Multibuffers Control Considering the Number of Replications in a DTN

Yuki Sugiyama and Shigeki Shiokawa
Graduate School of Information and Computer Science, Kanagawa Institute of Technology
1030 Shimo-ogino, Atsugi, Kanagawa 243-0292, Japan
E-mail: shiokawa@nw.kanagawa-it.ac.jp

Abstract

A delay/disruption-tolerant network (DTN) is a network that can operate in degraded environments, enabling end-to-end communication in discontinuous networks. A DTN uses a store-and-forward method. In this method, stored packets are conveyed by some terminals and forwarded to other ones. If there are packets having different priority, a low-priority packet may be pushed out from a buffer by a high-priority one. A crucial issue is that the arrival rate of low-priority packets is decreased by such pushing out. In this paper, we propose a method considering the number of packet replications. When the buffer is full, the node determines the packet to be discarded based on both the priority of the packet and the number of replications. By computer simulation, we show that the proposed method can increase the arrival rate of both low- and high-priority packets.

1. Introduction

In recent years, devices such as smartphones and tablets have rapidly become widespread. Infrastructure facilities such as base stations and public wireless local area networks (LAN) are also increasing. However, these facilities cannot be used when wide-area disasters occur. Also, communication methods that require infrastructure cannot be used. A network in which continuous connection among network terminals is impossible is called a subordinate environment network, and there is an issue that end-to-end communication is not always established. A delay/disruption-tolerant network is a network that realizes reliable packet transfer in such an environment [1].

Epidemic routing is one of the routing methods in a DTN. In epidemic routing, each terminal moves while accumulating packets in the buffer. If there is no terminal within its communication range, the terminal moves as it is. When another terminal exists within the communication range, the terminal replicates the packets in the buffer and transmits them there. The terminal that receives these packets saves them in its own buffer and moves. The terminal repeats these operations and delivers a packet to the target terminal.

2. DTN

A DTN is a network that enables packet transfer in a degraded environment using a store-and-forward method [1]. This method a packet transfer method, in which a terminal temporarily holds packets in its own buffer and sends them to another terminal as necessary. Therefore, the terminal can perform priority control or error correction control at the time of transmission.

3. Epidemic Routing

The epidemic routing protocol is a typical routing protocol in a DTN and an infective routing protocol. In this routing, each terminal moves while accumulating packets in the buffer. If there is no terminal within its communication range, the terminal moves as it is. When another terminal exists within the communication range, the terminal replicates the packets in the buffer and transmits them there. The terminal that receives these packets saves them in its own buffer and moves. The terminal repeats these operations and delivers a packet to the target terminal.

4. First-in-First-Drop

First-in-first-drop (FIFD) is a general packet destruction method [3]. When a buffer of a terminal is filled with packets and a new packet is received, the terminal must discard one of the packets. At this time, a packet that has been in the buffer for a long time has a high probability of having been replicated and transmitted to another terminal. Also, expiration of the packet lifetime may be close. For this reason, such a packet is considered as the one of whose existence has low value. Therefore, FIFD discards the packet in the head of the buffer and reduces the effect
of packet discarding. An example of FIFD operation is shown in Fig. 1. When the terminal newly receives packet A, it holds this packet if there is space in the buffer. However, since the buffer in Fig. 1 is full, the longest-held packet D is deleted, and the terminal stores packet A. FIFD does not consider the priority of the packet. Therefore, the terminal may discard a high-priority packet to save a low-priority packet.

5. Shared-Class-Based First-in-First-Drop

Shared-class-based first-in-first-drop (shared C–FIFD) is a discard method considering the priority of packets. In this method, each terminal has a high priority buffer, a low-priority one and a shared one. As a result, shared C–FIFD avoids the priority problem in FIFD and improves the arrival rate of high-priority packets.

