A Lightweight Ad Hoc Network Architecture and Routing Protocol Design

Shaopeng Sun[1](✉) and Kuo Liu[2]

[1] The 54th Research Institute of China Electronics Technology Group Corporation, Shijiazhuang 050081, Hebei, China
sunspmail@139.com
[2] Hebei General Hospital, Shijiazhuang 050057, Hebei, China

Abstract. Ad hoc network is a sort of self-organizing network that relies on no fixed infrastructure. Each node can act as a host to send and receive its own data, or as a possible router to forward data to other nodes. In the Ad hoc networks, routing algorithms are one of the main factors that constrain network transmission performance. Routing control overhead is an important indicator for evaluating routing protocols. Routing control messages can occupy network resources, and a large number of routing control messages can even lead to network congestion. This article proposes a lightweight network protocol that reduces routing control overhead from multiple aspects. Firstly, we propose a hybrid signaling model. Secondly, we simplify the routing process. Finally, we simplify the control messages.

Keywords: lightweight · signaling model · routing protocol · Ad hoc networks

1 Introduction

The Ad hoc network is a sort of self-organizing network that does not rely on fixed infrastructure [1]. The flexibility of Ad hoc networks has attracted widespread attention in both military and civilian fields [2], so Ad hoc networks have broad application prospects.

Each node in an Ad hoc network can act as a host to send as well as receive its own data, or as a possible router to relay data to other nodes, so routing algorithms are one of the main factors that constrain network transmission performance [3]. Classic routing algorithms include proactive routing algorithms and reactive routing algorithms. The proactive routing algorithm [4], also known as table driven routing algorithm, initiates the establishment of the whole network route after the nodes are deployed. Therefore, end-to-end transmission delays of communication are small, and control message costs increase when the number of nodes increases, but is almost unaffected by the traffic. Reactive routing algorithm [5] is also known as on-demand routing algorithm [6]. Only when a node has communication needs, it initiates the routing lookup process. Therefore, the end-to-end transmission delay of communication is greater than that of proactive routing algorithm. When there are few communication nodes, the control message overhead is

© ICST Institute for Computer Sciences, Social Informatics and Telecommunications Engineering 2025
Published by Springer Nature Switzerland AG 2025. All Rights Reserved
H.-H. Chen and W. Meng (Eds.): WiSATS 2024, LNICST 606, pp. 120–127, 2025.
https://doi.org/10.1007/978-3-031-86203-8_10

low; but as communication nodes increase, the cost of control messages increases, and may even exceed the cost of proactive routing algorithms.

This paper designs a hybrid signaling Ad hoc network architecture, and proposes a lightweight routing protocol.

2 Hybrid Signaling Ad Hoc Network Architecture

The network architecture of this paper is given as below (Fig. 1).

