An Efficient Name-based Loss Recovery for Wireless Content Centric Networking

Gue-Hwan Jung, Woo-Sung Jung, Young-Bae Ko
Department of Computer Engineering
Graduate School of Ajou University, Suwon, Korea
{guehwan, woosung}@uns.ajou.ac.kr, youngko@ajou.ac.kr

Jae-Hoon Kim, Jong-Han Park
Software R&D Center
Samsung Electronics, Suwon, Korea
{jaehoonk, jhan12.park}@samsung.com

Abstract—This paper focuses on improving the reliability of packet transmission in wireless CCN. Various studies on wireless CCN have been conducted; however, only a small number have addressed such an issue of reliability. Due to the packet size limitation of the MAC layer, a segment from the upper layer needs be fragmented into smaller pieces. The problem is that a receiver cannot rebuild the segment even when only a single fragment is lost. Of course, the traditional CCN provides a solution with interest retransmission. However, this scheme causes a significant network overhead as it requires all the fragments, even the one(s) successfully transmitted, to be requested for retransmission. To solve this problem, we propose a novel scheme in which only missing fragments are requested. We compare the proposed scheme to the interest retransmission mechanism by means of implementation in order to analyze its effectiveness.

Keywords—Content Centric Networking (CCN); Reliability

I. INTRODUCTION

In recent years, a new paradigm has been proposed, namely, content centric networking (CCN) [1]. In contrast to existing web caching and content distribution networks in the current IP-based architecture, each CCN router can cache freely, while simultaneously forwarding contents. Therefore, a CCN router can provide the requested content on behalf of the original content servers that aids in alleviating the network load, thereby improving user experience. Most of the research on CCN has been conducted in wired environments for efficient cache management and interest forwarding [2-4]. However, some researchers have attempted to apply the concept of CCN for the wireless domain as well [5-7].

Various issues, such as interest flooding, duplicate data packet transmission, and transmission reliability, need to be tackled for wireless CCN. Among those, we argue that the reliability issue is the most critical because each CCN node forwards packets based on wireless broadcasting that does not support any ACK mechanism. Most of the research that has been done into the reliability issue has focused on how to effectively set the interest retransmission timer; however, we attempt to solve a duplicate fragment transmission here.

CCN provides an interest retransmission mechanism using a timer, but this is not sufficient for improving performance, due to duplicate fragment transmission taking place. CCN assumes that an interest message requests content sequentially, based on a segment unit that forms part of the content. The size of the segments in CCN varies, and in most cases the entire segments cannot be forwarded through wireless links because of the packet size limitation of the MAC layer. Therefore, each segment must be divided into multiple fragments which the receiver then gathers to rebuild the original segment. However, if even one fragment is not received, the receiver is unable to reconstruct the segment. When the retransmission timer expires, the same interest will be retransmitted. Notice that the receiver may receive duplicate fragments because the interest requests all fragments that form part of the segment.

In this paper, we propose such a loss recovery mechanism for wireless CCN, which recovers only those fragments that are missing and includes a new layer. The layer replaces the IP layer; thus, it carries out all tasks normally handled by the IP layer, such as fragmentation and reassembling. The layer periodically checks a fragment buffer which can store fragments temporarily to find incomplete entries. If such entries are found, the proposed scheme requests the missing fragments by means of a control message. The recovery procedure makes use of unicast forwarding to guarantee reliability. The results of our study show that the proposed scheme improves throughput and effectively reduces the burden on the CCN layer.

II. THE MOTIVATION AND RELATED WORK

A. The Problem of Interest Retransmission

Broadcast and multicast communication are capable of effectively disseminating the contents from a source to all nodes that request a packet in a single hop. However in general, no acknowledgement is supported and therefore packet loss is likely to occur more frequently in wireless environments, which is one of the most common causes of the decrease in network performance when broadcasting is applied. Thus, wireless network performance is determined by the level of reliability.

As pointed out in the previous section, a traditional CCN architecture provides an interest retransmission mechanism that is based on the elapsed time between the initial request and the current time, as in the case with TCP. If the difference exceeds a certain fixed threshold determined by the system in advance, the node considers the forwarded interest or data packet as having been lost while being transmitted via wireless medium. When the timer expires, the same interest is retransmitted until the number of attempts reaches a threshold.

