Journal of Korea Multimedia Society Vol. 17, No. 6, June 2014(pp. 697-705)
http://dx.doi.org/10.9717/kmms.2014.17.6.697

콘텐츠 중심 네트워킹의 콘텐츠 이름 인코딩 기법

김 대 엽†

A Content-Name Encoding Scheme for CCN

Kim DaeYoub†

ABSTRACT

For enhancing network efficiency, content-centric networking (CCN) allows network nodes to temporally cache a transmitted response message(Data) and then to directly respond to a request message (Interest) for previously cached contents. Also, CCN is designed to utilize a hierarchical content-name for transmitting Interest/Data instead of a host identity like IP address. This content-name included in Interest/Data reveals both content information itself and the structure of network domain of a content source which is needed for transmitting Interest/Data. To make matters worse, This content-name is human-readable like URL. Hence, through analyzing the content-name in Interest/Data, it is possible to analyze the creator of the requested contents. Also, hosts around the requester can analyze contents which are asked by the requester. Hence, for securely implementing CCN, it is essentially needed to make the content-name illegible. In this paper, we propose content-name encoding schemes for CCN so as to make the content-name illegible and evaluate the performance of our proposal.

Key words: Future Internet, CCN, Identity Management, Anonymity, Privacy Protection

1. 서 론

초기 인터넷은 원격 호스트들 사이의 안전한 네트워크 연결을 제공하기 위한 목적으로 제안되었다. 그러므로 대용량 콘텐츠 전송 및 네트워크 트래픽 증가로 인하여 발생될 수 있는 네트워크 병목현상, 취약한 보안 구조로 인한 침해 사고 증가, 호스트의 빈번한 이동 시 발생하는 비효율성과 같은 인터넷이 갖고 있는 다양한 문제점들과 그 해결 방안은 초기 인터넷 설계 시 고려되지 않았다[1]. 이와 같은 인터넷의 문제점들을 해결하고 멀티미디어 콘텐츠 서비스를 효과적으로 지원하기 위하여 다양한 미래 인터넷 기술 연구가 진행되고 있다[2-4]. 특히, 미래 인터넷 기술들은 콘텐츠 제공자에게만 집중되는 콘텐츠 요청 메시지를 효율적으로 분산시키기 위하여 End-User, Service Provider, Proxy-Server 뿐만 아니라 네트워킹 라우터와 같은 다양한 네트워크 기기에 콘텐츠를 임시 저장할 수 있는 기능을 구현하고, 이와 같은 네트워크 기기에 저장된 콘텐츠를 이용하여 콘텐츠 요청 메시지를 분산처리 함으로써 네트워크 효율성을 높이려는 시도를 하고 있다.

미래 인터넷 기술 중 하나인 콘텐츠 중심 네트워킹(Content-Centric Networking, CCN)은 콘텐츠 이름(content-name)에 기반 한 패킷 포워딩 기술과 네트워크 기기/노드에 콘텐츠 임시 저장(Caching) 기능을 구현하여 효과적인 콘텐츠 전송을 제공 한다

※ Corresponding Author : Kim DaeYoub, Address : 17, Wauan-gil, Bongdam-eup, Hwaseong-si, Gyeonggi-do, Korea, TEL : +82-16-818-6913, FAX : +82-31-229-8284, E-mail : daeyoub69@suwon.ac.kr
Receipt date : Feb. 12, 2014, Revision date : Apr. 20, 2014
Approval date : May. 7, 2014

† Suwon Univ. Dept. Information Security
※ This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korean Government (No. NRF-2013R1A1A2008389).

[3, 4]. 즉, 효율적인 네트워킹을 위하여 CCN은 네트워크 경로 상에 있는 중간 네트워크 노드들이 수신된 콘텐츠(응답 메시지, Data)를 해당 노드의 Content Store (CS)에 임시 저장할 수 있다고 가정한다. 또한, CCN 노드는 콘텐츠 요청 메시지(Interest)를 수신하면, 수신된 Interest에 대응하는 Data가 CS에 저장되어 있는지 확인한 후, 대응되는 Data가 CS에 존재하면, 해당 Data를 콘텐츠를 요청한 사용자에게 전송하고 수신된 Interest를 더 이상 다음 노드에게 전송하지 않고 처리를 완료한다. 이와 같이 중간 노드에 의하여 Interest가 처리될 수 있기 때문에 콘텐츠 제공자에게 집중되는 Interest를 분산처리 할 수 있을 뿐만 아니라 전체 네트워크 트래픽 양 또한 효과적으로 줄일 수 있다.

