Cluster-based Cooperative Caching and Forwarding in CCN

Kyi Thar, Choong Seon Hong
Department of Computer Engineering ,Kyung Hee University
{kyithar, cshong}@khu.ac.kr

Abstract

Catching is one of the important issues in Content Centric Network (CCN). The default CCN caching scheme mainly focus for redundancy. However, the usage of the cache is inefficient because, CCN Routers store the whole Data in their Interest forwarding paths and multiple duplicated data are in the network. We would like to reduce cache storage usage and forward Interest message efficiently. In this paper, we propose a cluster based chunks de-duplication and forwarding mechanisms within an Autonomous System (AS). A content data is divided and parts of data are cooperatively stored in multiple routers. Thus, cluster based cooperative caching and forwarding mechanism can significantly reduce the Content storage size on each router as well as forwarding Interest traffic in the network.

1. Introduction

The Internet architecture is inefficient for current usage like a content distribution. Content Centric Network (CCN) is the new architecture for future Internet and an alternative way to look at networking. CCN is driven by the two types of packets, Interest and Data .CCN nodes use an Interest to request a Data. The Data is a Content object (videos, documents, etc.). In CCN, Content exists in the form of a sequence of segment data which called chunks and a Content Router stores all the chunks [1]. The default caching system is good for redundancy, but the usage of the cache storage is inefficient [2]. Non-cooperative network caches may lead to non-optimal behavior and non-popular data are stored in common paths, but popular data are dumped on the edges [4]. By default, forwarding is done by the multicast manner. Therefore, CCN Router suffers a lot of traffic by forwarding Interest.

The cooperative caching scheme is proposed in [3]. In this scheme, the chunks of Content data are stored in multiple routers. All the routers in the system need to update *Collaborative Content Store* (CCS) periodically. It causes CCN Routers suffer a lot of workload and, the forwarding Interest is not also efficient to retrieve the Content which are stored in another AS.

In this paper, we propose the Cluster-based cooperative caching and forwarding scheme to solve the problem of default CCN by constructing several Content-Router clusters within an Autonomous System (AS). The Contents are stored as sequences of chunks on one cluster without duplicate and routers can forward Interest and retrieve Content efficiently.

The rest of this paper is organized as follows. Section 2 present the proposed Cooperative Caching and Forwarding mechanism. Section 3 concludes the paper and future work.

2. Cooperative Caching and Forwarding Mechanism

In our proposed mechanism, consist of three parts, cluster forming, cooperative Data storage and cooperative Data retrieving. In this proposal, the nearby routers are formed as clusters. The chunks of the content data are stored in different routers without duplicate. In order to get non-duplicate chunks numbers, chunks filtering can be done by using simple modulus method as in [3]. Therefore, each router stores de-duplicate data and each router knows which chunks are stored in which router. When a router receives an Interest, it can reply the content data instantly if the data is stored in its content store. If content data is not found in its content store, the router can directly forward to the router which stores the content data for the request Interest.

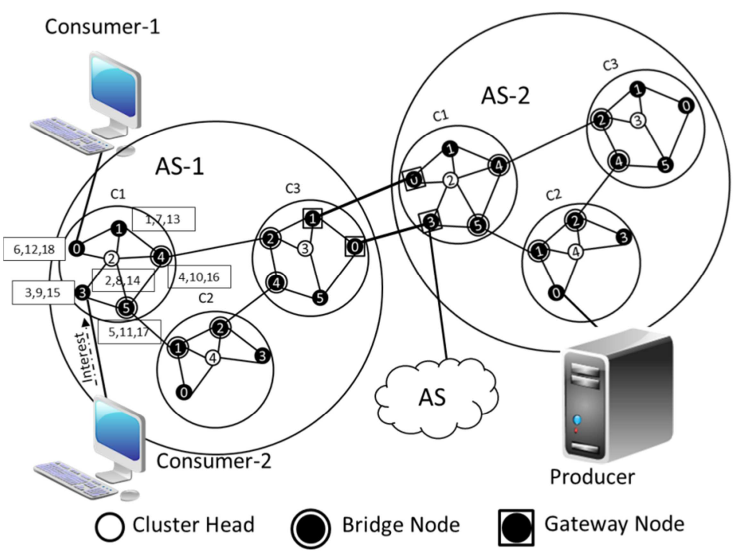

Figure 1: CCN cluster architecture

2.1 Cluster Forming

In a cluster there are three different routers: *Cluster Head* (CH), *Bridge Routers* (BR) and *Gateway Routers* (GR).BN is a router which connects with another cluster and GR which connects with other AS. CH collects the information of its own cluster and distributes that information to cluster members. So, cluster member can construct the *Cluster Member Table* (CMT).

