한국시뮬레이션학회 논문지
Vol. 26, No. 1, pp. 21-38 (2017. 3)
http://doi.org/10.9709/JKSS.2017.26.1.021
ISSN 1225-5904

향상된 보안 및 적용 가능성을 위한 컨텐츠 중심 네트워킹(CCN)의 새로운 아키텍처 연구

아쉬스 샤르마 · 김윤선 †

A novel architecture of CCN for better security and applicability

Aashis Sharma · Yun Seon Kim†

ABSTRACT

Information Centric Networking is changing the way how content is being transmitted. The shift from IP and host based networking towards content based networking scenario is growing day by day. Many researches have been done about different frameworks of ICN. Caching is an important part of ICN and many researchers have also proposed different ways for caching the data. With caching of data in intermediate devices like the network devices as well the user devices in some cases, the issue of content security as well as the role of the content producer becomes a major concern. A modified ICN architecture based on the current Content Centric Networking (CCN) model is presented in the paper. The architecture mainly focuses on involving the content producer in content delivery in the real time. The proposed architecture provides better security aspects for the CCN architecture. Apart from security the paper will also consider the issue of applicability of CCN architecture to replace the TCP/IP based architecture. The efficiency of the proposed architecture is compared with the previous CCN architecture based on the response time for a content delivery which shows very comparable level of efficiency. The paper than analyzes different beneficial aspects of the proposed architecture over the current architecture.

Key words : Information Centric Networking, Content Centric Networking, security, applicability, content producer

요 약

정보 중심 네트워킹 (Information Centric Networking)은 콘텐츠 전송 방식을 바꾸고 있다. IP 및 호스트 기반 네트워킹에서 콘텐츠 기반 네트워킹 시나리오로의 전환이 날로 증가하고 있고 ICN의 다른 프레임워크에 대한 많은 연구가 수행되었다. 캐싱은 ICN의 중요한 부분이며 많은 연구자들은 데이터 캐싱을 위한 다양한 방법을 제안했다. 네트워크 장치와 같은 중간 장치에 데이터를 캐싱하고 사용자 장치를 경우에 따라 사용하면 콘텐츠 제작자의 역할뿐 아니라 콘텐츠 보안 문제가 큰 관심사가 된다. 현재 CCN (Content Centric Networking) 모델을 기반으로 하는 수정된 ICN의 새로운 아키텍처가 이 연구에서 제시되었다. 이 아키텍처는 주로 실시간으로 콘텐츠 제공에 콘텐츠 제작자를 참여시키는데 중점을 둔다. 제안된 새 아키텍처는 CCN 아키텍처에 대한 보다 나은 보안 측면을 제공하고 보안 외에도 TCP / IP 기반 아키텍처를 대체하기 위해 CCN 아키텍처의 적용 가능성 문제를 고려하고 있다. 제안된 새 아키텍처의 효율성은 기존의 아키텍처와 매우 유사한 수준의 효율성을 보여주고 컨텐트 전달에 대한 응답 시간을 기반으로 이전 CCN 아키텍처와 비교하였다. 이 논문에서 현재 아키텍처에 비해 제안된 새로운 아키텍처의 다양한 이점을 제시한다.

주요어 : 정보중심네크워킹, 컨텐츠중심네트워킹, 보안, 적용가능성, 컨텐츠제작자

Received: 23 November 2016, **Revised**: 17 January 2017, **Accepted**: 27 January 2017

† **Corresponding Author**: Yun Seon Kim
E-mail: sean0831@handong.edu
Handong Global University,
Graduate School of Global Development & Entrepreneurship

1. Introduction

The shift of the use of the internet from communication to content delivery platform has given rise to the development of Information Centric

Networking. ICN has been placed as the best replacement for the IP based the internet or the current internet we know[1]. Content Delivery Networking (CDN) has been currently providing better solutions for the transmission of video based data but with the amount of growing data CDN will also be not able to provide the best alternative to handle the internet traffic. With the evolution of social media and Video on demand platforms the users are more focused on the final content they get instead of the source or the platform they are getting the content from. This has made the current internet more and more complex so the host based TCP/IP Internet is becoming too heavy to offer the best performance to the end users.

The scientific fraternity have been divided over the way to handle the change and the growth in the internet traffic[2]. Some researchers still look for the ways to improve the current internet architecture such that it will be best suitable for the new changes in behavior whereas many researchers have advocated for a switch to a next generation and unconstrained internet architecture. The interest towards the development of new architecture for the internet can be seen by the big number of researches done towards the development[3].

ICN has evolved as an alternative to the current internet architecture as it is more focused on delivery of the content instead of establishing a communication between hosts [4]. ICN also tries to remove the complexities of the current internet as its networking activities are all based in the named contents. ICN mainly focuses on routing the contents among its nodes to deliver a data instead of just routing the requests from the user to the requested server. It is a receiver driven architecture where the users express their interests for a given content and the network then takes the responsibility to find the content based on its name and deliver it back to the user based on the reverse route.

In information centric networking, the networking is done on the basis of the content and these content has to have proper naming techniques such that they could easily transfer within the provided network. Earlier in the host centric networking the host and the server had their specific naming which would help them transfer data after a connection is established. But in the content centric networking, the content will not be host dependent and thus can come in from any host within the network which means the content should have a specific name for it to be identified. Each information item should be uniquely identified and authenticated without being associated to a specific host and this ultimately would help the in-network caching of the content which is another feature of ICN[5].

ICN has evolved as a promising replacement for the current information architecture and has gathered a lot of attention. The establishment of Information-Centric Networking Research Group (ICNRG) within the Internet Research Task Force (ITRF) in 2012[3] shows the amount of attention gathered by the ICN architecture. The concept of ICN was initiated when ICN like design was described by Gritter and Cheriton in the TRIAD project in the early 2000[6] also when in 2002 Baccala expressed that the future internet should be shifted from point to point communication to delivery of named object[7]. But now after more than a decade after ICN like architecture have been first coined there are many researches and similar architecture proposed for the implementation of ICN[8][9]. Data Oriented Network Architecture (DONA)[10], Network of Information (NetInf)[11], Publish-Subscribe Internet Routing Paradigm (PSIRP)[12] and Content Centric Networking (CCN)[13] are some of the architectures which have been gaining a lot of interest from researchers. ICN is mainly based on addressing each content and delivering the contents based on the name of the contents from either the content producer or through the cached content within the network. The main backbones of ICN lies in i) naming the content, ii) forwarding or routing the content, iii) caching the content and iv) providing security to the content.

Caching being a main feature of ICN makes contents to be stored in between the network for a certain time. The duration for a content to be stored in cache depends on different criteria of the caching policies and the content replacement polices. So, if a content is stored in cache then then the request can be

delivered by the cache itself and the content producer will have nominal role in content delivery. As the content popularity in the internet follows the Zipf Law which means that the popular content might all be delivered through the cache and only unpopular content requests their content producers regularly. Although this decreases the load in the content producers, but removing the content producer from the content delivery scenario might bring in other issues. This feature could be exploited by some content producers who would inject the content into the network and through the caching the content could be stored in the network even after the content producer is out of the network. Although there are provisions for the replacement and removal policies of the cached content, but this could be exploited by the users by keeping the content alive in the cache though limited hit. This would create a scenario for a content to be alive even after the producer is non-existent and this can be used for unlawful purpose as well. Also, the un-involvement of content producers in content delivery would prevent the producers from getting real-time about the incoming requests. The producers could not get the real-time data of the content that are delivered through the in-network cached nodes.

