ACCEPTED FROM OPEN CALL

A Withered Tree Comes to Life Again: Enabling In-Network Caching in the Traditional IP Network

Kaiping Xue, Tingting Hu, Xiang Zhang, Peilin Hong, David S.L. Wei, and Feng Wu

The authors propose in-network caching in IP-based networks by adding a content identifier into a newly defined IPv6 extension header, where the new architecture is named CAIP. CAIP abandons the complicated name-based forwarding table in ICN, and instead integrates IP routing lookup with cache index lookup, which is compatible with the IP network and also inherits the proven advantages of ICN.

ABSTRACT

This article presents our work proposing in-network caching in IP-based networks by adding a content identifier into a newly defined IPv6 extension header, where the new architecture is named CAIP. CAIP abandons the complicated name-based forwarding table in ICN, and instead integrates IP routing lookup with cache index lookup, which is compatible with the IP network and also inherits the proven advantages of ICN. Cache index exchanging and cooperative caching are implemented between one-hop CAIP enabled neighboring routers, which is simple but efficient. Moreover, for per-chunk caching, as an extension, bitmap is introduced to merge multiple request packets into one. Performance analysis shows that CAIP gains significant performance improvement in terms of access delay and traffic load.

INTRODUCTION

Recently, Internet access from mobile devices has grown dramatically in popularity, and according to Cisco's VNI report [1], 80 percent of all IP traffic will be represented by video traffic in 2019. Hence, content retrieval applications will contribute to most Internet traffic. However, while the demand for multimedia contents has increased tremendously in recent years, the capacity growth of the wireless link, mobile radio network, and mobile core network cannot practically cope with the explosively growing bandwidth demand. Moreover, the IP network is designed and treated as dump pipes, which makes the network carry a large number of duplicate data.

In fact, consumers are usually interested in the contents themselves rather than where they are located. Meanwhile, in the current Internet, a significant proportion of the tremendous increasing network traffic is from duplicate requested contents. Therefore, a feasible method should be integrating content cache and delivery as a legacy network feature, which means the contents can be cached by any network entities equipped with high-performance storage. By means of some new network technologies, such as software defined networking (SDN) and locator/identifier separation, some research work has introduced in-network caching into the architecture design, such as [2, 3]. However, these schemes usually require centralized servers to dynamically maintain the mapping of cached contents and the corresponding locations, which will lead to the scalability issue. Based on the consideration of fully distributed processing, a number of innovative new Internet infrastructures shifting from the current host-to-host communication model to a receiver-driven content retrieval model have been proposed. These innovative network design schemes for the future Internet architecture are uniformly called information-centric networking (ICN)[4], which is currently being investigated and developed in several projects, such as Named Data Network (NDN) [5] and Data-Oriented Network Architecture (DONA) [6]. In spite of some distinctive differences between them (e.g., content naming, security mechanisms, routing strategies, cache management), they share a common property of a receiver-driven data exchange model based on content names (or identifiers). The primary goal of ICN is to facilitate in-network caching for universal content caching in every internal network node, and it enables routers to cache passing-by data to satisfy subsequent requests. It can effectively reduce the distance between the consumers and the content data, and the caching policy can adapt to any dynamic traffic without any specific deployment.

However, this type of clean-slate approach has created a trajectory that is to replace the current IP-based Internet. ICN comes with some significant drawbacks and is complicated to execute [7]. For instance, replacing IP by ICN as the main Internet protocol comes with burdens of not only tedious standardization procedures, but also agreements among many stakeholders involved in the current Internet, such as operators, vendors, and policymakers. Moreover, although the idea of hierarchical name structure is introduced in NDN, the huge volume of contents in the Internet still requires a huge number of content prefixes, which can result in the size of the forwarding information base (FIB) in NDN usually being several orders of magnitude larger than that of the IP routing table in the current Internet. Meanwhile, routers in NDN need to handle routing updating due to content publishing or deletion, and caching update as well as caching policy, which results in the FIB being updated much more frequently than the traditional IP routing table.

Furthermore, some aspects of ICN have not yet been recognized and still require practical

Digital Object Identifier:
10.1109/MCOM.2017.1600487

Kaiping Xue, Tingting Hu, Xiang Zhang, Peilin Hong, and Feng Wu are with the University of Science and Technology of China; David Wei is with Fordham University.

solutions, such as content name resolution, and efficient name-based forwarding table maintenance and lookup (e.g., the longest prefix matching of a variable-length hierarchical name). In addition, it is worth mentioning that a certain percentage of traffic (about 30 percent now, and it takes up a higher proportion of the session number) is still generated by end-to-end sessions and relies on host addresses (e.g., voice calls, emails, instant messages), in which involved contents are not repeated or requested by multiple customers. For these end-to-end sessions, compared to the IP network, ICN has no any advantages.

Therefore, an awkward situation arises. On one hand, with the progress of technologies, core network devices can have strong computing power and large enough storage capacity, in addition to transmitting capacity, but not be well utilized to reduce redundancy. How to leverage core network elements' growing computing and storage capacity to reduce redundancy is a pressing issue that demands prompt solutions. On the other hand, the future Internet architecture aims to replace the current network, but it is still debating, and more research is ongoing. Thus, no one else can completely replace the existing network architecture within a predictable time.