이와 같이 중간 네트워크 노드에 임시 저장된 콘텐츠를 네트워킹에 효과적으로 활용하기 위하여 IP 주소와 같은 호스트 Identity 대신 CCN은 그림 1-(B)와 같은 계층화된 content-name을 참조하여 Interest/Data를 전송한다. 또한, 콘텐츠 사용자의 프라이버시 보호를 위하여 Interest는 사용자에 대한 정보를 포함하지 않는다. 그러므로 네트워크 노드가 Data를 수신했을 때, 해당 Data를 사용자에게 전송하기 위해서 네트워크 노드는 Interest와 Interest의 유입 경로(Network Interface, Face)를 Pending Interest Table (PIT)에 저장한 후, PIT를 참조하여 수신된 Data를 사용자에게 전송한다.

그러나 이와 같은 content-name 기반의 네트워킹은 콘텐츠 사용자 및 제공자의 프라이버시가 침해될 수 있다는 문제점을 여전히 갖고 있다. 본 논문에서는 이와 같은 문제점을 해결하기 위해 앞서 제안된

winterstory.avi generated
by Alice, January, 2014

29A4CD87652FEA31 =
H(winterstory.avi, s1, Alice, January, 2014)

(A) Flat Content Name

/SU/IT/staff/Alice/winterstory.avi/20140108/s1

Routing Information Content Information

(B) Hierarchical Content Name

Fig. 1 Content Name for Future Internet.

CCN Name Privacy 보호를 위한 연구 결과들을 살펴본 후, 개선 방안을 제안한다.

2. 미래 인터넷 기술의 콘텐츠 이름

대표적인 미래 인터넷 기술 중에는 콘텐츠 요청 및 응답 메시지의 전송 경로를 결정하기 위해 유일한 콘텐츠 이름(식별자, Identity)을 사용하는 기술들이 있다. 이와 같은 기술들은 그 특성에 따라 다음과 같은 고유의 콘텐츠 이름 구조를 사용 한다[5].

2.1 Flat Content Name (FCN)

그림 1-(A)는 FCN의 구성 예를 보여준다. FCN은 콘텐츠의 식별자(Identity)를 유일하게 정의하기 위하여 콘텐츠 정보에 대한 해쉬 값을 콘텐츠 식별자로 사용한다. FCN 기반 미래 인터넷 기술은 사용자가 콘텐츠를 사용하기 위해서는 우선 Resolution System (RS)을 이용하여 해당 콘텐츠가 저장되어 있는 호스트의 정보를 확인한 후, 콘텐츠 요청 메시지를 해당 호스트로 전송하도록 제안한다. 그러므로 콘텐츠 제공자가 콘텐츠를 인터넷을 통하여 배포하기 위해서는 RS에 해당 콘텐츠 정보와 함께 호스트 정보를 반드시 등록해야만 하며, RS가 서비스 거부 공격과 같은 공격의 목표가 된 경우, 정상적인 콘텐츠 배포가 어렵다는 문제점이 있다. 그러나 콘텐츠 정보의 해쉬 값만을 이용하여 호스트 정보를 탐색하기 때문에 콘텐츠 요청 메시지 분석을 통한 개인 정보 유출과 같은 프라이버시 문제를 해결할 수 있다.

2.2 Hierarchical Content Name (HCN)

그림 1-(B)는 HCN의 구성 예이다. HCN은 RS와 같은 시스템의 도움 없이 콘텐츠 요청 및 응답 메시지를 전송할 수 있도록 콘텐츠 이름을 계층화된 네트워크 도메인 정보와 콘텐츠 자체 정보로 구성된다. 네트워크 노드는 요청 및 응답 메시지에 포함되어 있는 HCN의 계층화된 네트워크 도메인 정보를 이용하여 메시지 전송 경로를 결정한다. RS와 같은 별도의 시스템을 필요로 하지 않기 때문에 전체적인 네트워크 구성이 FCN에 비하여 간단하다는 장점이 있지만, HCN을 분석하여 콘텐츠를 요청한 사용자의 콘텐츠 이용 정보 및 콘텐츠 제공자의 정보를 파악할 수 있기 때문에 프라이버시 침해 문제가 발생할 수 있다.