The paper here thus proposes an architecture which would incorporate the producer in the content delivery. The proposed architecture would incorporate the role of content producer in each of the content delivery. The paper would then compare the benefits of the new architecture over the previous proposed architecture. The proposed architecture would be compared with the previous architecture over the response time for a simulation environment. The following section would first analyze the previous literature about the CCN architecture and would analyze the problems which is due to the un-involvement of the content producer. The later section the paper would present the results of simulation which would show the differences in the current and previous architecture.

2. Literature Review

2.1 Information Centric Networking

ICN as the name suggests is a networking architecture which is mainly centered around information instead of hosts. ICN architectures are mainly focused on naming contents and routing them in different ways for efficient content delivery. Many ICN architectures have been proposed and still being researched. Based on the nature of communication Tourani et al. [14] has categorized some common CCN architectures into two models as i) consumer-driven and ii) publish-subscribe. In the consumer driven architectures, the communication is initiated when a user sends a request for a data/content to the network and then the publisher sends the data back or in case of caching the cache sends the reply. Whereas in the publish-subscribe architectures the request is initiated by the content publisher as a advertisements of their data to the interested subscribers and then the subscribers sends the request to get those published content through the network.

2.2 Content Centric Networking (CCN)

CCN architecture is one of the most successful ICN architecture. The CCN architecture was proposed by Palo Alto Research Center (PARC) in a seminal publication published in 2009[15]. CCN focuses on mainly naming each content and forwarding interests based on the named contents. Other ICN architectures mainly lie in the application layer whereas CCN changes the network protocol layer from TCP/IP to named contents. But CCN does not change the hourglass model[16] of the IP bases network but the thin waist of the TCP/IP layer is replaced by content chunks. This architecture of CCN is free from middleware which were required to map the relationships between application's content-centered model and the Internet's address-centered model. So, the implementation of CCN is simpler and this results in increased communication efficiencies[40].

2.2.1 CCN Architecture

The CCN communications starts with the initiation of user for the request of a content. As ICN has mainly two message types: Interest and Content object. The user first sends an interest message with content name to the network. A network interface in CCN is usually represented by Face which might either be an interface to the network or to an application being executed on a CCN node. Caching is one of the major feature of CCN architecture. So, once the interest message reaches a node, it may reply to the interest with the content object if it has the copy of that named content. The cache at each node of the CCN network is named as Content Store (CS). So as soon as the interest reaches the node it checks CS for the requested chunk of content. If the CS has content stored, then it replies with the content else the interest is forwarded to next node.

The forwarding or routing of interest is based on the Forward Information Base (FIB) and Pending Interest Table (PIT). All the requests coming into a node if could not be replied from the CS then gets forwarded to the next node but before it is forwarded to the next node an entry for the interest is made on the PIT. The FIB consists of the forwarding entries for the interfaces connected to the node and based on these FIB entries the interest is send to the next node. The forwarding strategies are based on the many forwarding architectures like multipath forwarding or shortest path forwarding or the random forwarding[34]. So, the interest moves on towards the producer until it finds the content either in the CS of the nodes or in the content producer. The content travels back based on the reverse path per the PIT entries. So, it goes back to all the routers that has the PIT entries for the requested content. Also, the content is then cached on the intermediate nodes such that when the same content is requested it can be replied by the CS itself. The caching of contents on the nodes are based on different criteria as the cache size is small compared to the amount of content passing through the nodes. The following flowchart explains the working procedure of a CCN architecture. The figure below further describes the caching scenario of CCN and the role of PIT and FIB for multiple user scenario.

CCN architecture consists of mainly four basic components i) naming of the content ii) routing/ forwarding of the requests, iii) caching of the content in network nodes and iv) security. Many researches have been done in CCN revolving around these components and many ideas have been presented. Each of these components will be discussed briefly giving an idea of the way a CCN architecture works.

2.2.2 Naming

ICN mainly has two naming structures: the hierarchical naming and the flat names [17]. The ICN names are human readable not like the IP address and are used for identifying and routing the requests. This naming makes it easier for users to relate their real-world requirement with the name of the content they want. Naming not only provides unique identity of the content but also includes provides pertinent, usability, scalability and security to the content itself [18] [19]. The flat naming structure is very widely used but it is very helpful in the cases of the distributed hash tables lookups[20]. CCN uses the hierarchical naming system which clearly gives the user idea about what content it is requesting. In CCN each content is divided into small chunks which are usually identified uniquely. The combination of these chunks thus form a complete content. The below figure shows the naming structure in CCN.

As the naming structure shows each name gives a clear idea of what content the user is looking for. Each name is unique and the complete set of chunks creates a whole content. The chunk numbers are usually assigned using two numbering schemes: sequential numbering and random numbering [21]. In figure 11 the final part of the content name is the chunk number denoted by s such that s {001,002,003…}. Once the client gets the name of the first packet then by increasing the value of s the following chunks could be obtained which would generate the full content. The sequential scheme is easy to implement but is vulnerable to attacks related to traffic analysis[22]. In random numbering the value of s for first chunk is 001

but the next chunk cannot be obtained by increasing the value of s instead each chunk carries the value of s for the next chunk. This scheme limits the ability to generate requests for all the chunks at the same time. Similarly, the segment v1 denotes the version which means the user could download the same content with different versions. The version could denote quality of video or different production version of the same content.

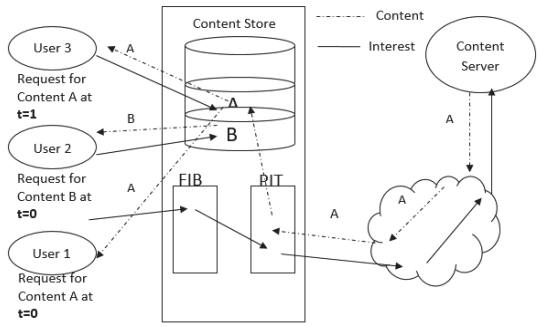

Fig. 1. CCN operation in multiple user's requests

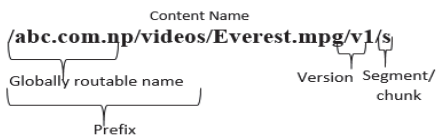

Fig. 2. Content Naming in CCN

2.2.3 Routing/Forwarding

Routing is CCN is forwarding a content request to the nearest content store which could either be a caching node or the content producer itself. As discussed earlier in CCN architecture FIB is the basis for all the routing activities in CCN. FIB replaces the routing table of the TCP/IP architecture. Within this framework of routing variety of effective routing alternatives for CCN is being researched. Any routing scheme that works with the current TCP/IP model would work in CCN because CCN's forwarding model is a strict superset of the TCP/IP model but with fewer restrictions[4].