Therefore, the appropriate approach is to gradually improve and evolve the current IP network architecture, which requires the features of optional in-network caching, supporting legacy TCP/IP-based applications, and being capable of processing traditional IP packets. We define an IPv6 extension header to make legacy IP packets content-aware. Also, we still use traditional IP routing and use longest prefix match (LPM) for IP forwarding. Recently, Detti *et al.* [8] defined an IP option to make IP packets content-aware; their work is named CONET. They do a measurement to show that the added IP option handling is not a critical performance bottleneck. However, it is essentially still an ICN scheme over an IP network, which has complicated name-based forwarding tables, and a source routing mechanism is introduced to guarantee bidirectional communications to traverse the same routers. The main purpose of CONET is to achieve the coexistence of traditional IP and ICN, and finally transit the network architecture from IP to ICN. However, the goal of our scheme is to enable traditional IP to have in-network caching with no need for complicated name-based forwarding. Like the Network Address Translation (NAT) technology, CAIP can make traditional IP to come to life again.

Our contributions in this article can be summarized as follows:

•We define a new IPv6 extension header to carry the content identifier, which can make legacy IP packets content-aware and enable the routers to have the capacity of in-network caching and content retrieval. For routers in the network, the processing of the extension header is optional, which means that it does not require all routers to support this newly designed extension header and the corresponding function processing. The existing legacy routers can still work well and just treat the IP packets with the new extension header as the legacy ones so that the new network architecture can be deployed step by step. The more CAIP enabled routers, the better the performance of the system.

•We enable each CAIP enabled router to implement local lookup in a cache index table and legacy IP-based routing with no need for complicated name-based routing and forwarding. This way, it can still deal with in-network caching and fast forwarding. The cache index exchanging and cooperative caching are implemented between one-hop neighboring CAIP-enabled routers, which can increase the local cache hit ratio without much interaction complexity.

•Aiming to reduce the number of request packets, we introduce the bitmap structure into the newly defined IPv6 extension header.

The rest of this article is organized as follows. We state our motivation and describe the proposed CAIP framework in the following section. Then operations in a router are described. Neighboring CAIP router discovery and cooperative caching policy are given. Moreover, for per-chunk caching, an extension of introducing bitmap in CAIP is proposed to reduce the number of requested packets. Then performance evaluation is shown, and the final section draws conclusions about our work.

Our Proposed CAIP Framework

The Main Design Motivation

CAIP is a new architecture to achieve in-network caching in the current IP network, which needs no change of the current host-to-host communication model; also, it is content-aware for content retrieving services. On one hand, because CAIP is intrinsically supported by the underlying IP routing, such end-to-end communication sessions could continue to be supported, and CAIP can achieve fast IP forwarding. On the other hand, CAIP facilitates in-network caching in core network so as to ensure that the content services such as video streaming can be fetched much more closely and quickly.

Basic Components of CAIP

For achieving in-network caching in the current IP network, a content identifier is carried in the extension header of an IPv6 packet, which makes the IP packet content-aware, so routers can quickly process the packet without deep packet inspection to fetch the content message. In addition, CAIP enabled routers can handle the processing of the new defined extension header in IP packets, manage content record entries, and have the capability of caching. Figure 1 shows the process of establishing a service session in CAIP, and also gives the format of the novel IP extension header. Such an architecture enables the following three main functionalities related to the life cycle of a session with the newly defined IPv6 extension header support.

IPv6 Extension Header Pre-Process (Box1): A consumer can send out one of two types of request messages: a legacy IP packet without the newly defined IPv6 extension header support, or an IP packet with the newly defined IPv6 extension header support. Whether it has the IPv6 extension header with the content identifier or not is determined by the type of service (e.g., for

For achieving in-network caching in the current IP network, a content identifier is carried in the extension header of an IPv6 packet, which makes the IP packet content-aware, so routers can quickly process the packet without deep packet inspection to fetch the content message.

Harnessing the content identifier in a novel IPv6 extension header can help the current IP network be content-aware, which would benefit from ICN specific in-network caching policy. This approach can be implemented with good support of current IP network without making substantial changes.

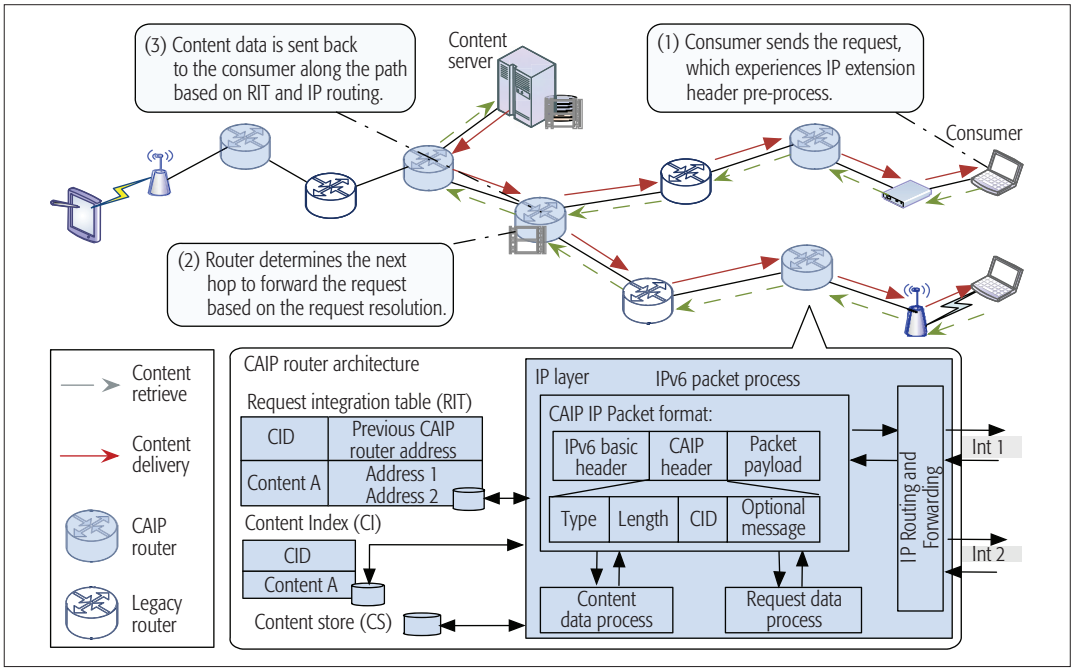

Figure 1. CAIP framework.

video service, this extension header is required, while for email service, it is not).