At the initial stage, a router declares itself as a CH and broadcasts the clustering request message to its one-hop neighbor routers. If a neighbor router receives a clustering request, if it is willing to be a member of the cluster, it will reply the request and becomes a cluster member. Otherwise, it can deny the request and declare itself as a CH or it can reply the request from the different clusters. When CH had collected the replies from the neighbor routers, it can know how many routers are in its cluster and CH assigns the label to each cluster member as $R_i,\ i \in \{1,2,\dots,N\}$, where N is the total number of routers in a cluster. Then CH assigns the router which connects to the other cluster as a BR and one which connects to the other AS as GR. In a cluster, there may be multiple BRs and GRs. After assigning BR and GR, CH distribute the cluster information to a cluster member to construct the CMT. So, every router in the cluster knows each other.

2.2 Cooperative Data Caching

In Content Centric Network, Data are constructed with chunks and each chunk is labeled with sequence number. When a cluster member router receives a data with labeled chunks, it stores only corresponding chunks to its content store.

The decisions to store or send Chunks are done by using *Cache decision Algorithm* (CDA). By this algorithm, router stores the corresponding chunks on its Content Store. After that, it will forward the rest of chunks to the other routers in the cluster. Other cluster member routers perform the same way and store only corresponding chunks in their content stores.

Algorithm 1: Cache decision

function Cache Decision (Chunk)
  j= chunkNo
    $k = j \bmod N$ // here, N is the number of router in cluster
    If $R_k == R_c$ **then** //here, R_c is the current router
      Store (*j*)
      Send (*j*) to the Consumer
    else
      Send (*j*) to the Consumer
      Send (*j*) to R_k
end

In the CDA, ***j*** represents the chunks number (0,1, 2, 3… j) of a content object and ***N*** represents the total number of routers in a cluster. Fig. 1 shows the detail procedure of chunks filtering and cooperative data storage mechanism. In AS-1, the router R_c receives a sequence of chunks ***j*** (1, 2, 3, 4… *j*) and it checks the chunks with CDA. By CDA, current router R_c stores only the chunk which ***k*** value is same with current router label and it sends other chunks *Consumer-1*by PIT .If the received chunks are not for current router R_c ,it sends those chunks to the Consumer-A and to the router R_k. So that, CDA guarantees that no duplicated chunks are stored in any router within a cluster.

2.3 Cooperative Data Retrieving

In cooperative Data Retrieving, when a router of cluster receives an Interest, it checks, if the content data is stored in its router or not. If the designate data is stored in its content store, it can deduce that the data for the interest is stored in the cluster members. Then, the corresponding chunks are directly requested to the dedicated router by using the *Data retrieving Algorithm* (DRA).

Algorithm 2: Data retrieving

function Retrieving (Interest)
  i=Interest.seqNo
    $k = i \bmod N$ // here, N is the number of router in cluster
    if $R_k == R_c$ **then** //here, R_c is the current router
      if InterestName ← ContentStore.Find (Name) **then**
        ChunkNo ← ContentStore.Check (seqNo)
        Return (Chunk)
      else
        Forward (Interest) to the *BR* or *GR*
    else
      Forward (Interest) to the R_k
      if InterestName ← ContentStore.Find (Name) **then**
        ChunkNo ← ContentStore.Check (seqNo)
        Return (Chunk)
      else
        Forward (Interest) to the *BR* or *GR*
end

In the DRA, ***i*** represents the Interest sequence number, ***N*** represents the total number of routers in a cluster. In figure 1, *Consumer-2* sends an Interest to router R_c which checks the sequence number of the chunk ***i*** with $k = i \bmod N$. If the k value matches current router R_c, which checks the content name and sequence number of chunks. If the content name and sequence number is found, the current router R_c will return request chunk .If the content is not found, current router R_c forwards the Interest to BRs or GRs.

If the Interest is not for current router R_c,it will forward to the dedicated router R_k .When Router R_k receives the Interest ,it checks content name and sequence number of chunk. If the content name and sequence number of the chunk is found, that chunk is returned. If do not found, router R_k will forwards the Interest to BRs or GRs. So, when request content is not in the cluster, the DRA can forward Interest effectively to other cluster and other AS.

4. Analysis

We analyze the CCN network with cluster and without cluster. Each router can store 5000 chunks and uses LRU policy. The input chunk value for analysis is 1000 to 10000 and one content consists of the 20 chunks.