A FIB contains the details of the faces to forward the content. A face is generally the same as the concept of interface. A face may be a connection to a network or directly to an application. When a consumer sends an interest packet to a face of a router/node then they are routed towards the either the content publisher through the longest prefix match of the content name in the FIB table. Similarly, while the reply for the interest request returns as a content request it follows the reverse path which is guided by the PIT. The overall running process of CCN routing follows the two-phase approach of routing similar like that of the TCP/IP.

• The first phase is the FIB population phase. In this phase the FIB is filled with entries pointing to locations where the content resides. For this the Interior Gateway Protocol (IGP) type protocol such as the Open Shortest Path First (OSPF) routing protocol is used[4].

• The second phase is the forwarding phase where the interest is forwarded to one or more faces that have prefix entries in the FIB and has the longest prefix match with the interest.

As shown in figure 1 the FIB at each node is populated with the prefixes and the interest is forwarded based on the FIB. For example, the user at figure 2 requests a content "/abc.com.np/Videos/ Annapurna/a.mpg/v1/001" then it is forwarded towards node B through Face B FB according to the longest prefix matching. Similarly, if the user requests a content "/abc.com.np/Videos/Everest/A.mpg/v1/001" then the FIB at C shows that it has more than one options.

2.2.4 Caching

Caching in CCN is considered as an integral part of its architecture. Caching is an important part of CCN because both content caching and the request routing are operated in the same network layer which makes the caching task more important as well as complicated [23]. Although CCN evolved mainly as naming content and routing them based on their names caching added another feature to CCN such that the contents could be kept anywhere within the network and delivered to the users. Any network nodes along the path from the requester to the producer holding the corresponding content can directly satisfy the end-user and consume

the request[24]. Meanwhile the nodes without the content in their cache can cache the content while they pass from their node as per their decision. Caching mainly consists of three main issues: What to cache? Where to cache and How to cache? For where to cache there are mainly two alternatives, i) in-network caching ii) off-path caching. Many researchers have identified the pros and cons of using both the caching. But CCN architecture prefers to use in-network caching due to its multiple benefits[25]. What to cache is another issue as the caching and the replacement of the cache should be done at line speed[26]. Also, the cache size of the nodes cannot be of very big limiting them to cache everything that passes through the node. So, proper decision making should be done to identify what to cache and what not cache. The issue of how to cache brings in the need for the best way to cache maximum content making the best uses of the available resources.

2.2.5 Security

The CCN security revolves around the content based security. All CCN content is authenticated with digital signatures and private content is protected with encryption[9]. Each CCN data packet is validated by a self-contained signature covering the name, the content and some data useful for using the signature verification. This gives CCN packet to be authenticable anywhere in the network. The nodes in the network path can verify that a name-content binding is signed by a particular key which it could obtain from the key locator included in the message. Effective ICN security means includes key distribution, encryption, digital signing, stale timing, binding, key has computed using trees, etc.[23] [27]. Another major security factor of CCN as well as the overall ICN architecture is the fact that no information is given unless a request is received to get that particular information. Despite the security features CCN security has other major dimensions and many researches have been regarding those. One of the major concern of security in CCN is due to the caching on content. As the content should itself be sufficient enough in CCN to take all the decisions, incorporating proper security to these contents is a major task. Some of the common challenges with security in CCN and the proposed solutions regarding them will be discussed in the next section.

2.3 Identified issues in CCN

2.3.1 Cache Pollotion

Caching is an effective tool of CCN because the popularity of the content in the internet follows the Zipf distribution[28], i.e. a small number of popular content are requested frequently whereas the rest of the content are sporadic. So even if a small content is cached the cache hit ratio would be higher compared to the total content passing through the node. But this feature could be exploited by attackers by adding unpopular content to the cache and hampering the cache distribution. Two common caches of cache pollution attacks identified are: locality disruption and false locality[14]. In locality disruption attack, the attacked continuously requests unpopular contents to disrupt the locality of the cache by adding new popular contents continuously. The false locality attack aims to completely change the popularity distribution of the local cache by requesting a set of unpopular contents from within the universe of contents. So, the cache will now have a new popularity distribution such that it will not cache the real popular contents as it is forced to cache the unpopular contents.

2.3.2 Timing attack

Timing attack is usually performed through a common node among the attacker and the victim. In timing attack the attacker probes content objects in the shared router to see what contents have been cached. It uses the difference in the time difference between a cache hit and cache miss to identify what contents have been cached. A lower time means that it's a cache hit and it gives an information that the same content has been requested by a user in the same local network through the same node. An attacker can use the time difference to identify whether the cache hit occurs in the shared router or some other router along the path of the request.

2.3.3 Un-Traceability

CCN also don't have any endpoint identifiers. As the messages are routed in a hob by hop basis, only the previous hop is known when a message is received. Also, the data trail is removed as soon as the data is sent back to the user. So, tracing back to users is not possible simply through the CCN architecture. It is tough for law enforcement agencies to monitor for anomalies as they cannot properly identify the victim by monitoring the in-path nodes. Whereas monitoring the whole path would be a tough job. Some researchers also proposed solutions where the users are forced to individually sign their requests which would help to identify user from a request. But digitally signed requests would increase the overall system overhead and d its implementation is questionable.

2.3.4 Individualization of data

As in CCN it is not necessary for a request to reach the actual provider, instead it could be replied through any caches along the path of the request. This means the user could access the data that has been fetched by some other user along the same path. So, the question also arises that how effective would it be for the user to be supplied with something that has been generated for a different user. Research by Sugiyama et al.[29] has concluded that more and more content from the web are being generated as per the individual user which means for each user the generated content might be different in some aspects. So, in such cases data generated for one user may be not exactly be same for data generated for a different user. CCN in this case has not much options left. Although the classification of content into static and dynamic has created an environment where only content which are not usually changed are categorized as static content and only these contents are cached. But the dynamicity of the internet is growing day by day with each content being sent as per individual user which means that the data to be cached will be decreased and thus limiting the proper application of CCN. As claimed by Kaushik[30] tomorrow's internet is going be a world of hyper-personalized tribes. Thus, CCN architecture must have some features such that it would help to deal the personalization of the content.

2.3.5 Statistics Infrastructure

Data regarding anything provides some valuable information. The data from the user access of the internet gives great information regarding the trend of users and their behavior. Content providers are interested to know the way their data is being accessed by the users. They want to have proper statistics regarding the population of users accessing their content and their behavior. This kind of information is essential to determine pricing of advertisements[31] and to optimize the services delivered. CCN does not provide the content publishers with the identifiers regarding the user which limits them to collect statistics about their user behavior. Also, not all the request for contents reach the content producers as the requests could be replied by the caches in the in-network caches. This prevents the content publishers to keep proper statistic regarding the actual amount of request they get for their content.