Request Resolution (Box2): For a packet with IP extension header support, the CAIP enabled router checks whether it has the requested content related to the identifier in the new defined extension header of the IP packet. If not, it will then determine the next hop toward the destination server based on IP routing. The request routing decision is then carried out hop by hop at each CAIP router until the content is reached.

Delivery (Box3): Content delivery is along the reverse of the request path based on forwarding states created during the associated request phase. In order to realize reliable transmission hop by hop, the packets should be reassembled first as in ICN. For sake of simplicity, in this article, we consider a packet carrying a small chunk without fragments.

Harnessing the content identifier in a novel IPv6 extension header can help the current IP network be content-aware, which would benefit from an ICN-specific in-network caching policy. This approach can be implemented with good support of the current IP network without making substantial changes. For data transmission of multimedia services, we define the IPv6 extension header format as shown in Fig. 1. The Type field indicates whether the IP packet is upstream (a request packet) or downstream (a content data packet),which should be further allocated by the Internet Assigned Numbers Authority (IANA). The Length field gives the variable length of the IP extension header in bytes. The CID field specifies the content identifier to identify the carried or requested content data.

Routers in CAIP can be classified as legacy routers, which do not support CAIP handling, and CAIP routers, which support CAIP handling. The deployment of CAIP routers is managed by the infrastructure providers (e.g., network operators) according to the consumers' geographical distribution, historical traffic load statistics, operators' will, and so on. From the perspective of performance effect, for consumer-concentrated areas and aggregated traffic areas, the more CAIP routers are deployed, the better the performance.

Each CAIP router contains three more content management related components than the legacy ones: a request integration table (RIT), a content index (CI), and a content store (CS). The forwarding model of a CAIP router is the same as the legacy IP router, which is based on the IP routing table and forwarding table. The CS is a temporary cache of chunks in accordance with cache strategies. In order to facilitate querying the presence of the requested content in the CS, we consider using a CI, which maps the chunk identifier to the cache position that points to the local CS. The RIT contains forwarding state information for each unacknowledged Interest packet, that is, maintaining the addresses of the former-hop neighbor routers from which each individual request comes, thereby ensuring that the data can be responded to correctly. A CAIP router basically receives one or more requests for the same content identifier forwarded from its neighboring routers, and the router only needs to forward it once so as to avoid duplication of the request forwarding. In other words, if the requested content identifier exists in the RIT, forwarding the subsequent request for the same content would be no longer performed in a CAIP router, and just one copy of content traverses the reverse path to this CAIP router, which in turn significantly reduces the traffic load.

Operations

The corresponding process that is required to be implemented within each CAIP router is best illustrated by the example in Fig. 2. The retrieval of content data involves a sequence of packet processing phases, in which the upstream and downstream processes are discussed separately.

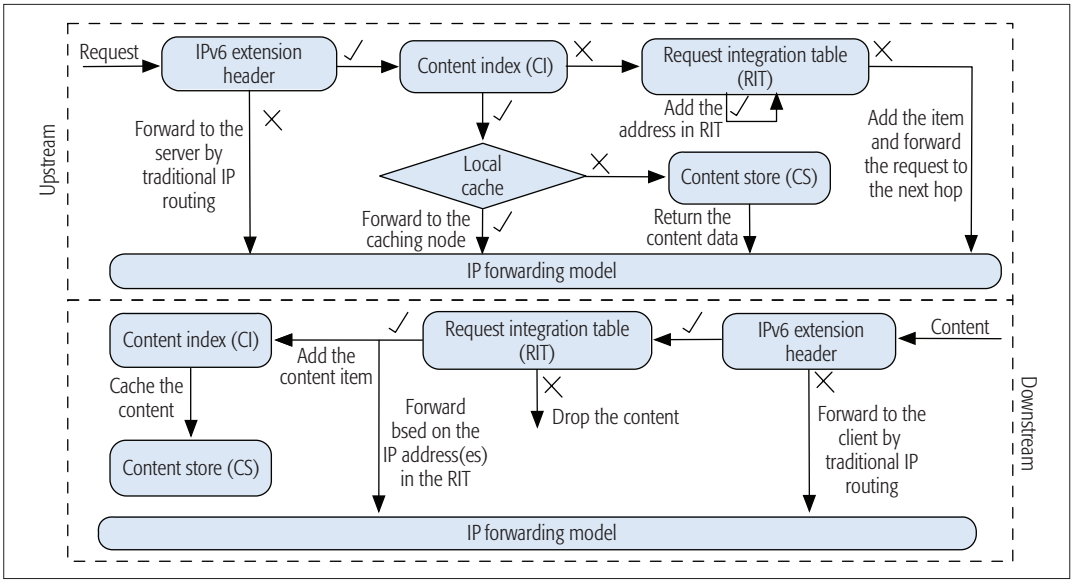

Figure 2. Forwarding process at a CAIP router.