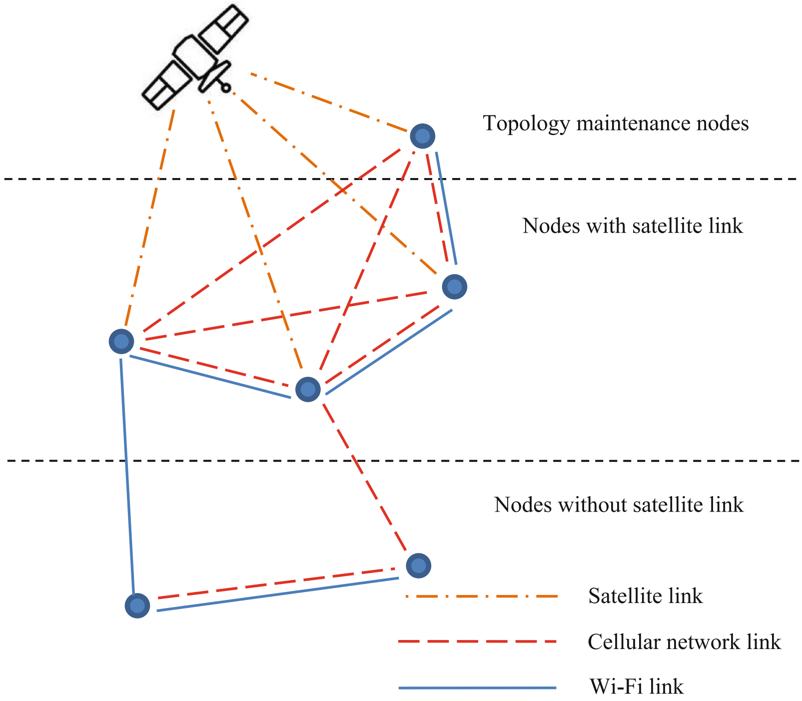

Fig. 1. Network architecture

Each node in this network architecture has three network interfaces, i.e. satellite interface, cellular network interface and Wi-Fi interface. All these nodes form an Ad hoc network. The hybrid signaling model is given as below (Fig. 2).

The advantage of common channel signaling is that the control message is separated from the datagram, and the control message does not occupy the bandwidth of the datagram and does not affect the normal transmission of data. Only when the satellite link of the node fails, the cellular network link or Wi-Fi link is used to transmit control messages through channel associated signalling. At this time, the signaling transmission method is the same as that of traditional routing protocols, but this part of signaling only accounts for a small proportion in this scheme. Therefore, compared with traditional routing protocols, the impact of control messages on user data is minimal.

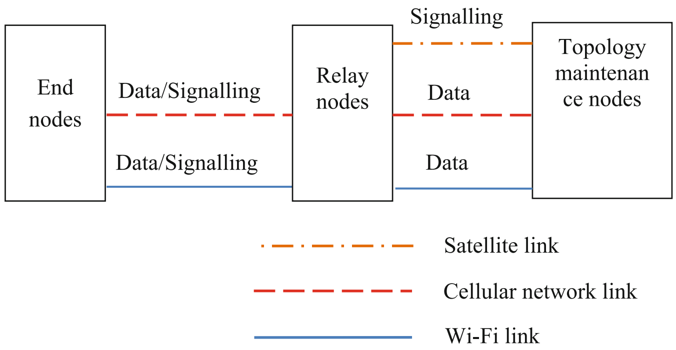

Fig. 2. Hybrid signaling model

3 Simplified Routing Process and Control Messages

In reactive routing protocols, for example, AODV [7], each source node needs to initiate a routing query process when searching for each destination node. In a network of 128 nodes, each routing request message needs to be sent or forwarded 127 times, and when the link fails, the source node needs to restart the routing research process. Each routing re-search process will send or forward 127 routing request messages. It can be seen that in a large-scale network, the reactive routing protocols will bring significant routing control overhead.

In active routing protocols, for example, DSDV [8], as nodes have equal status and do not have a logical center, each node needs to exchange local routing information with the other nodes. In active routing protocols, the cost of control messages hardly increases with the increase of communication node pairs. As the number of communication nodes increases, the number of datagrams in the Ad hoc network increases, and the proportion of routing control messages decreases. Therefore, in largescale networks, active routing protocols have better control message overhead than reactive routing protocols. Therefore, the proposed routing protocol in this paper is based on the active routing protocol and improved.

In the proposed routing protocol in this paper, the topology maintenance node is first selected based on the logical status and security of the nodes. This node is responsible for maintaining the topology of the Ad hoc network. A topology maintenance node should have an available satellite link so that it can report the local topology to the satellite which will can connect the entire network. Each node only needs to upload the local neighbor node table to the topology maintenance node, and there is no need to repeatedly exchange routing tables between nodes, thus saving a lot of control message overhead. After the topology maintenance node generates a network wide routing table, it is then broadcasted down to each node. During each round of routing update, each node only has one upload action and only the topology maintenance node performs one downlink broadcast, thus saving a lot of control message overhead.