3. Proposed Architecture

The proposed architecture of CCN is similar to the basic CCN architecture in many aspects. It varies with the current architecture in the way a content is fetched. Although the theme of CCN is kept but content fetching is performed in a bit different way. The request starts with the generation of an interest request by the user to the nearest node through its connected face. The node checks for the requested content in the CS and if the content is not found, it forwards the request to next node based on the entry at FIB. Till here the working remains the same as the current architecture. But if the content is found in the CS then at this case the request is not replied by the node itself. As we want a content to be always associated with the content publisher we do not give the right for the CS to deliver the content. But making the content reply the content itself changes the whole idea of the CCN. So, the interest even in the case of content match goes to

the next node but at this case but with a change in the interest message.

The first node where the cache hit occurs, flags the interest message as cache hit and forward this message to the next node. Now on the following nodes the flag is checked to identify if the interest is cache hit or not. If the interest is cache hit, then at this case the node just forwards interest message to the next node. If not, then it looks for its own CS to find the same process repeats till the interest reaches the content publisher. So, although the content is found in the local cache the request travels till the content publisher. The content publisher now can get all the contents that has been sent for content it has published. The forwarding of request also involves one more step. At each node when an interest message in received, an entry for it is made on the PIT. This must be done even for cache hit as the request has to come back uses the request in PIT in reply to the message.

For the path back, the request again follows the same phenomenon as the general architecture of following the reverse path through the entries at the PIT.

The content producer should make some decisions before it sends back the response. If the request is cache hit, then the content has a choice to forward the interest request through the same path such that the content is delivered from the cache through the cache hit router. But at some cases the content producer might want to send a content by itself instead of using the content in the cache. In this case the content producer has the option to do that and thus a new content message is sent back in response of the interest request. In the case of no cache hit in the interest request the content is sent back as normal procedure. Whereas in the case of cache hit and the content producer does not want to send a new content then it forwards back the interest message back through the same node but here the producer has an option to add some meta-info to the interest message. After this the content message is sent back to the previous node.

The reverse path of the request might be either a interest request itself or a content message. If a content message is returned from the server, then the reverse request remains the same as in the current CCN architecture. It gets either cached in the in-network routes or just gets delivered to the requesting user. The flow of the request gets a bit longer than that in the CCN architecture where a cache hitting message might just get replied form the in-network cache. Although there is not much change in the proposed architecture, some of the new aspects of the proposed architecture is discussed further.

3.1 Forwarding/Routing

The forwarding scheme of CCN is completely based on the strategies employed in the router. It is based on the FIB entries in the routers. The proposed architecture plans to make no change on the FIB entries so the forwarding strategies would remain the same. Once the interest enters it is forwarded based on the name, FIB entries and other routing strategies for either an interest request, or a cache hit interest request

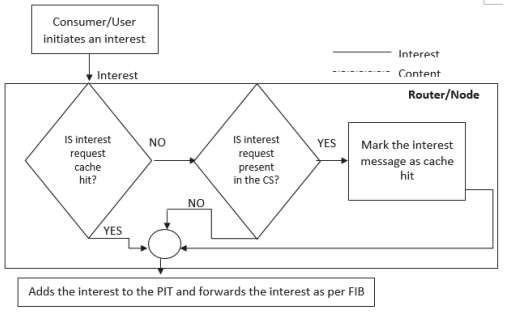

Fig. 3. Flowchart of an interest message moving through a router/node

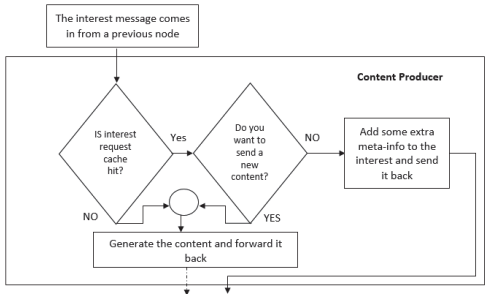

Fig. 4. Flowchart of an interest message inside a content producer

or a content message. For the reverse path of the messages, the PIT entries are used. In the proposed architecture, each of the interest entering the router will have a PIT entry thus the reverse path of the message would be easily identified for both the interest message and the content message.

For the case of interest duplication, i.e. when an interest for same content is entering a node when an entry for that content is already in the PIT: a new entry will be made on the PIT instead of interest aggregation. As in the proposed architecture all the requests should reach the producer so interest aggregation is not the option. Even if a cache hit takes place then the interest aggregation will be done on the PIT entry. Making all the entries might increase the size of the PIT. But adopting the encoding-based idea proposed by Dia et al could help decrease the PIT size and increase the access frequency requirement.

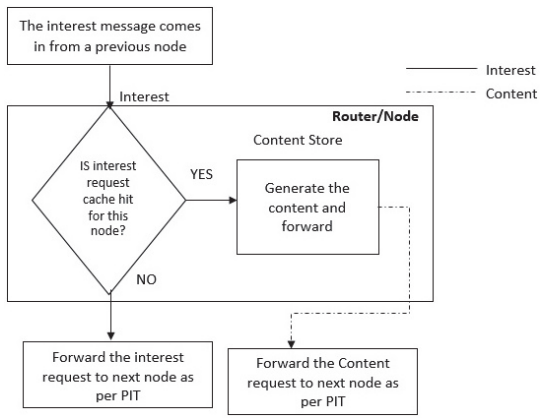

Fig. 5. Flowchart of interest message back from the producer

3.2 Caching

The proposed architecture follows similar caching scenarios as the CCN architecture. Caching being one of the main aspect of CCN, there has been many researches done for the development of caching technologies. Caching remains the same as earlier and in-network caching as considered by many researchers is considered best alternative for caching contents.

4. Research Methodology

The paper has presented a change in the current architecture of CCN. Although the proposed architecture is different is some ways, most of the architectural framework is same. CCN/NDN has many simulators designed which could run a simulation environment for the basic CCN architecture. CCnx[32], ndnSIM[33], ccnSim[34] and OMNet++[35] are the most common examples of CCN simulators. Most of these simulators are open source and is written in C++. For this research, we chose to use ndnSIM as the simulation platform. It is based on ns-3 network simulator framework. It provides all the required things to evaluate the transport behavior of networking in CCN. The simulator is implemented in a modular fashion, using separate C++ classes to model behavior of each network-layer entity.

Although we propose a new architecture for the communication in CCN based environment, we are going to use to the current CCN architecture to run our simulations. We are going to use response time as our point of comparison. As one of the major benefits of using ICN or CCN is fast communication or lesser use of bandwidth which is overall improve the network performance[36]. We will show the comparison of the response time requested data in the current CCN architecture and the proposed CCN architecture. As we do not have simulation environment for the proposed CCN architecture, we will calculate the response time for the proposed architecture mathematically using the results from ndnSIM simulations. We will also use the response time to verify security from some of the defined security threats in the CCN architecture.