In figure 2, chunks are stored uniformly on each router of the cluster based CCN network. So, five routers store 1000 incoming chunks as 200 chunks on each router and that figure show the impact of the cluster based Cooperative caching .If the cluster is bigger, the usage of cache on each router is smaller and smaller.

Cluster based CCN stored chunks uniformly on each router. So that, the cluster based CCN network can store more diversity content than the normal one.

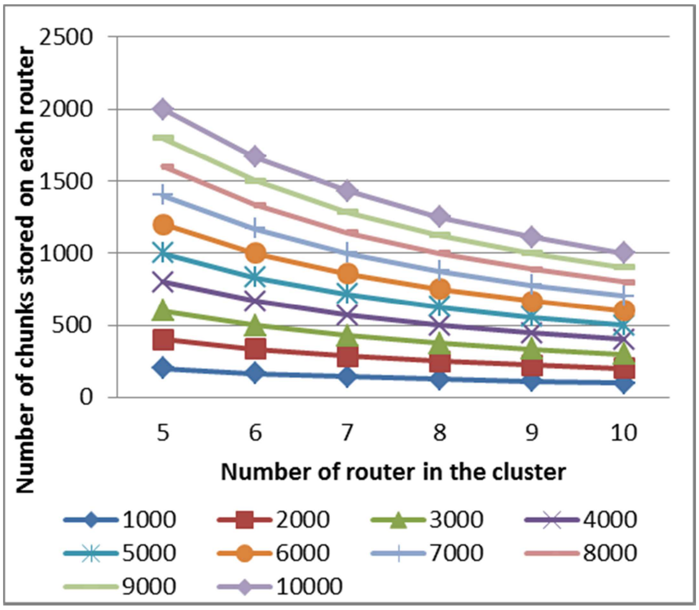

Figure 2: The number of chunks stored on each router in cluster based cooperative caching

Figure 3, shows the number of chunks stored on the normal Content Centric Network. By comparison of two caching mechanism, the propose cluster based cooperative caching can reduce cache storage size on each router than normal CCN caching.

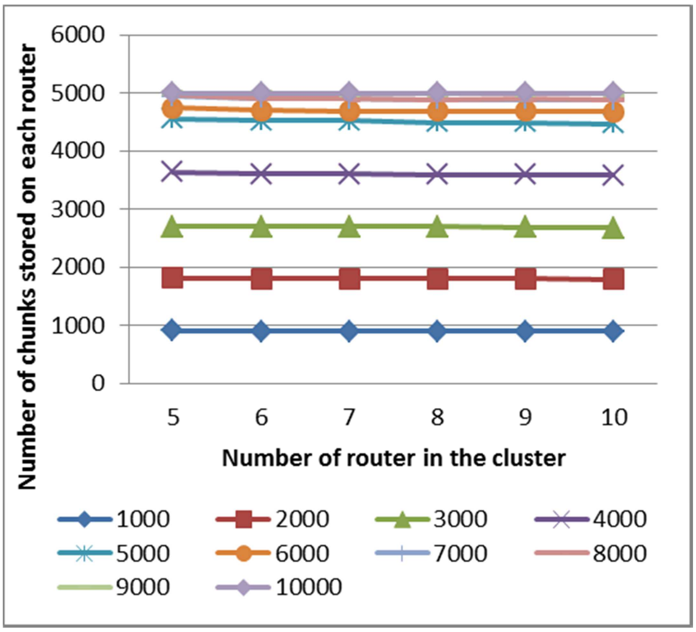

Figure 3: The number of chunks stored on the normal content centric network

3. Conclusions

We have presented a Cluster-based Cooperative caching and Forwarding for CCN. This work enables the router to stores Contents without duplicate within cluster using the simple modulus method as [3] and each router can forward and retrieve Content effectively .In our future work, we will do a detailed analysis of the proposed mechanism.

Acknowledgement

This paper was supported by NIA (National Information Society Agency). Dr. CS Hong is the corresponding author.

References

[1] L. Zhang et al., “Named Data Networking (NDN) Project”, PARC Technical Report NDN-0001, October 2010.

[2]V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, andR. L. Braynard, “Networking named content,” in Proc. of ACM CoNEXT, 2009

[3] Z. Li and G. Simon. Time-shifted tv in content centric networks: The case for cooperative in-network caching. In IEEE ICC, 2011.

[4]Saha, Sumanta, Andrey Lukyanenko, and Antti Yla-Jaaski. "Cooperative caching through routing control in information-centric networks." INFOCOM, 2013 Proceedings IEEE. IEEE, 2013.