To further reduce the control message overhead in the network, nodes with failed satellite links no longer blindly broadcast NT (Neighbor Table) messages. Instead, relay optimization algorithms are used to find the best forwarding relay in the region. Nodes with failed satellite links in the region forward NT messages to the relay node, which integrates NT messages and forwards them uniformly to nodes with valid satellite links, which then forwards NT messages to the topology maintenance node; After receiving the TC (Topology Control) message forwarded by the satellite link from the topology maintenance node, the TC message is then relayed to the node in the region where the satellite link has failed.

When a node determines that the local satellite link has failed, it broadcasts a HELLO message with ReqRelay = 1. After the neighboring node receives this mes-sage, it broadcasts the local NT message to the nearby node. In this area, nodes that receive NT messages and satellite links fail calculate the optimal relay node to try to reduce the control message cost of the relay process. A node with a satellite link failure that is 1 hop away from the effective relay node of the satellite link can directly send NT messages to the relay. Therefore, this paper mainly discusses the relay selection of nodes that are two-hop far away from the effective relay node of satellite link, as shown in Fig. 3.

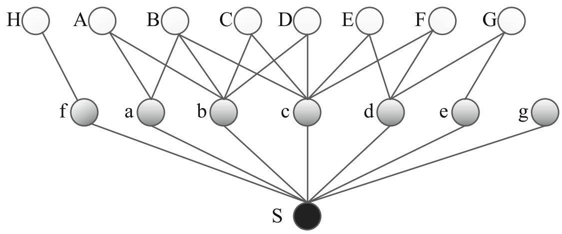

Fig. 3. Relay selection topology

The specific process is as follows:

(1) By calculating the connectional degree between a one-hop neighbour node and its two-hop neighbour(s), it can be obtained that $Neighbour_f$ = H, $Neighbour_a$ = A & B, $Neighbour_b$ = A & B & C & D, $Neighbour_c$ = B & C & D & E & F, $Neighbour_d$ = E & F & G, $Neighbour_e$ = G, $Neighbour_g$ = Null, the one-hop neighbours are sorted from small to large with their two-hop neighbour counts, and nodes with connectivity of 0 are removed. The sorted one hop neighbours are {e, f, a, d, b, c}.

(2) Count the one-hop neighbours connected by two-hop nodes, we have a set $N_2(i)$ = {A, B, C, D, E, F, G, H}, and their connectional degrees are 2, 3, 2, 2, 2, 2, 2, 2, 1, respectively.

(3) If $N_2(i)$ is empty, the optimal relay calculation ends; Otherwise, try to exit the optimal relay calculation for each node in set $N_1(i)$, and the operation starts from the first node on the left, and ends on the first node on the right. Determine if the corresponding elements in the link array corresponding to the nodes in $N_2(i)$ connected to the node are all greater than or equal to 1 after subtracting 1 from the count value. If so, it indicates that this very node is unnecessary and is removed from set $N_1(i)$. Subtract 1 from the elements in the set and repeat step (3); if not, proceed to step (4).

(4) Put this node into the preferred relay set, remove all its two-hop neighbours, and then repeat step (3) until $N_2(i)$ is empty.

In step (3), a one-hop neighbour is attempted to quit the optimal relay choosing procedure in descending order of connectivity. For the topology shown in the figure above, one-hop neighbour nodes g, e, a, and c quit the optimal relay choosing procedure in sequence, and nodes f, d, and b are chosen as the optimal relay nodes in sequence.