An example of shared C-FIFD operation is shown in Fig. 2. Consider a case where the terminal newly receives low-priority packet E and high-priority packet F in the case of no vacancy in both priority buffers. In this case, the terminal discards high-priority packet Y and low-priority packet Z, which are the longest-staying packets in the respective-priority buffers. Also, high-priority packet C and low-priority packet D, which are the longest-staying packets in the shared buffer, are moved to the respective-priority buffer. Finally, newly received packets E and F are stored in the shared buffer.

6. Proposed Method

In conventional shared C-FIFD, the arrival rate of high-priority packets is improved by their preferential treatment. However, the total arrival rate of packets of both priorities is decreased. Therefore, we propose a method to improve the overall packet arrival rate by giving priority with respect to the number of replications.

In an epidemic routing, the terminal cannot always transmit all the packets. In other words, the number of replications of each packet is not uniform. The reason why the longest-existing packet in the buffer is discarded in FIFD is that this packet is considered to be replicated the most number of times. In practice, however, such a packet is not always the most replicated one in the buffer. Therefore, in the proposed method, the number of replications is recorded in the packet then the terminal discards packets with a large number of replications.

When a terminal encounters another terminal possessing data, it replicates the data that it does not have. As shown in Fig. 3, the number of times the data is replicated is held by many terminals. Thus, even if terminal A deletes data with a high number of replications, it will not be affected because the other terminals hold the same data.

The buffer operation is similar to that in the conventional method; each terminal is provided with a buffer for high-priority packets, a buffer for low-priority ones and a shared buffer. When a packet is received, it is stored in the same-priority buffer as the packet, and when there is no more space in each-priority buffer, the packet is stored in the shared buffer. When the shared buffer is full, a packet having the same priority as the received packet moves from the shared buffer to the corresponding-priority buffer. Thereafter, the packet with the largest number of replications in the buffer is discarded. Also, this method judges that the discarded packet may be sufficiently replicated and does not make a new request for this packet during a certain period. Our proposed method aims to improve the overall packet arrival rate by replicating many packets that are not often flowing in the network. Therefore, it keeps the packets with a small number of replications in the buffers for a long time.

An example of the operation of the proposed method is shown in Fig. 4. Consider the situation that high-priority packet A, with six replications, and low-priority packet B with ten replications are newly received when there is no space in any buffer. In this situation, our method discards high-priority packet X, which has the largest number of replications, 34, and low-priority packet M with 33 replications. After discarding these packets, high-priority packet I with 18 replications, and low-priority packet G with 17 replications are stored in the respective priority buffers. Then, packets A and B are stored in the shared buffer.
buffer. In addition, as shown in Fig. 5, terminals a and b communicate with each other and refer to each other's holding packet list to determine which packets to request. Since terminal a does not hold packets B, C and Z, they are candidates to be requested. At this time, since the packet deleted from the buffer is not newly requested during a certain period, packet Z is excluded from the request. Terminal b requests packets A, E and S because none of them are listed in the discarded packet list.

![Figure 4: Example of operation of proposed method](image)

![Figure 5: Packet request using discarded packet list](image)

7. Simulation

We demonstrate the effectiveness of the proposed method by computer simulation. As a simulation model, we assume a damaged area in the event of a disaster. The communicating distance between two terminals is set by considering a case where many shielding objects exist. Table 1 shows the simulation parameters. The simulation area is 500m square, in which 200 terminals are arranged randomly. Each terminal generates one packet per second, and the destination terminal of the packet is randomly selected among terminals in the network. The moving speed of the terminal is set among three patterns of 1 to 3, 1 to 7, and 1 to 10 m/s, and the movement of each terminal satisfies the Random Way Point model [4]. The communicating distance of each terminal is 10m. The buffer capacity is 45 packets, and we consider two types of packet: high-priority packets and low-priority packets. The packets generated ratio of high-priority and low-priority is 1: 9. The buffer size is 45 and 15 packets are allocated to each of the three kinds of buffer.

| Parameter          | Value                              |
|--------------------|------------------------------------|
| Area size          | 500(m) × 500(m)                    |
| Number of terminals| 200 units                          |
| Terminal moving speed| 1-3, 1-7, 1-10(m/s)               |
| Communication range| 10(m)                              |
| Data generation rate| 1 (packet/s)                      |
| Simulation time    | 2000(s)                            |
| Data lifetime      | 1000(s)                            |
| Buffer size        | 45 packets                         |

8. Results

In order to verify that the proposed method is effective, we compare the arrival rate and the number of arrival packets for the proposed method with those for the conventional method, shared C-FIFD.