Upstream Process: A consumer requests the service by issuing a request, which includes the content ID in the IP extension header. Meanwhile, the consumer should know the destination IP of the service and include it in the IP header. Legacy routers treat these types of request packets and legacy packets the same, and just forwards packets to the next hop according to the IP routing. For a CAIP router, each packet will experience a series of the following processes. The CAIP router first confirms whether there is a new defined extension header in the packet. For a request packet without the extension header, the router will simply forward it based on IP routing. Otherwise, the router will implement the following steps to reflect the benefit of in-network caching in CAIP. First, the CAIP router checks the CI for matching data. If there is an entry for the given content ID, the router will return the content data to the former CAIP router, whose IP address is recorded in the fourth field in the IP option of the received request packet. Otherwise, the router checks whether the content ID exists in its RIT. If a matching entry exists, it only needs to add the IP address of the former CAIP router in the matched RIT entry. If there is no matched entry, the router will add a new record in the RIT entry, which includes the previous CAIP router address. Then the CAIP router forwards the request to the next hop based on the IP routing. Moreover, the same operation is performed continuously in the path to the destination, until a suitable intermediate caching node is reached that has cached the corresponding content, or the destination server to retrieve the corresponding content in the worst case. The content server in CAIP should also be modified to process the IPv6 extension header to get the content identifier and then return the corresponding content data, which is also carried in packets with the new defined extension header.

Downstream Process: When receiving content packets with a CID, for a legacy router, it just forwards the packet to the next hop based on the IP routing. For a CAIP router, it will look up the RIT and forward the data packet to all nodes with the IP addresses listed in the entry with this content ID. As there are several legacy routers between the two neighboring CAIP routers, we can use an IP-in-IP tunnel mechanism (or other suitable means) to achieve direct transmission between the two neighboring CAIP routers. Subsequently, it wipes the matching RIT entry, and caches the content into the CS, and also updates the index table in the CI.

Content packets always go through the reserve path of a corresponding request packet. This means that each content packet can pass in reverse the CAIP routers the related request packet has passed. This mechanism ensures the feasibility of request aggregation, and the content data delivery can be made only one in some paths and separated to multiple ones to the downstream nodes according to RIT entries.

Neighboring CAIP Router Discovery and Cooperative Caching Policy

An important issue behind our architecture is deploying a simple but effective caching policy to make the best use of in-network caching and reduce redundancy. In this article, we do not adopt a complicated cooperative caching mechanism in our architecture, but only introduce a cache index exchange and one-hop neighbor cooperative caching mechanism named NCC.

At first, the CAIP router needs to discover its neighboring CAIP routers. In general, if two routers are at a distance of one hop physically, they can be defined as neighboring routers. However, there might be some legacy routers lying between the two neighboring CAIP routers in our architecture, as shown in Fig. 1. Thus, we cannot use the usual way of broadcasting to discover the neighboring router, which may induce large-scale network traffic. Hence, we need a simple but effective strategy to achieve neighboring router discovery.

As previously mentioned, we justify that due to the request integration, each content packet should reversely pass the CAIP routers through which the related request packets have already passed. The simple method is that the latter

The content server in CAIP should also be modified to process the IPv6 extension header to get the content identifier, and then returns the corresponding content data, which is also carried in packets with the new defined extension header.

We found that in this basic operation mechanism, the IP addresses of the previous CAIP routers have been recorded during the upstream and downstream. We thus can also use this mechanism to find neighboring routers with no need of much more extra overhead.