The optimal relay set for the current NT message sending can be composed of subtracting the pruning amount and adding new increments from the optimal relay set for the previous NT message sending, as shown in the following equation:

$$\xi_{cur} = \xi_{last} - \xi_{del} + \xi_{add} = \xi_{keep} + \xi_{add} \tag{1}$$

where ξ_{last} represents the optimal relay set for the previous NT message sent, while ξ_{keep}, ξ_{add}, and ξ_{del} represent the invariant, added, and removed elements of the existing optimal relay set compared to previous selection set, where $\xi_{keep} = \xi_{cur} \cap \xi_{last}$. According to the above equation, as long as the network topology changes not significantly, there will be significant redundancy between the current NT message content and the previous NT message. If sent repeatedly, it will cause significant network bandwidth waste. In response to this issue, this article proposes an NT content adaptive sending mechanism, and the improved NT message content is as follows:

$$\xi_{cur} = \begin{cases} \xi_{keep} + \xi_{add}\left(\xi_{keep} \le \xi_{del}\right) \\ \xi_{del} + \xi_{add}\left(\xi_{keep} > \xi_{del}\right) \end{cases} \tag{2}$$

In the above equation, the information in the currently sent NT message is dynamically modified according to changes in the Ad hoc network topology. A specific NT message sending situation can be divided into the following three situations:

(1) If the network topology nearby varies and $\xi_{del} < \xi_{keep}$, the NT message includes pruning and new increments.
(2) If the network topology nearby varies and $\xi_{del} \ge \xi_{keep}$, the NT message contains invariant and new increments.
(3) No NT messages is sent when the topology around the node has not changed.

When most nodes in the network are moving at high speeds and the communication link is unstable, based on the optimal relay selection algorithm mentioned above, the stability factors of nodes (links) are considered. After the periodic interaction of HELLO information, each node calculates the real-time survival time of the symmetric link and uses it for subsequent optimal relay and routing selection. Define node reliability as the average lifetime of links between nodes and neighbor nodes. The calculation formula for the reliability L_i of the i^{th} neighbor node is as follows:

$$L_i = \frac{t_0 + \sum_{i=1}^{D} f(t_i)}{D+1} \tag{3}$$

where

$$f(t_i) = \begin{cases} t_0, t_i \ge t_0 \\ t_i, t_i < t_0 \end{cases} \tag{4}$$

where t_0 is the link lifetime between the local machine and its neighbour node, t_i is the link lifetime between the 1-hop neighbour node and its i^{th} 2-hop neighbour node it covers, D is the connectivity of the neighbour node, and f is the conversion of the link lifetime between the 1-hop neighbour node and its 2-hop neighbour node. Since the existence of the link between the local machine and the 1-hop neighbour node is a prerequisite for the availability of this neighbour node, t_0 reflects the time that this neighbour node can still exist. If the survival time of other links in adjacent nodes is greater than t_0, within the extra time, the one hop link is no longer reliable, and the link survival time at this time has no practical significance, so it is converted to t_0; If ti is less than t_0, it is converted to t_i. The link lifetime here refers to the estimated duration of the link, which can be estimated based on the node's GPS information or received signal strength.

In the path calculation process, the Dijkstra algorithm based on hop count and link lifetime is used. Firstly, the shortest route is calculated according to the hop counts, and the weight of each edge in the topology is set to 1. The Dijkstra algorithm is used to find out the shortest route from the local machine to other nodes. Then, survival time of each link that each shortest path passes through is estimated, and the path is weighted using concavity, i.e.,

$$Weight_{route} = \min\{Weight_{link} | link \in route\} \quad (5)$$

Arrange all the shortest paths in descending order of weight, and when selecting the optimal relay, combine the weight information of the optimal relay set mentioned earlier to comprehensively select the relay node.

On the other hand, the predicted link lifetime is also used to dynamically adjust the sending period of HELLO messages. If the estimated link lifetime is longer, it means that the network topology near this node is stable, and sending period of HELLO messages increases accordingly. The next sending period of HELLO messages is,

$$Next\_Interval = Hello\_interval + \left\lfloor \log_{10} \frac{\min(link\_survival)}{Hello\_\mathrm{interval}} \right\rfloor \quad (6)$$

where *Next_Internal* is the cycle for the next HELLO message to be sent, *Hello_Interval* is the cycle of current HELLO, *link_ survival* is the link lifetime of one hop neighbor nodes. The more stable the local topology, the longer the cycle of HELLO messages, and the lower the routing message overhead.