First, from Fig. 6, the arrival rate of high-priority packets in the proposed method is up to by about 10% higher than in the conventional method. This is because the number of high-priority packets in the buffer can be increased to more than that in the conventional method by discarding packets with a large number of copies. When the terminal speed is 1-3 m/s, the difference between the two methods is less than the cases of 1-7 m/s and 1-10 m/s. This is because when the speed of terminal is small, the number of opportunities for communication decrease. In this case, the number of packet requests also decreases, and there is no significant difference in the priority of stored packets in the buffer.

For the low-priority packets, the difference between the arrival rates is larger than that for the high-priority packets. In the simulation, the generation ratio of low-priority packets to high-priority packets is set to 9 : 1. Therefore, in the proposed method, increasing the kinds of both priorities packets in the buffer, the kinds of upon low-priority packet become large than that in the conventional method. Also, when the moving speed of the terminal is 1-3 m/s, the difference between the two methods is small for the same reason as that for the high-priority packets.

![Figure 6: Packet arrival rate](image)

In order to confirm that the arrival rate increases with an increase of the kinds of both priorities packets in the buffer, we count the number of arrived packets at
simulation times of 500, 1000, 1500 and 2000 s. The results are shown in Figs. 7 to 9.

Figure 7 shows the number of arrived high-priority packets. It is confirmed that more high-priority packets arrived in the proposed method than in the conventional method during the simulation in most situations.

Figure 7  shows the number of arrived high-priority packets. It is confirmed that more high-priority packets arrived in the proposed method than in the conventional method during the simulation in most situations.

Figure 8 shows the number of arrived low-priority packets. We confirmed from Fig. 8 that the number of arrived low-priority packets in the proposed method exceeds that in the conventional method at any moving speed. In the conventional method, buffer control to prioritize the high-priority packets is performed. Therefore, as a result of increasing the number of arrived high-priority packets, many low-priority packets are discarded and the number of arrived low-priority packets decreases. In the proposed method, since the packets to be discarded are determined by the number of replications, many low-priority packets are not discarded and are retained.

From Fig. 9, it is shown that the total number of arrived packets with both priorities is higher than that of the conventional method at all terminal speeds. In the proposed method, since the number of replications is used as an indicator to discard packets, the number of discarded packets without sufficient replication decreases. Therefore, the number of packets arriving at the destination terminal increases.

Through these results, we showed that the control of the buffer by the proposed method is effective for improving the overall packet arrival rate.

9. Conclusion

In this paper, we proposed a method to improve the reachability of low-priority packets without significantly decreasing that of high-priority packets in a DTN. In the future, we plan to consider a method involving the exchange of packets.

References

[1] M.Y.S.Uddin, D.Nicol, T.Abdelzaher and R.Kravets: A postdister mobility model for delay-tolerant networking, IEEE Proceedings of Winter Simulation Conference, Austin, TX, pp.13-16, Dec.2009.
[2] S.Tajima, T. Asaka and T. Takahashi: Priority control using multi-buffer for street pass networks, IEICE Technical Report, NS2013-138, pp25-30, Dec.2013.
[3] L.Leela-Amornis, and H. Esaki: Heuristic congestion control for message deletion in delay tolerant network, Smart Spaces and Next Generation Wired/Wireless Networking Lecture Notes in Computer Science, Vol. 6294, pp.287-298, 2010.
[4] J. Yoon, M. Liu and B.Noble: Random waypoint considered harmful, IEEE INFOCOM Twenty-second Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE Cat.No.03CH37428), Vol.2, pp.1312-1321, 2003.