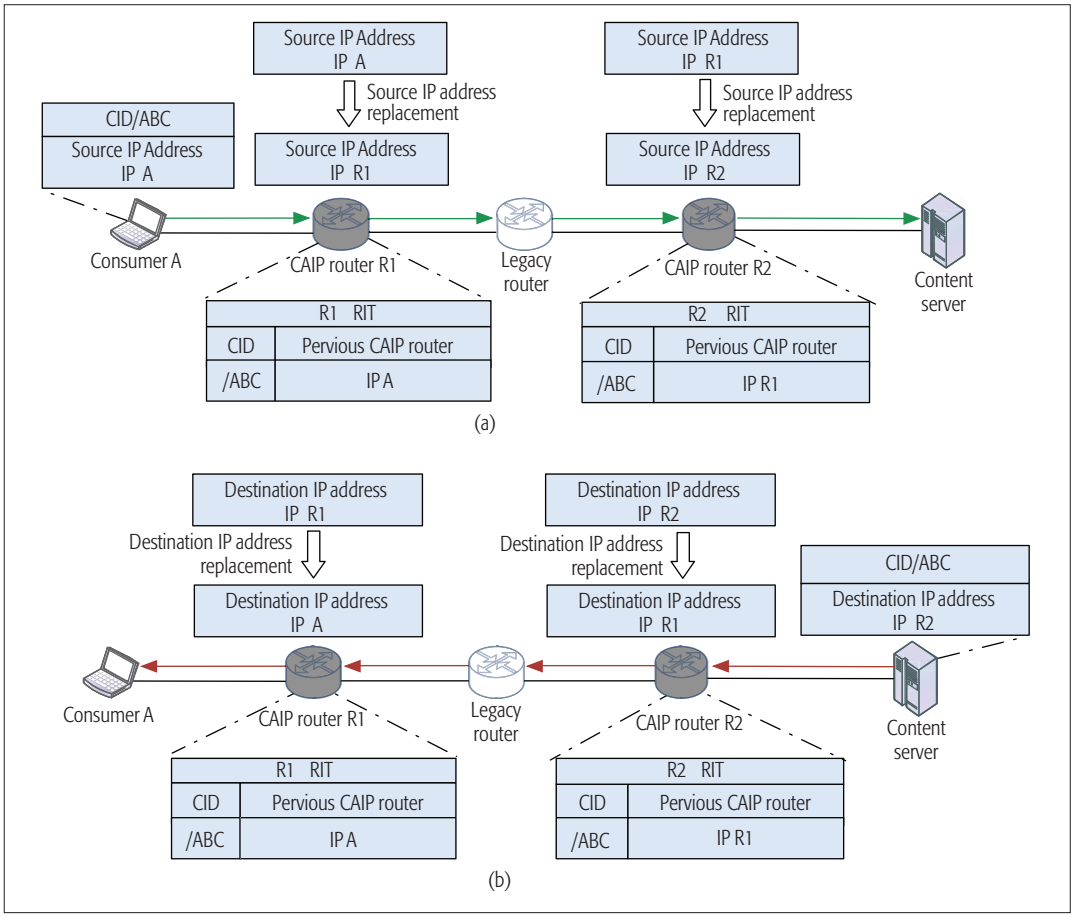

Figure 3. IP address replacement scheme: a) upstream IP address replacement; b) downstream IP address replacement.

CAIP router maintains IP addresses of the previous CAIP routers during the upstream process. However, there might be some legacy routers lying between the two neighboring CAIP routers. Therefore, we propose an IP address replacement scheme, shown in Fig. 3, to guarantee that content packets always go through the reverse path of the corresponding request packets. We have found that in this basic operation mechanism, the IP addresses of the previous CAIP routers have been recorded during the upstream and downstream. Thus, we can also use this mechanism to find neighboring routers with no need for much more extra overhead.

The IP address replacement scheme works as follows. As Fig. 3a shows, when the CAIP router R1 cache misses and the RIT in the CAIP router R1 has no corresponding item, the source IP address will be recorded into the RIT. Then R1 should let the next CAIP router or the content server know its IP address due to the request integration. Therefore, it replaces the source IP address with its own IP address. Then the CAIP router R2 that receives this request would know about the IP address of R1. Since there are no other CAIP routers between these two CAIP routers, we can define these two CAIP routers as neighbors. For the downstream, as shown in Fig. 3b, R2 can deliver the content to R1 by means of replacing the destination IP address with R1 based on RIT. This IP address replacement scheme not only guarantees the path symmetrically, but also achieves neighboring router discovery simultaneously.

When the neighbor relationship is established, routers are still not aware of what contents neighboring routers have cached when making caching decisions. To eliminate this unawareness, each CAIP router exchanges its own cache index with its neighbors. With neighbors' cache indices and their own cache indices, nearby routers can implicitly cooperate to serve each other's requests. NCC is a real-time policy. When receiving a new content object, the router should determine whether to cache it or not. The router first checks its neighbor table. If the new content object is already presented in the neighbor table, this indicates that at least one of its neighbors has cached the content, and the router will not cache it again. Otherwise, the router adds the new content object to its own CS.

Furthermore, other cooperative caching policies for ICN [9, 10] can also be further considered for our proposed architecture. We investigate simpler but more effective cooperative caching mechanisms and give a detailed performance evaluation.

Extension: Per-Chunk Caching

In the vast majority of ICN studies, due to the limitation of the underlying maximum transmission unit (MTU), a large chunk must be fragmented to several fragments and re-assembled at each router. For the re-assembling operation, all the frag-

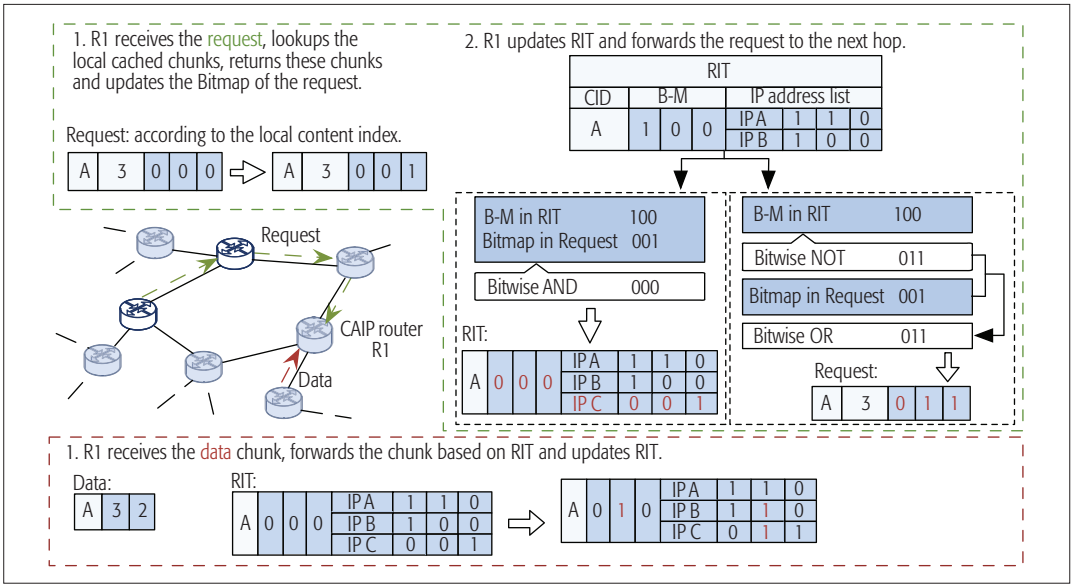

Figure 4. Per-chunk caching.

ments must be gathered, which leads to spending much time on gathering. Small size chunks can avoid being fragmented into too many fragments and are flexible for being cached in the router, but the data consumer has to send Interest packets frequently, especially for high definition video delivery. Therefore, we adopt small chunk size for transmission, and group multiple sequential chunks as a big chunk (named Content).