In the proposed routing protocol, neighbor nodes are no longer notified of failure through separate routing error messages (such as RERR in AODV). Instead, in incremental update messages, the weight of the neighbor node (network interface) is reset to -1, indicating that the neighbor node (network interface) has failed. After receiving this message, the topology maintenance node resets the weight of the affected route to -1 and notifies other nodes in an incremental update manner, thereby saving the control message overhead of routing error notification.

4 Experimental Results

This section compares control message costs of our proposed routing protocol and two common-used routing protocols. We change the number of communication pairs, and the experimental results are given below (Tables 1, 2 and 3).

Table 1. Control message cost of our proposed routing protocol.

Communication pairs	Percentage of control messages	Percentage of data
10	75.0%	25.0%
20	60.0%	40.0%
30	50.0%	50.0%
40	42.9%	57.1%
50	37.5%	62.5%
60	33.3%	66.7%
70	30.0%	70.0%

Table 2. Control message cost of the classic reactive routing protocol AODV.

Communication pairs	RREQs	RREPs	RERRs	Percentage of data
10	86.5%	4.1%	2.3%	7.1%
20	89.6%	3.3%	2.3%	4.8%
30	89.9%	3.9%	3.2%	3.0%
40	89.5%	4.2%	3.5%	2.8%
50	90.4%	3.7%	2.9%	3.0%
60	90.0%	4.2%	4.2%	1.6%
70	90.3%	3.7%	3.2%	2.8%

Table 3. Control message cost of the classic proactive routing protocol DSDV.

Communication pairs	Percentage of control messages	Percentage of data
10	79.0%	21.0%
20	73.0%	27.0%
30	66.6%	33.4%
40	62.9%	37.1%
50	58.2%	41.8%
60	54.5%	45.5%
70	52.8%	47.2%

It can be seen that our proposed routing protocol has the lowest control message cost compared with either the classic reactive routing protocol or the classic proactive one.

References

1. Ananto, T.S., Ahmad, T.Z., Agung, N.: Wireless Ad hoc networks on motorcycle ride-hailing services: a comparative analysis of 802.11n and 802.11p. 2022 5th International Seminar on Research of Information Technology and Intelligent Systems (ISRITI), pp. 34–39 (2022)
2. Yohitha, N.P.M., Annavarapu, C., Kodi, S.S.: Route optimization of vehicular ad-hoc networks (VANETs). In: 2022 3rd International Conference on Intelligent Engineering and Management (ICIEM), pp. 442–446 (2022)
3. Ashraf, M.A., Marianne, A.A.: Ad-hoc networks performance based on routing protocol type. In: 2022 2nd International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC), pp. 464–467 (2022)
4. Feng, C.L.: Simulation of the performance metrics for securing DSDV routing protocol based on trusted environments. J. Phys. Conf. Ser. **2068**(1), 12–47 (2021)
5. Sangi, A.R., Alkatheiri, M.S., Anamalamudi, S., et al.: Spectrum handoff aware AODV routing protocol for cognitive radio vehicular ad hoc networks. Complexity **2021** (2021)
6. Peng, S., Liu, J., Weng, H.: An anti-interference on-demand routing algorithm for LEO satellite networks. In: 2022 IEEE 95th Vehicular Technology Conference: (VTC2022-Spring) (2022)
7. Natalia, K., Jan, P.: Performance analysis of AODV routing protocol and its modifications for MANET. In: 2023 World Symposium on Digital Intelligence for Systems and Machines (DISA) (2023)
8. Irshad, K., Somshekhar, D., Manjula, S.H.: Comparative evaluation of OLSR, DSDV and AODV routing protocols across varied propagation loss models in VANETs. In: 2023 IEEE 5th PhD Colloquium on Emerging Domain Innovation and Technology for Society (PhD EDITS) (2023)