3. 콘텐츠 중심 네트워킹

미래 인터넷 기술 중 HCN을 이용하는 대표적인 기술로 CCN을 들 수 있다. 그림 2는 CCN에서 Interest와 Data를 처리하는 절차를 설명한다. (1)~(6)은 Interest 처리 절차를 설명하고, (7)~(10)은 Data 처리 절차를 설명 한다:

(1) Face 0을 통하여 Interest가 수신된다.

(2) CS에 수신된 Interest에 대응되는 Data가 저장되어 있는지 확인한다. 만약 저장되어 있다면, Face 0을 통해 해당 Data를 전송한다.

(3) CS에 수신된 Interest에 대응되는 Data가 저장되어 있지 않다면, PIT에 해당 Interest에 대응되는 정보가 있는지 확인한다. 만약 있다면, 해당 정보의 incoming Face 필드에 Face 0을 추가한다.

(4) PIT에 대응되는 정보가 없다면, FIB (Forwarding Information based) 테이블을 참조하여 수신된 Interest를 전송할 Face (ex. Face 2)를 선택한다.

(5) PIT에 수신된 Interest와 incoming Face 정보를 기록한다.

(6) FIB에서 선택한 Face를 통하여 수신된 Interest를 전송한다.

(7) Data가 수신된다.

(8) PIT에 수신된 Data에 대응되는 Interest 정보가 있는지 확인한다. 만약 없으면, 해당 Data는 폐기 처리 된다.

(9) PIT에 수신된 Data에 대응하는 Interest 정보가 존재하면, CS에 Data를 저장한다.

(10) Data를 대응되는 Interest 정보의 incoming Face들을 통해서 전송한다.

4. CCN Content-Name 인코딩

4.1 암호화 기반 Content-Name 인코딩

그림 2에서 설명한 것처럼, CCN은 Interest에 포함된 content-name을 기반으로 FIB 정보를 탐색하여 Interest를 전송할 Face를 결정한다. 그러므로 content-name의 일부 또는 전부를 인코딩/암호화하여 Interest를 생성할 경우, 해당 Interest 전송 경로 정보를 FIB에서 찾을 수 없게 된다. 이와 같은 경로 탐색 문제를 해결하고, 콘텐츠를 요청한 사용자 및 콘텐츠 제공자의 프라이버시를 보호하기 위한 방안에 대한 연구가 필요하다.

이와 같은 문제점들을 해결하기 위한 content-name 인코딩 방안이 다음과 같이 제안되었다 [6,7].

(1) Flexible Name Encryption (FNE): 사용자는

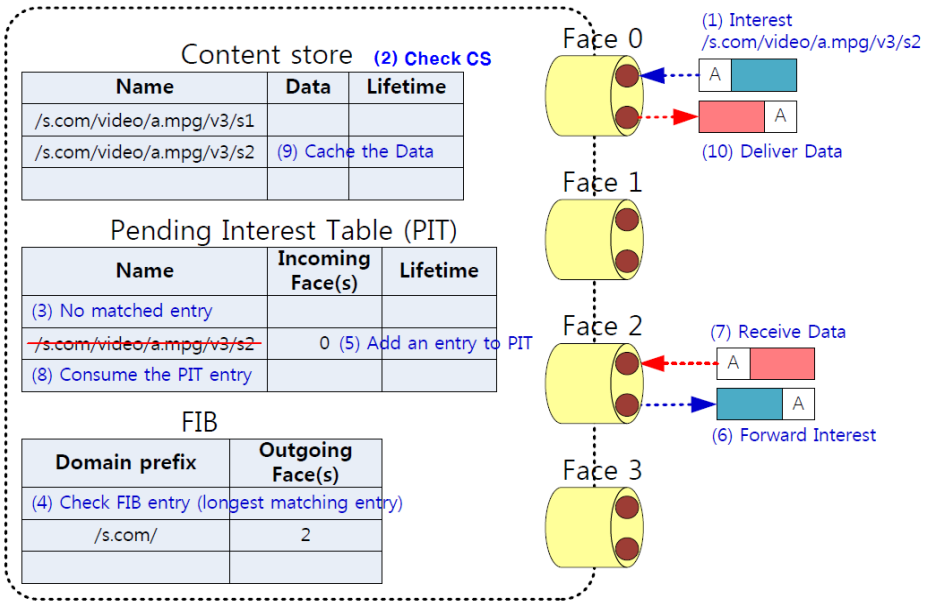


Fig. 2 CCN Forwarding Model.