Here, we further describe how we use a bitmap structure in the defined IP extension header to reduce the number of request packets. We update the CID field in the request packet to the triple <CID, bitmap length, bitmap> and update the CID field in the data packet to the triple <CID, total number of chunks, Chunk No.>. Each content consists of multiple sequential chunks, which are identified by sequential sequence numbers as 1, 2, ... Each request packet is embedded with a CID with an optional bitmap structure, which can be utilized to request all or part of the whole content. The bitmap length indicates the number of chunks in a content that the user wants to request. In an *n*-bit bitmap, each bit represents a chunk and sets to 0 or 1, where 1 indicates that the chunk with the corresponding sequence number has been received, and 0 indicates that the corresponding chunk has not yet been received and is still in request. RIT should also be modified in such a way. For each CID, there are multiple entries, where each value of "previous CAIP router address" is followed by a corresponding bitmap. We also define a new bitmap structure named B-M following each CID in RIT, which is the result of the bitwise operation of AND on all bitmaps corresponding to the same CID.

As shown in Fig. 4, during the upstream process, when each CAIP router along the path receives a request packet, it first checks whether it has the chunks corresponding to the bits of the bitmap with 0. (If some chunks are confirmed to be cached in one of the neighboring CAIP routers, it will redirect the request to fetch the chunks.)Then this router reversely provides the matched chunks and modifies the corresponding bits of the bitmap in the request to 1. If the updated bitmap is not full of 1s, the router stores it together with the source IP in the IP header of the packet (as the "the previous CAIP router" value) in RIT. Then the router further updates the bitmap in the packet by implementing the bitwise OR operation on the bitmap in the request and the result of the NOT operation of the corresponding B-M in RIT. The router further forwards the updated request packet to the next hop if there are still one or more 0s in the bitmap. Finally, the router updates B-M. During the downstream process, if when receiving a content data chunk, the router finds all the previous one-hop addresses with the specific CID and having 0 in the specific bit of the bitmap, it forwards the chunk to all these routers. The router then updates 0 to 1 in the above-mentioned bits and updates B-M. If there is a bitmap full of 1s, the corresponding IP address and bitmap can be removed from the specific CID.

Performance Evaluation

The performance of CAIP was evaluated using NS3. We use the BRITE topology generation tool to generate the test topology [11]. Based on Waxman's probability model [12], the topology consists of 100 routers (setting CAIP routers randomly), 20 end hosts, and 20 original servers. The data access pattern is Zipf distribution [13], which states that the relative probability of a request for the *i*th most popular content is proportional to $1/i^{\alpha}$ with the shape parameter α. As our work is focused on how to achieve in-network caching in an IP network and improve cache utilization, we mainly evaluate the performance through two metrics:

- Cache hit ratio, which is defined as $(N_a - N_s)/N_a$. N_a is total number of requests generated, and N_s is the number of server hits.
- Traffic load, which is calculated by $\Sigma_{all\ requests}(RequestContentSize \cdot HopCount)$. We use percentages on the vertical axis for this metric, which represents the traffic load ratio of one specific scheme to the scheme of no caching.

For each CID, there are multiple entries, where each value of "previous CAIP router address" is followed by a corresponding bitmap. We also define a new bitmap structure named "B-M" following each CID in RIT, which is the result of the bitwise operation of "AND" on all bitmaps corresponding to the same CID.