4.1 Simulation Environment

Using the ndnSIM simulation environments gives the privilege to use the predefined network topologies provided with the simulation. For our simulations, we have used i) 3-node network ii) 9-node grid network iii) 6-node bottleneck network iv) 11-node 2-bottleneck network. We have used a link of 1mbps for all the network topologies. Apart from this we have used a

Cache store of 10 packets. This is a very small cache storage but as our packet transfer is low we have decreased the storage size. We have mainly used three cache replacement policies namely i) Least Recently Used ii) First In First Out and iii) Random caching. As for the forwarding strategies, we have opted for the multipath forwarding strategies as it gives similar caching efficiency as shortest path with added benefits as per the research by Rossi and Rossini[37]. We have used a defined delay of 10ms for all our simulations to make the responses traceable for the simulation platform and to examine each packet carefully. We have also limited the packet queue to 20 packets to prevent the nodes being flooded with requests.

The simulations have been run for packet size of 1024 bytes and 512 bytes simultaneously. To get better results we created simulation environments with multiple content producers and multiple consumers. Similarly, we used multiple prefix options with different producers within the same network. The values for the response time for the sent packets for these variety of simulation environments were analyzed.

5. Simulation result and finding

We ran simulation environments for the current architectures for different topologies to identify the response times per packet in each of these scenarios. We first analyzed the response times for the individual packet for all these networks without the cache enabled. The results we obtained showed that all the topologies work in the line speed which means that the processing time among the nodes and the producers are negligible. We ran the same simulation to find the response times for a cache enabled is the nodes. Also for the cache enabled nodes the response time was equal to the line speed. As per the conducted simulation the interest packet transfer time per node was identified to be 0.010224s. This data is same for the case of a cache-less CCN network simulation environment. It means that a packet of data even at the architecture we have proposed will travel at the line speed.

To calculate the response time for each of the packet in our proposed environment we devised a formula using the response time of the obtained from the simulation environment. The response time for an interest request for a 1024 byte of data is given by

$$Y = X + (0.010224*2*B)$$

where Y is the response time for the proposed architecture, X is the response time for an interest request for 1024 byte of data in the CCN architecture and B is the number of extra hops to be travelled by the interest packet. As in the proposed architecture, each interest request should reach the producer even if the cache for the same content is found within the network path. So B is the number of hops travelled by the interest message to reach the producer. Also as the interest message should travel back to the cache hit router before it generates a content message we have multiplied the B by 2 to cover the reverse path as well. As per our simulation result 0.010224 is the per hop time required by an interest message.

As the response time for a request is directly proportional to the distance of the cached node, so we have run simulations for cached content at different nodes to compare the results. The graph below shows the response time of an interest message for a 1024 byte of data.

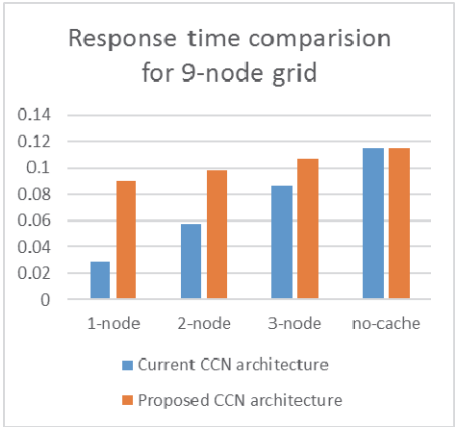

Fig. 6. Response time difference for a current CCN and proposed CCN architecture

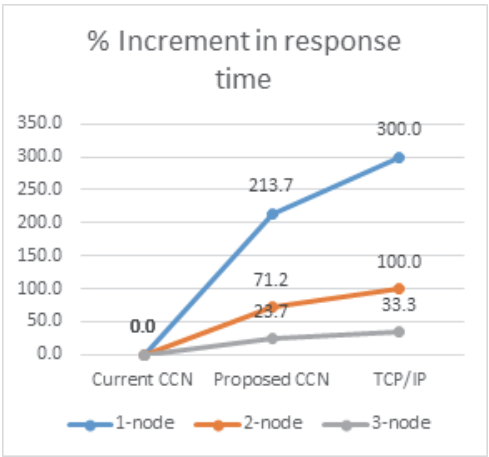

Fig. 7. % Increment of response time in different networking architectures

The graph shows the data for cache hit at different node levels. If no cache is hit, then in both the cases the response time is same. The graph shows the response time increases if the cache hit occurs at a further node but in case of the of the proposed architecture the difference in response time among the different cache hitting nodes is very low. We can see that the response time difference between the current architecture and the proposed architecture is very high. Although the response time for the proposed architecture is higher than the current CCN architecture it is still better than the current TCP/IP architecture nearly by 100%. Even at the worst case of which is the cache hit at the 3rd node we can see the proposed architecture response time increases by 23.7% which could not be considered as low.

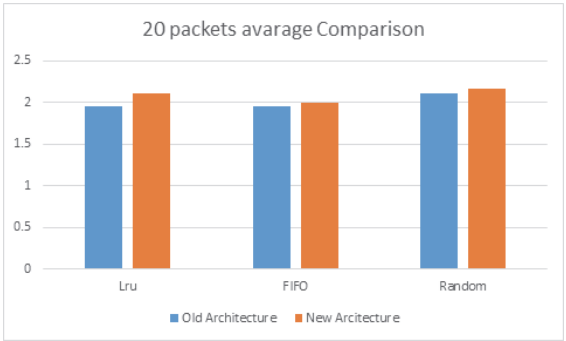

Fig. 8. Comparison of response time for 20KB data

To get a better picture of the difference of response time we ran a simulation for a bundle of packets. As in general an average size of a file size revolves around 20 KB we bundle a request for 20KB of data and then identify the average access time for the whole data. We analyze this time for multiple caching policies to know the difference. We have chosen LRU, FIFO and random caching as the caching policy for the content caching and the content is forwarded using multipath forwarding strategy. We have run this simulation is a 9-node grid network environment. Now in this case out of 20KB data not all have come from the producer nor all are accessed from the cache. This is the reason why the obtained response time is not the simple multiple of the response time of one packet. Also, as in the 9-node grid the caching could occur at the multiple nodes. So, the obtained response time for 20KB is a combination of response time for data cached at multiple nodes. We ran this simulation multiple time and got different data each time. To make comparison we need to obtain the response time for 20KB data in the proposed architecture. To calculate the response time, we used the reference from the cache hit data obtained from [38]. The experiment by them showed the cache hit ratio for different cache replacement and cache decision policy. So, using that data reference we can see that the cache hit ratio for a LRU caching policy is around 60%. So, we make assumptions that the when accessing a 20KB data, the cache hit would be around 50%. Using this reference, we can calculate the response time for the 20KB data in the proposed architecture. Among the 50% cache we randomly divided the cache among the tree nodes and averaged the response time for different cache distribution. We did the same for FIFO and random caching strategy and used the same cache hit data reference. The obtained data is plotted in the graph below.

This graph presents a different picture than that presented by the earlier comparison in the response time. Here the response time in the proposed architecture is not even 10% higher than the current CCN architecture. The difference is similar for different caching strategy. This data shows that the

proposed architecture will give similar performance to that of the current CCN architecture. But in terms of security the proposed architecture will give a better security from some of the security threats. We would discuss the effects of the new architecture on some of the common security threats.

