Cache Mechanism for 6LoWPAN Based on Mesh-under

Fusheng Ji [a], Xiaoli Jing [b]

Chong qing University of Posts and Telecommunications, Key Laboratory of Optical Communication and Networks, Chongqing 400065, China

[a]89282262@qq.com, [b]jingxiaoli320@126.com

Abstract

In IPv6 over Low-power Wireless Personal Area Networks, when a node transfers these fragmentations based on Mesh-under, in order to be capable of restructuring the packets in destination, the missing fragmentations should be retransmitted in the means of end-to-end which results in greater energy consumption, higher latency and lower packet's reassembly rate. To solve the problem, this paper proposes a mechanism which uses the intermediate node to cache, manage and retransmit lost fragmentations. The simulation results show that the proposed mechanism can effectively improve the packet's reassembly rate.

Keywords

6LoWPAN, Mesh-under, fragmentation cache, retransmission.

1. Introduction

In November 2004, the Internet Engineering Task Force (IETF) set up the 6LoWPAN working group, designed to put IPv6[1] in the underlying IEEE 802.15.4 standard for wireless personal area networks, but also by introducing adaptation layer to optimize the relevant protocols[2]. In addition, 6LoWPAN has two kinds of routing protocols that is based on Route-over and based on Mesh-under, the former is based on the network layer, and the latter is based on the adaptation layer.

Based on Route-over, the fragmentations' transmission process[3] is that all fragmentations of the entire packet are restructured and re-fragmented in hop-by-hop way until the destination. Its feature is higher delivery rate, but more complex process and larger transmission delay. And based on Mesh-under, the fragmentations' transmission process[4] is that each fragmentation is individually transmitted from the source to the destination, and the fragmentations' transmission is completed and the destination completes fragmentations' reassembly. Its feature is that the transmission process is simple and lower delay, but the larger delivery rate and energy consumption.

Aimed at the drawback which transmits the fragmentations based on Mesh-under, the literature [5] proposes a mechanism that the source retransmits the destination lost fragmentations (Retransmit the Partial Fragments Based on the Source Node, RPF_SN), this mechanism points out that if the destination detects a fragmentation missing, all received fragmentations are cached and the source retransmits the lost fragmentations. Compared with the traditional source retransmits all fragmentations mechanism, RPF_SN improves network performance in some extent, but there are still a series of performance problems. Therefore, this paper proposes a the fragmentation's retransmit and buffer (Cache and Retransmit the Partial Fragments, CR_PF) mechanism, namely cache the transmission failure fragmentations in hop-by-hop and retransmit them from the cache node. Simulation results show that the proposed mechanism can effectively improve network performance, such as the packet's reassembly rate and so on.

2. Mechanisms accounting

In 6LoWPAN, in addition to the first and the last fragmentations, all intermediate frag- mentation size is equal. Therefore, this article assumes that the packets of source transmitted are the same size, and the packets are fragmented which leads to each fragmentation size is equal. The basic idea of the mechanism proposed in this paper is that the node caches those transmission failure fragmentations in hop-by-hop based on automatic repeat request (ARQ) from MAC layer; After that, node effectively manages cache space and retransmits the cached fragmentations according to the mechanism.

2.1 Cache process of data fragmentation.

In this paper, the adaptation layer of each node is set a certain volume retransmission cache. Based on Mesh-under fragmentation transmission, it makes the destination wait for receiving all fragmentations of a packet. Once a fragmentation is lost during transmission, it will make the entire packet's reassembly process stop, lead to the overall network performance degradation.

In 6LoWPAN, the link layer uses the IEEE 802.15.4 standard which uses ARQ mechanism to ensure the link layer data transmission reliability. Based on Mesh-under fragmentation transmission, it uses the ARQ mechanism from the MAC layer. According to the ARQ mechanism, the MAC layer of a node waits for the ACKnowledge (ACK) message from the next hop after sending fragmentation. Within the time that waiting for receiving an ACK message, if a node do not receive the message, it will perform retransmission (It is the maximum number of retransmission that depends on the predesigned retransmission times, its default value is 3 and its range is 0 to 7)[6]. After the default number of retransmission, if a node do not receive the ACK message which is returned by the next hop, it indicates that the fragmentation transmits failure in hop-by-hop. In order to avoid that the data fragmentation is retransmitted from the source, it should be cached in current node which transmits fragmentation failure; otherwise, as long as the node receives an ACK message returned by the next hop within the default number of retransmission, the data fragmentation transmission is successful, and it should continue the fragmentation subsequent transmission.