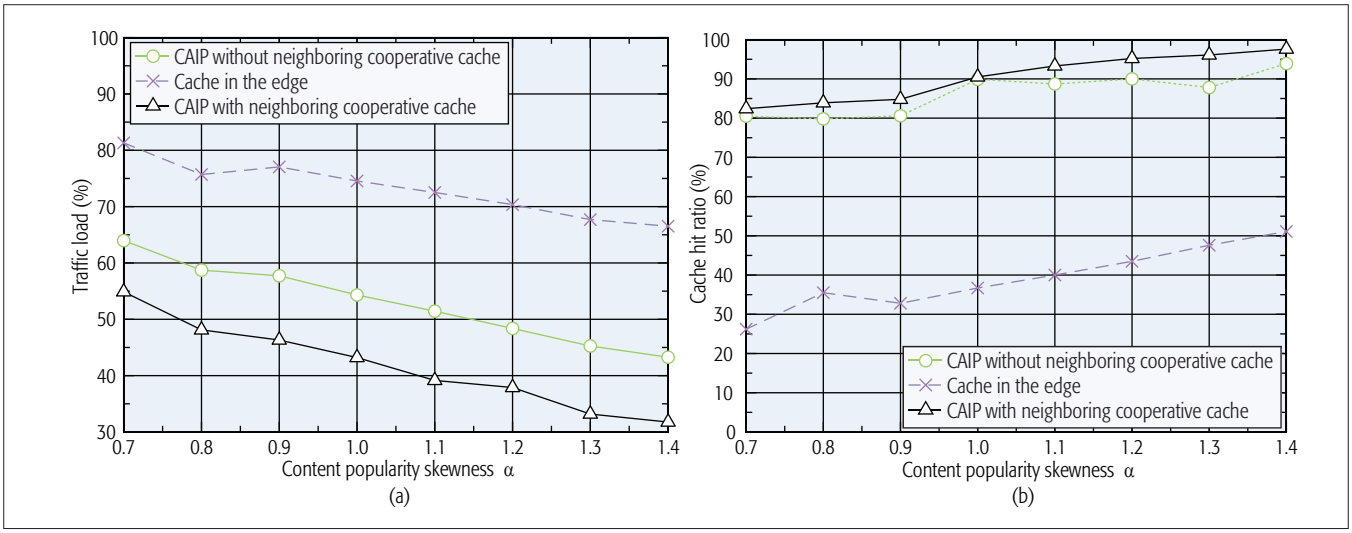

Figure 5. Performance evaluation of CAIP with neighboring cooperative cache, CAIP without neighboring cooperative cache, and cache in the edge: a) traffic load (the ratio of the experimental results to the results of using traditional IP without caching); b) cache hit ratio.

(a) Performance for content index lookup							
Size of content index (*K*)	8	16	32	64	128	256	512
Lookup throughput (MSPS)	55.6	47.6	38.5	32.2	26.3	20.4	14.5
(b) Performance for IPv6 RIB lookup							
IPv6 RIB size (1*1000)	5	10	15	20	25	30	35
Lookup throughput (MSPS)	3.87	3.84	3.78	3.72	3.68	3.66	3.63
(c) Performance for NDN FIB lookup							
NDN FIB size (1*100000)	1	2	3	4	5	6	7
Lookup throughput (MSPS)	1.06	0.96	0.93	0.80	0.69	0.66	0.65

Table 1. Performance evaluation of content index lookup, IPv6 RIB lookup, and NDN FIB lookup.

As shown in Fig. 5, CAIP makes performance remarkably better in cache hit rate and traffic load than that of caching in the edge and no caching (the case of traditional IP without caching). As CAIP achieves in-network cache in which each CAIP router can cache any passed content, consumers can retrieve the content from a closer router rather than that server. Moreover, with the popularity skewness α increasing, more requests are associated with the popular contents that have been cached already, so CAIP can better improve the overall network performance. Moreover, CAIP with neighboring cooperative cache policy can reduce the content redundancy and improve the utilization of cached contents, so it can further decrease the total traffic load, as shown in Fig. 5a, and improve the cache hit ratio, as shown in Fig. 5b.

Furthermore, we conduct an experiment to evaluate the performance of our CAIP's forwarding engine. We have implemented our design of the CAIP router's process engine via programming Linux Kernel 4.4.0-51-generic on a GB-BSi-7HA-6500 mini-pc platform, which running OS Linux 16.04. The platform hardware configurations are CPUs with Intel Core i7-6500U and 32 GB of DDR4 2133. We first evaluated the CI lookup throughput by varying CI target sizes from 8K to 512K entries. The CI is implemented using a chained hash table of size 512K. The size of a queried content label is about 20–60 bytes. Table 1a shows the lookup throughput by increasing the CI size. Then we compare our CAIP IPv6 forwarding engine with the NDN forwarding engine. According to the BGP Routing Table Analysis Report (https://bgp.potaroo.net/, accessed Mar. 25, 2017), we collect several real IPv6 Border Gateway Protocol (BGP) tables of different sizes. The experimental results of IPv6 lookup speed on the real-life BGP tables are shown in Table 1b. The NDN forwarding engine is implemented by deploying an NDN Forwarding Daemon (NFD) [14] based on the same platform as CAIP's. We generate an NDN FIB (around 1 million) from a URL dataset named URLblacklist (http://urlblacklist.com/, accessed Mar. 25, 2017). Name traces are generated to mimic the content labels by appending randomly generated directory paths to name prefixes in the FIB as the method in [15]. Table 1c shows the lookup speed as FIB size grows.

To put the numbers in context, the line rate of forwarding translates to 1.488 Mpackets/s with 64-byte frames on a 1 Gb/s link. We can see that the CI lookup speed is 14.5 MSPS (million searches per second) when the load factor (size of inserted CI/size of hash table) is 1. The CI search has only a little effect on the overall forwarding engine. According to the BGP Routing Table Analysis Report, the size of BGP forwarding table entries is about 35,000 in 2017. According to our IPv6 routing lookup experiment, the lookup throughput is about 3.63 MSPS when the size is about 35,000. Thus, the IPv6 routing lookup can satisfy the requirement of the line rate. But for the NDN FIB lookup, the lookup throughput is far lower than the IPv6 routing lookup, and the real size of FIB is far larger according to our experiment.

Conclusion

Along with the development of mobile Internet, the current network can hardly bear an explosive increase of global multimedia traffic. In order

to improve network resource utilization, content distribution and duplicated data utilization in IP networks are regarded as important issues. For the new proposed ICN, just as "far hydrolyze, not close thirsty," there is still a long way to go. In this article, based on the consideration of the forward compatibility, we propose a novel content-aware IP-based architecture, named CAIP, which can not only draw the advantage of in-network caching from ICN, but also guarantee the simplicity of the end-to-end model. CAIP enables the traditional IP network to have better vitality, and in turn results in a significant impact on network development.