However, it is difficult to effectively guarantee reliability when using only the CCN layer. Most segments are larger than the maximum transmission unit (MTU). Therefore, the
segments must be divided into multiple fragments in order to pass the link layer; however, reliability cannot be guaranteed for each fragment when using broadcast communication. As a result, a receiver may not be able to reassemble the segments one at a time in situations where fragment loss has occurred. In such an environment, after a certain amount of time has passed, the retransmission timer in the CCN layer will expire for a segment that is related to missing fragments. Thus, the same interest is forwarded in an attempt to retrieve the missing fragments, which may lead to unnecessary medium utilization due to the fact that useless fragments are also retransmitted. To reduce this overhead, in our method only the missing fragments are requested.

Fig. 1 shows the total number of duplicate fragments received according to the number of devices, until all clients have downloaded a complete file. Note that these experimental results were gathered with the same testbeds environment of Section V. In the figure, 4 KB and 8 KB represent the segment sizes, and it can be seen that the number of duplicate fragments at 8 KB is much larger than those at 4KB, because more frequent interest retransmission takes place with increased sizes. Particularly in the case of the original CCN 8 KB, the shape of the slope appears to be exponential as the number of devices increases, which means that the interest retransmission generates a huge recovery overhead.

B. Related Work on Reliability Issue

One of the easiest ways to increase reliability in broadcast communication is to send a packet to each destination, for example, by means of unicasting in the MAC layer [10]. However, this may increase the delay and network load due to ACK implosion and the forwarding of multiple duplicate packets because of packet loss. To reduce these problems, representatives are selected from each group to send the ACK to the source instead of all nodes in the group [11]. Another method is to exploit the RTS and CTS control packets in the broadcast medium window [12]. Although the described schemes can help to improve reliability, it is difficult to apply them to wireless CCN due to the cost and inefficiency that results from control message transmission.

For CCN, [7-8] propose an adaptive retransmission timer based on the round trip time, as with TCP. The retransmission timer is dynamically updated whenever a packet arrives at the requester. If restoration is not successful after the maximum number of attempts, the interest retransmission is ceased. This process is more effective than when a fixed timer is used, since the adaptive timer is able to reflect the current network status. In [9], a local repair algorithm that is based on application properties in the wired environment is proposed. Applications that are related to the file being downloaded must receive all segments from the sender. In live streaming applications for conference calls, however, it is not necessary for all missing segments to be repaired due to time-out. For separation validity, each application allows the interest packet to determine a priority as well as a specific segment range. For improving reliability, an effective interest retransmission mechanism based on the characteristics of the applications or network status is necessary, but this cannot solve the fundamental problem of duplicate fragment transmission due to fragmentation. The aim of our study is therefore to resolve this issue by means of loss recovery based on the fragment unit. It is possible to recover loss effectively at a lower cost than that of retransmission in the CCN layer.

III. THE PROPOSED SCHEME

CCN has been proposed as a new paradigm in which the IP layer can be replaced. However, the most representative open source, CCNx still uses the IP layer [13]. To apply the design concept of the CCN, we introduce a new link adaptor (LA) layer in between the CCN and the MAC layer as a substitute for the IP layer. Fig. 2 represents an overview of this novel network architecture. Here, LA layer performs two new functions, namely, lost fragment recovery and duplicated packet filtering, in addition to all the functions of the IP layer, such as fragmentation, reordering and defragmentation.

In order to support these additional functions, we define a new header, as in Table I, and also maintain a fragment buffer that can store fragments temporarily, as in the IP layer. The LA header is located behind the MAC layer and the role of the new fields is similar to that of the IP header for fragmentation and reassembling. To reduce the size overhead, we define certain fields based on a bit unit. Therefore, it is necessary for the reserved field for alignment in order to improve compiler performance. The type of packet is represented by 0 for control or 1 for CCN. Segment ID assists in distinguishing between the various segments of the content. Fragment offset is defined by the sequence of fragments where the right-most bit represents the first fragment. Thus, the system can support a maximum of eight fragments.
The system is capable of supporting the fragmentation and defragmentation processes by making use of the header. If the LA layer receives a packet that exceeds the MTU, the layer implements fragmentation. Each fragment is trimmed criterion on MTU, including the LA header, and the fragment offset is set according to the sequence of the fragments. After the fragmentation has been completed, each fragment is copied and stored in the fragment buffer, and then, the fragments are forwarded to MAC layer. The receiver is able to determine whether all fragments have been received by comparing the total number of fragments to the number of fragments received up to that point. If the numbers are identical, the layerreassembles the fragments sequentially and forwards a reassembled packet to the CCN layer. The duplicated packet filtering has a purpose to that of opportunistic forwarding. LA layer maintains a table which stores information about sent or received packets for a certain period of time. When LA layer receives a packet from CCN layer which has already registered in the table, the layer drops the packet. It is performed before the MAC layer and helps to prevent duplicate packet transmission. The fragment buffer manages fragments by means of a doubly-linked list data structure using the Linux kernel API, and the system restricts the buffer size for scalability. Every time a new entry is added, that entry is registered at the tail point of the fragment buffer. When the fragment buffer becomes full, eviction entries are selected in order, starting with the oldest.