In caching the data fragmentation, the introduction of the cache entry(CE) is used to distinguish each fragmentation, such as CE (SA, PT, FO), which SA represents the source node address, PT indicates the packet label, FO represents the fragmentation offset and so on. At the same time, it need to open a fragmentation retransmission timer (RTO), setting RTO[7] puts to use a round trip time(RTT) of the data fragments arrives at the receive end, such as formula 1.

$$RTO = 1.5 \times RTT \tag{1}$$

2.2 Cache management and fragmentation retransmission.

From the fragmentation caching process, retransmission cache space of the node is limited. As fragmentation is cached that it may occur the following problems: first, the retransmission cache is overflow, resulting in the need retransmission fragmentation is lost, so these fragmentations will be retransmitted from the source; second, caching node waits for receiving retransmission request from the destination, and it may cause the node cache overflow while the retransmission time which it is waiting for increases.

To solve these problems, this paper considers them according to the following. When the remaining retransmission cache is small, the cache not only stores the transmission failure fragmentations but also avoids cache space overflow, reduces the overall transmission delay. So, it is resolved by adjusting the send fragmentation rate of the last hop node and initiating cached fragment retransmission mechanism.

Specific process is as follows. In order to be able to cache more data fragmentations and do not make the cache overflow in a limited cache space, it need to effectively regulate the send fragmentation rate of the last hop node according to the storage condition of the node cache space. Assume that the last hop node is A, and the next hop node is B. If the remaining retransmission cache of the node B is large enough, it is not to change anything. If the remaining retransmission cache of the node B is not large enough but enough to cache data fragmentation, you need to send an notification message (NM) to inform the last node A. Node A receives the NM message, it may adjust the send fragmentation rate according to the relevant information from the NM message. For example, it is shown in equation 2.

$$r_s^{'} = r_s\left(1 - \frac{nL}{c}\right), \quad L \le c_{rest} \le \frac{1}{2}c \tag{2}$$

Where r_s represents the sending fragmentation rate of the original node A, n represents the cached fragmentation number in node B, L represents a single cached fragmentation size, c represents the total size of the retransmission cache, c_{rest} represents the remaining retransmission cache size of a node. As shown in the calculation formula 3.

$$r_s = \frac{k}{m} \tag{3}$$

Where k represents the receive window size of node B on a moment, m represents the send window size of node A.

On the basis of adjusting node transmission fragmentation rate, in order to reduce the network delay that it needs to promptly transmit the cached fragmentations to the destination. Thus, there are two situations as follows.

First, a node receives an ACK message from the adaptation layer. The destination needs to restructure the received fragmentations, but it is only after receiving all fragments of a packet that it is able to restructure successfully. Therefore, before failing to receive all fragmentations of a complete package, the destination need to cache all received fragmentations, and generate an ACK message according to the cached fragmentations information and send it to the source direction. If an intermediate node receives the ACK message and it can find the corresponding fragmentation in accordance with the ACK message, the node retransmits the fragmentations and deletes them; otherwise, if there is no need to retransmit the fragmentations, the ACK message is transmitted to the source direction.

Second, the retransmission timer of a cache node expires. While a new fragmentation is cached, the node has opened a retransmission timer. If the node does not receive an ACK message from the adaptation layer and the retransmission timer expires, the node retransmits the cached fragmentations according to the first-in-first-out[8] principle and delete them; otherwise, it depends on the circumstances and deals with it accordingly.

3. Simulation analysis

In order to assess the performance of the mentioned mechanism, this paper uses the OMNeT++ simulation software for comparing CR_PF mechanism with RPF_SN mechanism of the simulation reference [5] mentioned in the case of the packet's reassembly rate. Table 1 lists the relevant simulation parameters.

Table 1 Set of Simulation Parameters

Simulation parameters	Default values
Simulation delay/h	2
Node bandwidth/kbps	250
Route protocol	LOAD
Retransmission timer/s	5
Max retransmission times in MAC	3
Packet size/B	500
Retransmission cache size/kB	2
Send packet interval/s	5
Transmission region/m	176.889
Simulation region/m^2	400*400

Destination's reassembly rate.

