 3 shows the average computation time of the proposed method with $m=3$
to determine the optimal topology. Because various kinds of nodes with high to low processing abilities should be considered for practical MANETs, we evaluate the computation time using Raspberry Pi 3 model B widely used for embedded devices. The results show that although the computation time required for our method is approximately twice that required for the existing method, a computation time of several tens of milliseconds is sufficient to determine the optimal topology. Hence, our method is practical under the simulation condition.

For practical usage, we must determine the applicability of the proposed method by considering a balance between the increase in computation time and reduction in energy consumption.

4 Conclusion

A topology control method for MANETs is proposed, in which the topology of each update interval is optimized by considering both the dynamic positions of one-hop neighbors and the cumulative energy consumption over the update interval. The simulation results indicate that the proposed method is more effective when the data size of a hello message is larger (or the amount of data sent during the update interval is smaller) and the number of nodes increases. In addition, the proposed method is fast enough to determine the optimal topology even when using a CPU widely used for embedded devices.

Fig. 2. Applicable conditions of the proposed method for 50 nodes at maximum moving speed of 60 per T under different values of $\frac{d_{\text{data}}}{d_{\text{hello}}}$.

Fig. 3. Computation time to determine the optimal topology of update interval.