We propose more efficient recovery mechanism, called as “Name-based Loss Recovery (NLR),” which is based on the fragment unit and exploits information of CCN layer, such as content name and segment ID. It is expected to effectively reduce the burden of the interest retransmission mechanism. If any fragments cannot be recovered by means of NLR, they will be restored by the interest retransmission. NLR requests missing fragments via lost fragment request (LFR) message. To increase reliability, both the LFR message and the fragment that is to be retransmitted by the message are forwarded by unicasting. However, even if the unicast improves reliability, it may generate duplicate transmissions. For example, if more than two neighbors of the destination of a LFR message want to receive the same fragments simultaneously, it is preferable for the missing fragments to be forwarded via broadcasting one at a time in terms of network resources. However, when using this method there is still the possibility of failure. In order to prevent continuous packet loss, we adopt the unicast recovery process according to the inherent nature of the recovery to place emphasis on the issue of reliability.

LFR packet consists of the number of elements field and multiple elements. It is located after the LA header. The number of elements field (4 bytes) represents the number of incomplete entries in the fragment buffer. Each element contains the content name, the segment ID and the fragment offset resulting from missing fragments. Multiple elements are forwarded to the same destination. Before an element is added, the system determines whether the entry has already existed in a recovery-ready queue based on the destination, which helps that the elements become aggregated. NLR is subdivided into two major processes: name-based segment recovery for missing segments, and name-based fragment recovery for missing fragments.

### A. Name-based Segment Recovery

Name-based segment recovery makes use of two assumptions. Firstly, segments arrive at the receiver sequentially, because the interest requests segments in order. Secondly, all segments in the contents have the same number of fragments, with the exception of the last segment. According to these assumptions, the receiver is able to anticipate the ID of the next segment that will reach it, based on the ID of the current segment. During this process, whenever the system receives a segment from the MAC, it determines whether the sequence is correct or not. If it is proved to be incorrect, the system will try to make this name-based segment recovery. The process can be conducted in various ways, and in our study it is implemented by means of a value that stores the maximum segment ID based on the segments received thus far. Each time a fragment is received, this value is updated. If the value is not increased sequentially when a fragment is received, segments between the value and the segment ID in the header are considered to have been lost. After the missing segments are found, all offset fields of the elements include all fragments by setting each bit in offset to 1. However, if the lost segment ID is less than the value which stores the maximum ID, it is impossible for the missing segments to be found. In this case, the segments can be recovered by means of interest retransmission. While this process includes missing segments, it is possible that many elements will be generated at one time. To overcome this problem, we place a limitation on the number of segments that can be recovered.

### B. Name-based Fragment Recovery

In this process for lost fragment recovery, a fragment buffer is inspected periodically to determine whether any incomplete entries exist. The period is determined in advance as a system parameter, and must be smaller than the interest retransmission interval; otherwise duplicate fragments may once again be generated frequently. This period affects performance in terms of delay and network load considerably. The more the period is increased, the more fragments are generated at the same time, and the effect may be even greater if the network is congested. Furthermore, with this period a minimum delay is inevitable, which may be critical for time-sensitive applications; therefore, it is important to select the most appropriate period length. The best way to do this is to set the period according to the situation.
and application properties; however, we exploit fixed periods in this paper.