5.1 Cache Pollution

Preventing cache pollution attack is one of the major objective of the proposed architecture. In cases when a server places its content on the caching routers to decrease its share of load, the router would now at least make the router to verify the content in the cache before each content is sent back to the requesting user.

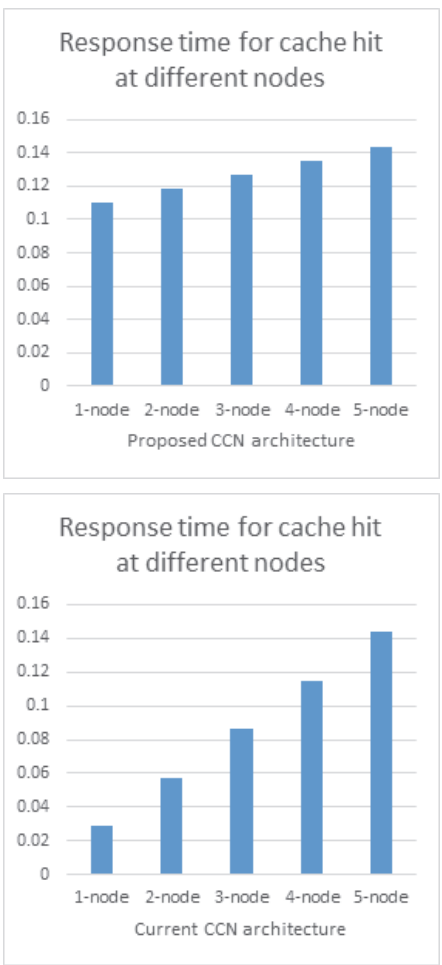

Fig. 9. Response time for cache hit at different nodes for current and proposed CCN architecture

In case of attacks that tend to change the cache locality or introduce false locality by trying to change the cache popularity mechanism, we propose to use the machine learning based methods like CacheShield proposed by Conti et al.[39]. This method sets a threshold value based on learning from previous attacks.

Apart from this case our aim is to also solve the special case of cache pollution where a content without a authentic producer can be kept in the cache to fulfill further incoming requests. As in the earlier architecture a content could remain in the cache even if the producer is not working or out of the network. This gave privilege for the illegitimate users to use the cache to share illegal content without by just using the content producer for a limited amount of time. But with the proposed architecture this case is not possible as each cached content could not be accessed if the content producer could not be reached. An incoming request for all traffics is sent to the respective content producers and if the producer does not exist, these requests are dropped. So even if the content is present in the cache it could not be accessed until the content producer is accessible. If the content producer is not accessed for a certain period of time, then the content cache will be overwritten by other contents. This would prevent the cases of unlawful data being shared just based on the cache hit criteria.

5.2 Timing Attack

The proposed architecture also works in towards reducing the timing attacks. Timing attacks are usually done to identify the data access behavior and pattern of other users from the same network. This is possible because the response time for the contents which are accessed from the nearest cache is less comparable to the one delivered from the server. But this approximation gets removed from the proposed architecture. The response time between a cached content and the cache less content is very low which minimizes the chances of the attacker to identify the content request pattern of the other users. The graph below shows the response time for cache hit at different nodes in the proposed architecture.

Figure 25: Response time for cache hit at different nodes for current and proposed CCN architecture

As the second graph shows, in case of the current architecture the time difference between the response time for cache hit at the closest node and other nodes is high. This means that an attacker can easily distinguish between the cache hit request at different nodes and thus identify the contents requested by users in its own network. But now in case of the proposed architecture, the time difference for cache hit is shown in the first graph. As showed by the graph the response time for cache hit for different nodes are very less. This means that even if the request is replied by the cache at the second caching node from the attacker, the attacker will not be able to distinguish the between these requests. This now prevents the attacker to easily identify the content access pattern of others users of the same network.

5.3 Un-Traceability

Un-traceability is another issue not properly solved in the case of the current CCN architecture. Although privacy advocates have been arguing that the current profiling of users is depriving them from the privacy rights which makes un-traceability as a feature than a problem. As in the case of the proposed architecture, there is a limited amount of traceability options available. This has been made possible by the fact that the interest request at least reaches the content producer before it is replied by the cache. So, this gives the privilege for the user sides to add some meta-info to the interest request that could be used by the servers to keep trace of the users. But the routing/forwarding method of CCN also promotes un-traceability as the reverse path of each request is determined by the incoming request and after the requests returns the path trace is also removed. The connectivity of all requests to the producers in the proposed architecture gives an open window for the identification of the new methods to make the users traceable from the producer side. Currently no proper mechanisms for the solution of un-traceability has been identified with opportunities for future research.

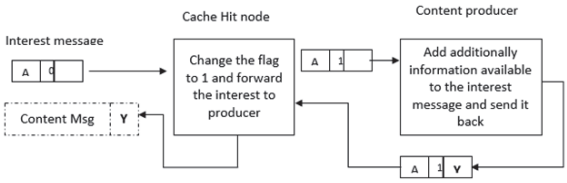

Fig. 10. Adding individualization content in the interest message

5.4 Individualization of data

This is another major solution proposed by the architecture. As discussed earlier about the lack of CCN architecture for individualization when it accessing contents from the cache. The proposed architecture has made individualization of data in CCN architecture possible. As the request packets should reach the content producer before it accesses the cached content, this provides the content producer to individualize the send content with some extra user specific content. As explained in the design of the proposed architecture as in figure 16, the content producer has a choice to decide to send a new individualized content or use the content from the cache. This gives the producer to change the content delivered to the user based on the user indicators. The proposed architecture has also presented an idea for added space for extra information in the interest request. This space would be used by the content producer to add the additional information which would help to individualize the content delivered. If packet architecture could be changed to add this additional information, then the main content could still be transferred from the cache while the content producer would just add individualization content to the interest message which would be transferred to the content message from the in-network caching node.

As shown in the figure the interest message could have some optional space for some additional information which the producer should add to the content. So, when the cache hit happens the flag is changed to 1 which means that when the interest reaches the content producer it just adds the additional information to the interest message and sends it back to the cache hit node. In the cache hit node when the

content message is generated the additional information in the form of Y as in the above figure will be added to the content message which gets delivered to the users.

Although we have proposed this method for generating more individualizing content, there are other major changes that needs to take place for this phenomenon to happen. The mechanism for transfer for additional information to the content message should be researched and also the user device and applications should be able to integrate the additional message they receive in the form of Y into the content. The major challenge that would come into this would be the work in line speed such that there would be no latency in the content delivery.

5.5 Statistic Infrastructure

As discussed earlier about the lack of statistic infrastructure in CCN prevents the content producers to track the real-time data access rate for their content. It also would not let them analyze the usual behaviors to improve their user experience. The solution to this problem could be use of report generators from each node to which would send the information to the producers in regular interval of time. But this adds computational load to the nodes as well as this data would not be real time information. The proposed architecture makes is easy to keep track of the statistics for the content producers. The identifiers for each user could be attached to the user generated request message and as the request reaches the content producers they get the statistics they need and they get it in real time. The proposed architecture solves this issue with no extra computational load added to the nodes.