Based on the proposed fragmentation caching and retransmission mechanism, the intermediate node caches the transmission failure fragmentations in hop-by-hop and retransmits the fragmentations. It increases the number of restructure successful fragmentations in a certain period of time. The packet's reassembly rate is calculated as follows.

$$F = \frac{number\ of\ destination\ restructure\ packet}{number\ of\ source\ sending\ packets} \quad (4)$$

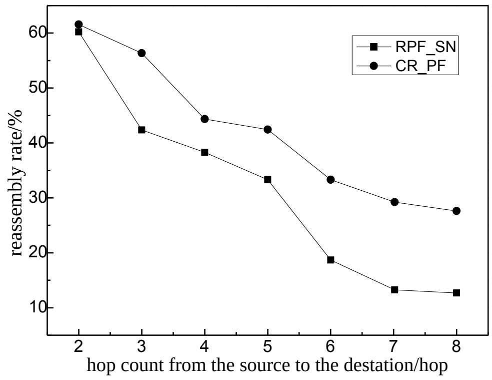

Fig. 2 The packet's reassembly corresponding to different hops

Figure 2 reflects that the packet's reassembly rate changes according to the hops' change from the source to the destination. The figure shows that as the distance increases from the source to the destination, the packet's reassembly rate of two mechanisms is gradually declining. For the RPF_SN mechanism, its fragmentation retransmission uses the method of the source retransmission, so that the delay and energy of the node used are more. That is to say, in a certain period, if the node consumes a certain amount of energy, restructuring packets number is relatively small. Compared with the RPF_SN mechanism, the CR_PF mechanism uses an intermediate node to cache the transmission failure fragmentations in hop-by-hop, thus avoiding from the source retransmitting the lost fragmentations, which relatively reduces the retransmission time and energy consumption. Therefore, compared with the previous article mentioned mechanisms, the CR_PF mechanism is able to effectively improve the packet's reassembly rate.

The energy utility value.

The energy utility value indicates the network energy consumption in a certain amount of successful restructure packets, which is calculated as follows.

$$E = \frac{number\ of\ destination\ restructure\ packets}{network\ energy\ consumption} \quad (5)$$

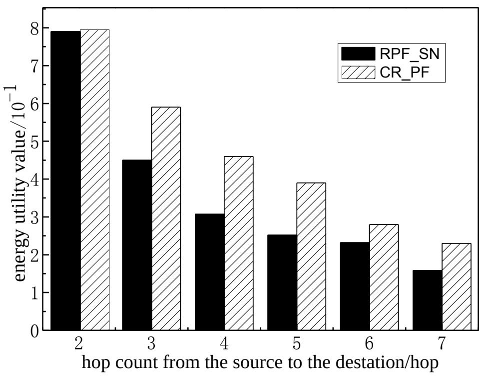

Fig. 3 The energy utility value corresponding to different hops

Figure 3 reflects both mechanisms that the utility value of energy consumption changes according to the hop count from the source to the destination. The figure shows that as the distance increases from the source to the destination, the utility value of energy consumption decreased, but on the whole, the effectiveness of the proposed mechanism is to be higher than the values used in RPF_SN mechanism. The reason is that the mechanism is based on the intermediate node to cache transmission failure fragmentations and retransmits them from the node. This not only reduces retransmission hop count and saves energy, but also makes the utility value of energy consumption higher.

4. Conclusion

In order to improve the 6LoWPAN networks' reassembly rate based on Mesh-under route, the paper proposes a fragmentation retransmission mechanism on the basis of the source retransmits the missing fragmentation mechanism in the literature [5]. The mechanism caches the transmission failure fragmentations on the transmission nodes, and these nodes' cache space is to be managed and these cached fragmentations are retransmitted to shorten the retransmission delay. Simulation results show that the proposed scheme can improve the packet's reassembly rate.

References

[1] Zhang M, Tang H, Long W, et al. An Overview of Mobile IPv6 [J]. *Communications Technology*, 2007(9): 26-28.

[2] Kushalnagar, N., et al., IPv6 over low-power wireless personal area networks (6LoWPANs): overview, assumptions, problem statement, and goals. RFC4919, August, 2007. 10.

[3] Winter T, Thubert P, Hui J, et al.. RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks [J]. *Internet proposed standard RFC*, March 2012.

[4] Kim K, Park S D, Montenegro G, et al. 6LoWPAN ad hoc on-demand distance vector routing (LOAD)[J]. *Network WG Internet Draft (work in progress)*, 2007, 19.

[5] Hu C H, Sun M H. An Enhanced Route-over and Mesh-under Routing Strategy [J]. *Communications Technology*, 2011, 12(44): 72-76.

[6] IEEE Computer Society. IEEE Standard for Information technology– Local and metropolitan area networks – Specific requirements – Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low Rate Wireless Personal Area Networks (WPAN)[S], IEEE Std 802.15.4-2006, October 2006.

[7] Dunkels A, Alonso J, Voigt T, et al. Distributed TCP caching for wireless sensor networks[C] //3rd Annual Mediterranean Ad-Hoc Networks Workshop, 2004.

[8] Wang J X, Lin B G. Active queue management algorithm based on particle swarm optimization [J]. *Journal of Computer Applications*, 2013, 33(02): 390-396.