We now provide an explanation of NLR. Fig. 3 shows the differences in the segment recovery process between IP-based CCN architecture and the proposed architecture. We assume that a content segment is divided into three fragments. Even if a receiver has never asked for the segment numbered 3, it can receive the one in the case when other nodes have made the same request. As shown in Fig. 3, if all fragments of Segment 2 are lost, the receiver can recognize quickly it compared to the interest timer upon receiving Segment 3. Also, if received segment id is larger than three, the receiver can receive segments between 2 to the received segment id. The reason why it is possible is because there is high probability that the sender stores the fragments. The procedure may help to recovery lost segments and receive rapidly segments in advance before request via sequential interests.

Fig. 4 presents the differences from the fragment recovery process’s point of view. If the receiver has not received fragment numbered 2, it cannot reassemble the original segment, namely, Segment 1. In the IP-based CCN, the IP layer must wait until the timer has expired to perform interest retransmission. However, the process inspects the fragment buffer for incomplete entries. In detail, in Fig. 5, node B requests /Ajoy, and then node A sends fragments that form part of segment 1. While these fragments are being sent, Fragment 2 is lost; therefore, node B cannot completely reassemble the fragments. After the fragment buffer is inspected, node B knows which fragments have not yet been received. The node creates a LFR message by setting each field, as shown in Fig. 5, based on the information obtained from the fragment buffer. Deciding on the destination of the LFR message is an important problem for performing recovery reliably. There is a high probability that a node that has sent fragments a moment ago, node A, now has the missing fragments. Therefore, we decide that node A will be the destination of the message. In order to apply this, the source MAC address of the received fragments is stored in the fragment buffer, with the exception of segments from the CCN layer.

There are certain points that must be taken into account for effectively conducting NLR. Firstly, unnecessary LFR messages may be generated repeatedly in situations where the destination may not have received the missing fragments. Therefore, it is necessary for the system to include a criterion to judge whether the recovery process will be successful. For this reason, we define a maximum count of attempts as a system parameter, in order to prevent continuous transmission that is of no use. Secondly, if an incomplete entry is selected as an eviction entry, the same situation as described above may occur. For example, when a retransmitted fragment that has been generated as a result of an eviction entry arrives at the node, a new incomplete entry will be created, because the existing entry has already been removed. Therefore, when selecting an eviction entry, one of the following two conditions must hold. Firstly, an eviction entry must contain all of its fragments, and secondly, the maximum transmission counts of a LFR message must be equal to the threshold. If either of these conditions is satisfied, the selected entry can be removed.

The number of fragments making up a segment determines the recovery coverage of NLR. As fragmentation increases, the recovery carried out by NLR becomes more frequent. This in turn increases the probability of the node receiving at least one of the fragments. However, NLR cannot recover the lost in the situation where no fragments of a certain segment are received or the lost segment ID has a lower value than the received maximum segment ID. In such cases, the loss overcomes only
the interest retransmission. Therefore, the two processes work interdependently, as indicated by the number of interest retransmissions in the results.

IV. PERFORMANCE EVALUATION

A. Environments

We assume a scenario where audiences can download a helpful material from a presenter before or during the presentation. We conducted our study using nine devices running Linux 3.4.37. Each device had an Intel Core i7 processor 3630QM with 8 GB DDR3 memory. In addition, each device had an Intel wireless chipset capable of supporting 802.11a/b/g/n. Our experiment was conducted in the 2.4 GHz band, and all devices were operating in ad hoc mode. Both interest and data packets are forwarded via broadcasting. Therefore, the packets are basically delivered using the lowest data rate in order to cover long distances. However, we believe that it is difficult to download text files or support multimedia contents using this data rate, which is essentially provided by the device driver. Thus, it should be noted that it is necessary for the broadcast rate to be modified, which can be achieved by using a soft MAC in the device. We modified the broadcast rate to be 24 Mbps, but did not modify the unicast rate. We set the MTU at 1500 bytes in our experiment.

Nine devices, consisting of a content server and eight clients, were included in the sing-hop. Each device was equipped with CCNx [13] in order to implement the CCN layer. All clients requested a text file of 10 MB from the content server simultaneously, and we set the segment sizes at both 4 KB and 8 KB in order to analyze NLR performance according to the number of fragments. Therefore, the number of fragments is three and six in the 4 KB and 8 KB sizes respectively. We fixed the interval of the interest retransmission timer at one second. The number of attempts of interest retransmission was unlimited, because the file being downloaded could not endure loss, as opposed to, for example, a video streaming service. In addition, the maximum number of attempts of NLR was set to three. The period during which NLR could find incomplete entries was fixed at 200 ms so that the attempts of NLR could be exhausted within the interest retransmission interval.