6. Discussion and Conclusion

The proposed architecture gives a new framework for implementation of CCN without much of change on the basic principles of CCN. CCN network architecture is somewhat untraceable as per the routing phenomenon. As the user does not have fixed path to get the content and the generated path is removed each time the request is complete, which makes the CCN architecture untraceable. Not many researchers have looked for this issue but some have identified using user identifiers as a part of solution to make the users traceable. We also analyzed the issue of cache pollution as a part of security aspect of CCN. Cache pollution creates a scenario where the cache is filled up with unsuccessful and unused data while the real data is being fetched from the content producer itself. The cache pollution phenomenon could also be used by unlawful users to keep in their data in the cache without the existence of the actual content producer or server. Apart from this we discussed about the unavailability of statistical framework to keep track of necessary data for the content producers. We also discussed about the issue of lack of personalization/individualization of content in the CCN environment when most of the content are delivered form cache.

Analyzing different aspects of the CCN regarding its working and security we came up with a new approach toward the access of content in the CCN architecture which would solve some of the issues that have not been solved in the first section of our research. For this in the second part of our research we devised a slightly changed working architecture for CCN which involves the content producer for each of the content deliver. The proposed architecture is very much similar with the current CCN architecture and the only change is the way the content of the cache is accessed. The proposed architecture involves the travelling of the interest message all the way to the producer although the content is available in the cache of the in-network node. The interest message travelling to the content producer is not to access the content but to verify the producer is still in existent as well as allow the producer to be involved in the communication process. Despite the interest message is send to the producer, the content is delivered for the in-network cache which would not much hamper in the efficiency of the proposed architecture. We then discussed about the effects of the proposed architecture in naming, routing and caching of the CCN architecture. The proposed architecture did not change much of the naming and

caching scenario but made some slight modifications on the routing of the messages.

As CCN have some inbuilt simulators we used ndnSIM simulator to verify the usability and benefits of our proposed architecture. As the simulators were not designed for our proposed architecture we could not use them directly to evaluate our architecture. Instead we ran the simulators for the current CCN architecture and remodeled the results to obtain the values for our architecture. We used response time for content access as the indicator to make comparisons of our results. We obtained the response times for different network scenarios in the current CCN architecture. Then we obtained the response times for our proposed architectures using the mathematical formula obtained from the analysis of the response times of current CCN architecture.

Our results showed that the proposed architecture could provide solution to some of the identified security issues of CCN. Also, the proposed architecture did not hamper the previously defined security measures of the CCN architecture. We also analyzed the performance of the current and the proposed CCN architecture based on the response time and got very comparable results. Although the response time for a content access increases in our proposed architecture, but the increase is very low as compared to the security benefits and added feature for the CCN architecture.

CCN architecture has possibility for the use of cache as storage even after the content producer is not in existent. This feature could be exploited by unlawful users to make a storage space for their unlawful contents such that they could not help responsible as the server itself is not in the network. The availability of a content even after the server is down for some minutes poses a threat. CCN architectures also lacks the framework for collecting the data for access of the servers as each server does not receive all the traffic that are generated for it due the data delivery of the in-network caches. Apart from this CCN architecture lacks the framework to support individuality of data as every content in the internet is personalized as per the user characteristics. We proposed a slightly modified version of CCN architecture to solve these issues. We designed the architecture of the proposed solution such a way that it will not change the basis of CCN but would provide the solutions for our identified problems.

The proposed solution was then compared with the current CCN architecture in terms of the response time for the requested content. The comparison showed that the response time for the proposed architecture is slightly higher than that of the current CCN architecture. The difference of the response time was considerable which derived the conclusion that the new architecture would work with almost similar efficiency as the current architecture.

The results of the response time also showed that the proposed CCN architecture also prevents the timing attack as the difference is response time for the content accessed from different levels of cache is very low. In the current CCN architecture the difference in response time for the content accessed from different levels of caches are very high which makes it easy for attackers to identify if the content is accessed from cache or not.

The proposed architecture also solved the issue for the cache pollution with unlawful data without any content producers. As the proposed architecture prevents no content be delivered without the request reaching the content producers, which means that any content in the cache without a producer cannot be accessed. If a request is sent for this kind of cache, then the interest is dropped as it finds no producer and ultimately the cache is also replaced as it would have no hits. This solution would effectively eliminate the risk of a cached content being accessed from the cache without the producer of the authorization of the producer.

The proposed architecture thus gives a solution to major of the identified problems related to the CCN architecture as well as open opportunities for the solution of some other identified problems as well. Further research in this architecture would bring the more advancements for the proper implementation of CCN as replacement of the current TCP/IP network.

References

1. Trossen, D., Sarela, M., & Sollins, K. (2010).

Arguments for an information-centric internetworking architecture. ACM SIGCOMM Computer Communication Review, 40(2), 26-33
2. Rexford, J., & Dovrolis, C. (2010). Future Internet architecture: clean-slate versus evolutionary research. Communications of the ACM, 53(9), 36-40.
3. "IRTF Information-Centric Networking Research Group (ICNRG)". Irtf.org. N.p., 2016. Web. 17 Aug. 2016.
4. Jacobson, V., Smetters, D. K., Thornton, J. D., Plass, M. F., Briggs, N. H., & Braynard, R. L. (2009, December). Networking named content. In Proceedings of the 5th international conference on Emerging networking experiments and technologies (pp. 1-12). ACM.
5. Sourlas, V., Flegkas, P., & Tassiulas, L. (2014). A novel cache aware routing scheme for Information-Centric Networks. Computer Networks, 59, 44-61. http://dx.doi.org/10.1016/j.bjp.2013.12.002
6. Cheriton, D. R., & Gritter, M. (2000). TRIAD: A new next-generation Internet architecture.
7. Baccala, B. (2002). Data Oriented Networking.
8. Arianfar, S., Nikander, P., & Ott, J. (2010, June). Packet-level caching for information-centric networking. In ACM SIGCOMM, ReArch Workshop.
9. Jacobson, V., Smetters, D. K., Thornton, J. D., Plass, M. F., Briggs, N. H., & Braynard, R. L. (2009, December). Networking named content. In Proceedings of the 5th international conference on Emerging networking experiments and technologies (pp. 1-12). ACM.?
10. Koponen, T., Chawla, M., Chun, B. G., Ermolinskiy, A., Kim, K. H., Shenker, S., & Stoica, I. (2007, August). A data-oriented (and beyond) network architecture. In ACM SIGCOMM Computer Communication Review (Vol. 37, No. 4, pp. 181-192). ACM.?
11. Ahlgren, B., D'ambrosio, M., Dannewitz, C., Eriksson, A., Golic, J., Gr?nvall, B., ... & M?kel?, J. (2010). Second netinf architecture description. 4WARD EU FP7 Project, Deliverable D-6.2 v2. 0.
12. Ain, M., Trossen, D., Nikander, P., Tarkoma, S., Visala, K., Rimey, K., ... & Kj?llman, J. (2009). D2. 3-architecture definition, component descriptions, and requirements. Deliverable, PSIRP 7th FP EU-funded project.
13. Jacobson, V., Smetters, D. K., Thornton, J. D., Plass, M. F., Briggs, N. H., & Braynard, R. L. (2009, December). Networking named content. In Proceedings of the 5th international conference on Emerging networking experiments and technologies (pp. 1-12). ACM.
14. Tourani, R., Mick, T., Misra, S., & Panwar, G. (2016). Security, Privacy, and Access Control in Information-Centric Networking: A Survey. arXiv preprint arXiv:1603.03409.
15. Lagutin, D., Visala, K., & Tarkoma, S. (2010). Publish/Subscribe for Internet: PSIRP Perspective. Future internet assembly, 84.
16. Akhshabi, S., & Dovrolis, C. (2013). The evolution of layered protocol stacks leads to an hourglass-shaped architecture. In Dynamics On and Of Complex Networks, Volume 2 (pp. 55-88). Springer New York.
17. Vasilakos, A., Li, Z., Simon, G., & You, W. (2015). Information centric network: Research challenges and opportunities. Journal Of Network And Computer Applications, 52, 1-10. http://dx.doi.org/10.1016/j.jnca.2015.02.001
18. Almeida, F. L. F., & Louren?o, J. M. (2012). Information centric networks-design issues, principles and approaches. International Journal of Latest Trends in Computing, 3(3). ?
19. Zhang, X., Chang, K., Xiong, H., Wen, Y., Shi, G., & Wang, G. (2011, October). Towards name-based trust and security for content-centric network. In 2011 19th IEEE International Conference on Network Protocols (pp. 1-6). IEEE.
20. Stoica I, Morris R, Karger D, Kaashoek MF, Balakrishnan H. Chord: a scalable peer- to-peer lookup service for internet applications. In: Proceedings of the 2001 conference on applications, technologies, architectures, and protocols for computer communications, ser. SIGCOMM '01; 2001.
21. Misra, S., Tourani, R., & Majd, N. E. (2013,