Acknowledgment

This work is supported in part by the National Natural Science Foundation of China under Grant No. 61379129, the Key Research Program of the Chinese Academy of Sciences (CAS) under Grant No. ZDRW-KT-2016-2-5, Youth Innovation Promotion Association CAS under Grant No. 2016394, and the Fundamental Research Funds for the Central Universities.

References

[1] C. V. N. Index, "Forecast and Methodology, 2014–2019 White Paper," tech. rep., Cisco, 2015.
[2] Y. Cui *et al.*, "SDN-Based Big Data Caching in ISP Networks," *IEEE Trans. Big Data*, 2017; https://doi.org/10.1109/TBDATA.2017.2651901, accessed June 27, 2017.
[3] A. Venkataramani *et al.*, "MobilityFirst: A Mobility-Centric and Trustworthy Internet Architecture," *ACM SIGCOMM Comp. Commun. Review*, vol. 44, no. 3, 2014, pp. 74–80.
[4] B. Ahlgren *et al.*, "A Survey of Information-Centric Networking," *IEEE Commun. Mag.*, vol. 50, no. 7, July 2012, pp. 26–36.
[5] L. Zhang *et al.*, "Named Data Networking," *ACM SIGCOMM Comp. Commun. Review*, vol. 44, no. 3, 2014, pp. 66–73.
[6] T. Koponen *et al.*, "A Data-Oriented (And Beyond) Network Architecture," *ACM SIGCOMM Comp. Commun. Review*, vol. 37, no. 4. ACM, 2007, pp. 181–92.
[7] G. Carofiglio *et al.*, "From Content Delivery Today to Information Centric Networking," *Computer Networks*, vol. 57, no. 16, 2013, pp. 3116–27.
[8] A. Detti *et al.*, "CONET: A Content Centric Inter-Networking Architecture," *Proc. ACM SIGCOMM Wksp. Information-Centric Networking*, 2011, pp. 50–55.
[9] G. Zhang, Y. Li, and T. Lin, "Caching in Information Centric Networking: A Survey," *Computer Networks*, vol. 57, no. 16, 2013, pp. 3128–41.
[10] M. Zhang, H. Luo, and H. Zhang, "A Survey of Caching Mechanisms In Information-Centric Networking," *IEEE Commun. Surveys & Tutorials*, vol. 17, no. 3, 2015, pp. 1473–99.
[11] A. Medina *et al.*, "BRITE: An Approach to Universal Topology Generation," *Proc. IEEE MASCOTS 2001*, 2001, pp. 346–53.
[12] B. M. Waxman, "Routing of Multipoint Connections," *IEEE JSAC*, vol. 6, no. 9, 1988, pp. 1617–22.
[13] L. Breslau *et al.*, "Web Caching and Zipf-Like Distributions: Evidence and Implications," *Proc. IEEE INFOCOM 1999*, vol. 1, 1999, pp. 126-34.
[14] A. Afanasyev *et al.*, "NFD Developer's Guide," tech. rep. NDN-0021, rev. 6, NDN Project, 2016.
[15] H. Dai *et al.*, "BFAST: High-Speed and Memory-Efficient Approach for NDN Forwarding Engine," *IEEE/ACM Trans. Networking*, vol. 25, no. 2, 2017, pp. 1235-48.

Biographies

Kaiping Xue (kpxue@ustc.edu.cn) received his B.S. degree from the Department of Information Security, University of Science and Technology of China (USTC), Hefei, in 2003 and received his Ph.D. degree from the Department of Electronic Engineering and Information Science (EEIS), USTC, in 2007. Currently, he is an associate professor in the Department of Information Security and Department of Electrical Engineering and Information Science (EEIS), USTC. His research interests include next-generation Internet, distributed networks, and network security.

Tingting Hu (hutingt2@mail.ustc.edu.cn) received her M.S. degree from the Department of EEIS, USTC, in 2017. Her research interests include future Internet architecture design and network functions vitalization.

Xiang Zhang (mm1201@mail.ustc.edu.cn) received his B.S. degree from the Department of Information Security, USTC in July 2015. He is currently a graduate student in communication and information systems in the Department of EEIS, USTC. His research interests include next-generation Internet and network security.

Peilin Hong (plhong@ustc.edu.cn) received her B.S. and M.S. degrees from the Department of EEIS, USTC, in 1983 and 1986. Currently, she is a professor in the Department of EEIS, USTC. Her research interests include next-generation Internet, policy control, IP QoS, and information security. She has published 2 books and over 150 academic papers in several journals and conference proceedings.

David S. L. Wei (wei@dsm.fordham.edu) received his Ph.D. degree in computer and information science from the University of Pennsylvania in 1991. He is currently a professor in the Computer and Information Science Department at Fordham University. He has authored and co-authored more than 100 technical papers in various archival journals and conference proceedings. Currently, his research interests include cloud computing, big data, IoT, and cognitive radio networks.

Feng Wu (fengwu@ustc.edu.cn) received his M.S. and Ph.D. degrees in computer science from the Harbin Institute of Technology in 1996 and 1999, respectively. He is currently a professor with USTC, and the Dean of the School of Information Science and Technology. He has authored or coauthored more than 200 high-quality papers. His research interests include image and video compression, media communication, and media analysis and synthesis.

CAIP makes performance remarkably better in cache hit rate and traffic load than that of caching in the edge and no caching (the case of traditional IP without caching). As CAIP achieves in-network cache in which each CAIP router can cache any passed content, consumers can retrieve the content from a closer router rather that the server.