B. Results

We measured the number of interest retransmissions, as shown in Fig. 6 (a). As previously reported, we did not put any limitations on the retransmission count. Therefore, interest retransmission occurred continuously, until the missing segments were successfully retrieved. The number of packets when using the 8 KB size is larger than that of the 4 KB size because of the number of fragments. In the case where NLR was applied, the measured values were zero, which means that NLR reduces the number of retransmission considerably by recovering only the missing fragments.

Fig. 6 (b) illustrates the success rate of NLR when the number of clients was eight. The number of fragments in LFR represents the number of missing fragments for all clients, while the number of received fragments by LFR represents the fragments that are forwarded by LFR messages to fill on the holes. The number of missing fragments requested by clients through LFR messages was approximately 762 and 1572 in 4KB and 8KB. The success rate was yielded by the number of fragments in LFR divided by the number of received fragments by LFR. The rate was calculated to be exactly 100% for both 4KB and 8KB, as shown in Fig. 6 (b), which verifies the reliability of NLR. Based on the results achieved thus far, we anticipate that the proposed scheme will be effective in situations where only one fragment has been received from the source.

We also evaluate the average throughput, as illustrated in Fig. 6 (c). The average throughput is calculated by dividing the total file size by downloading time for an individual device. The number of fragments in the 8 KB segment size is double that of the 4 KB one, which means that it is difficult for all fragments to be received for reassembling due to packet loss. As expected, the result indicates that the performance when using the 4 KB size was superior to that achieved when the 8 KB size was used. In addition, when NLR was used, better performance was achieved than the IP-based CCN. The deviations in the average throughput remained at similar levels for each device, despite the number of devices being higher than in the IP-based CCN, which shows that the reliability of NLR is comparatively and stable when using unicasting. We furthermore periodically measured the downloaded file size per second, which we do not describe in this paper. In the case of the IP-based CCN, we regularly came across sections where the file size did not increase for the interval, because the IP-based CCN must wait for the retransmission interval to elapse before recovering missing fragments. On the other hand, in case of NLR, the downloaded file size increased constantly, as was evident whenever we checked the size. According to this finding, NLR also has an effect on real-time multimedia applications in terms of continuity of play.

We showed the total number of duplicate fragments in Fig. 6 (d). In detail, when the number of clients is eight, 2196 and 280 are recorded in IP-based CCN and in NLR in 4KB. Also, in 8KB, 20350 and 1038 are measured respectively. The result shows that substantial useless fragments are generated because of the interest retransmission. The result also verifies that the recovery process based on fragment unit effectively reduces the overhead.

However, the average throughput was increased slightly in the NLR when the network overhead of the two was compared. This can be explained by the effect of the interest retransmission. For example, if the same segment is missing in three different devices, with a bit of luck the IP-based CCN may recover the lost fragments with one interest retransmission, in spite of the devices being different. Even if the same data packet or interest is forwarded in the CCN layer, these are filtered by means of duplicate checking in the LA layer. In the case where the segment ID is the same but the missing fragments differ, the effect is still valid. In an environment where the same content is requested using broadcast communication, the IP-based CCN benefits more from this effect. In contrast, the NLR can recover the lost three times by using unicasting in a situation where three devices have one fragment of the same segment missing, because they cannot overhear each other. All of these observations may affect the results obtained.
V. CONCLUSION

We propose a recovery mechanism based on the fragment unit and system architecture in order to reduce network overhead and the number of interest retransmissions in CCN. The short interval on the interest retransmission quickly copes with the fragment loss. However, its success is uncertain compared to NLR. Also, it may generate considerable network overhead because of using broadcast communication. NLR periodically inspects the fragment buffer to search for incomplete entries. Therefore, it is ideal for the period to be set dynamically, depending on the situation and application properties, in order to improve performance. Even if a fixed period is exploited, as in this paper, the implementation results show that the scheme improves performance. We also expect NLR to detect network status according to the variation in the number of elements. Therefore, in future work we will develop a congestion control algorithm for CCN via NLR.

ACKNOWLEDGMENT

This paper has been partially supported by IT/SW Creative research program supervised by the NIPA (National IT Industry Promotion Agency) (NIPA-2013- H0502-13-1059)

REFERENCES