August). Secure content delivery in information-centric networks: Design, implementation, and analyses. In Proceedings of the 3rd ACM SIGCOMM workshop on Information-centric networking (pp. 73-78). ACM.

22. Lauinger, T., Laoutaris, N., Rodriguez, P., Strufe, T., Biersack, E., & Kirda, E. (2012). Privacy implications of ubiquitous caching in named data networking architectures. Technical report, TR-iSecLab-0812-001, iSecLab.
23. Rossini D, Rossi D. A dive into the caching performance of content centric networking. Technical report, Telecom ParisTech; 2011.
24. Muscariello, L., Carofiglio, G., & Gallo, M. (2011, August). Bandwidth and storage sharing performance in information centric networking. In Proceedings of the ACM SIGCOMM workshop on Information-centric networking (pp. 26-31). ACM.
25. Udugama, A. (2015). Enhanced Forwarding Strategies in Information Centric Networking (Pd.D). University of Bremen.
26. Arianfar, S., Nikander, P., & Ott, J. (2010, November). On content-centric router design and implications. In Proceedings of the Re-Architecting the Internet Workshop (p. 5). ACM.
27. Ahlgren, B., Dannewitz, C., Imbrenda, C., Kutscher, D., & Ohlman, B. (2012). A survey of information-centric networking. IEEE Communications Magazine, 50(7), 26-36.
28. Adamic, L. A., & Huberman, B. A. (2002). Zipf's law and the Internet. Glottometrics, 3(1), 143-150.
29. Sugiyama, K., Hatano, K., & Yoshikawa, M. (2004, May). Adaptive web search based on user profile constructed without any effort from users. In Proceedings of the 13th international conference on World Wide Web (pp. 675-684). ACM.
30. Kaushik, P. (2016). Tomorrow's Internet: A World of Hyper-Personalized Tribes?. WIRED. Retrieved 4 September 2016, from https://www.wired.com/insights/2014/03/todays-internet-world-hyper-personalized-tribes/
31. Hu, Y. Performance-based Pricing Models in Online Advertising. SSRN Electronic Journal. http://dx.doi.org/10.2139/ssrn.501082
32. CCNx | PARC's implementation of content-centric networking. (2016). Ccnx.org. Retrieved 30 June 2016, from http://www.ccnx.org/
33. Afanasyev, A., Moiseenko, I., & Zhang, L. (2012). ndnSIM: NDN simulator for NS-3. University of California, Los Angeles, Tech. Rep.
34. Chiocchetti, R., Rossi, D., & Rossini, G. (2013, June). ccnsim: An highly scalable ccn simulator. In 2013 IEEE International Conference on Communications (ICC) (pp. 2309-2314). IEEE..
35. Nikolaos Vastardis Personal Webpage. (2016). Privateessex.ac.uk. Retrieved 30 June 2016, from http://privatewww.essex.ac.uk/~nvasta/ICNSim.htm
36. Ghodsi, A., Shenker, S., Koponen, T., Singla, A., Raghavan, B., & Wilcox, J. (2011, November). Information-centric networking: seeing the forest for the trees. In Proceedings of the 10th ACM Workshop on Hot Topics in Networks (p. 1). ACM.
37. Rossi, D., & Rossini, G. (2011). Caching performance of content centric networks under multi-path routing (and more). Relat?rio t?cnico, Telecom ParisTech.
38. Rossini, G. & Rossi, D. (2013). Evaluating CCN multi-path interest forwarding strategies. Computer Communications, 36(7), 771-778. http://dx.doi.org/10.1016/j.comcom.2013.01.008
39. Conti, M., Gasti, P., & Teoli, M. (2013). A lightweight mechanism for detection of cache pollution attacks in Named Data Networking. Computer Networks, 57(16), 3178-3191.
40. Udugama, A. (2015). Enhanced Forwarding Strategies in Information Centric Networking (Doctoral dissertation, Faculty of Physics and Electrical Engineering, University of Bremen).

Aashis Sharma (aashis.sharma23@gmail.com)

2012 Bachelor of Engineering in Electronics and Communication Engineering, Kathmandu Engineering College, Tribhuvan University
2013 Computer Engineer- National Information Technology Center
2016 Masters in ICT Convergence, Handong Global University, Pohang, South Korea
2017-현재 Specialist (Internal Audit) - Ncell Pvt. Ltd.

관심분야 : Content Centric Networking (CCN), Network Security, Data Security, Cloud Computing and Encryption Technologies

Yun Seon Kim (sean0831@handong.edu)

2001 한동대학교 경영학사, 전산공학사 학사
2004 Unviersity of Pittsburgh Information Science 석사
2010 Wayne State University Industrial and Systems Engineering 박사
2010 YS sercurity and computers, Robust system inc. 대표
2013 연변 과학기술대학교 상경학부 교수
2015～현재 한동대학교 재직 중

관심분야 : Knowledge management, intelligent system, data analysis, global development