계층화된 content-name을 구성하는 각각의 이름 요소 (Name Component) 중 외부에 공개되지 않기를 바라는 name-component를 선택한다. 선택된 각각의 name-component에는 대응하는 상위 네트워크 도메인과 도메인 게이트웨이가 존재하며, 해당 도메인 게이트웨이는 자신의 유일한 암호화 공개키 (Public Encryption Key)를 공개한다. 그림 3-(A)에서와 같이 사용자가 name-component 중에서 "uci"를 인코딩 하려 한다면, "uci"의 상위 이름 요소인 "uc"에 대응하는 도메인 게이트웨이의 암호화 공개키를 획득한 후, "uci"를 획득한 공개키로 암호화해서 Interest를 생성한다. 이렇게 생성된 Interest가 "uc"에 대응하는 도메인 게이트웨이에 전송되면, 해당 게이트웨이는 Interest의 암호화 된 부분("uci")을 복호화하여 Interest의 content-name을 수정한 후, 하위 도메인 호스트("uci")로 전송한다.

(2) Concentric Name Encapsulation (CNE): CNE는 그림 3-(B)와 같이 Interest와 Data의 이름 요소를 계층적으로 암호화한다. 특히, 사용자는 Interest를 생성할 때 인코딩 될 name-component 뿐만 아니라 Data 암호화에 사용될 Session Key (SK)를 함께 암호화 하여 전송한다. 도메인 게이트웨이 중 SK를 복호화한 게이트웨이는 SK를 이용하여 수신된 Data를 암호화 하여 사용자에게 전송한다.

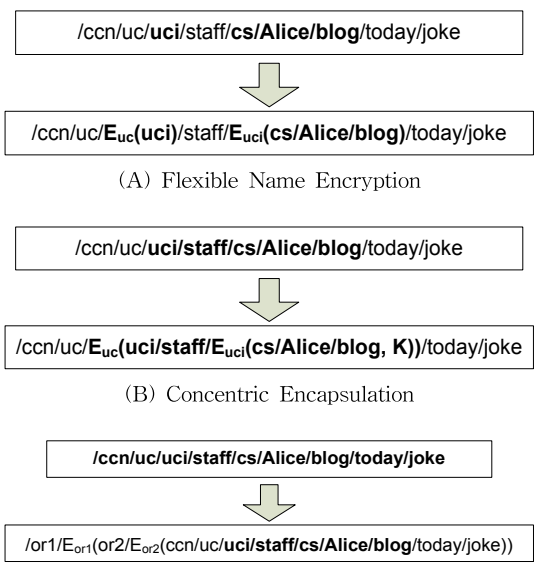


Fig. 3 CCN Name Encryption Scheme.

(3) Full Encapsulation using Onion Routing (FEO): FEO는 Onion Router(OR)을 이용하여 콘텐츠 사용자 호스트의 주변 호스트가 해당 사용자가 생성한 Interest의 내용을 분석하지 못하도록 함으로써 사용자의 프라이버스를 보호하기 위하여 제안되었다. 그림 3-(C)는 다수의 OR들을 이용한 인코딩 방안을 설명한다. OR 기능을 갖는 호스트는 Identity와 암호화 공개키를 Onion Router Manager (ORM)에 등록한다. 콘텐츠 사용자는 ORM로부터 최신 Onion Router List를 전송 받은 후, 해당 List에서 최소 2개 이상의 OR을 무작위로 선택한다. 선택된 OR의 공개키를 이용하여 Interest의 content-name을 계층적으로 암호화한다. 이 때, 해당 Interest가 선택된 OR들을 경유하도록 선택된 OR의 도메인 이름을 Interest에 추가하여 캡슐화(Encapsulation) 한다. 이와 같이 캡슐화 된 Interest를 수신한 OR은 Interest에서 자신의 도메인 이름 요소를 삭제한 후 복호화 비밀키를 이용하여 Interest를 복호화한 후, 복호화 된 Interest를 CCN을 통하여 전송한다.

4.2 해쉬 기반 Content-Name 인코딩

앞서 제안된 암호화 기반 content-name 인코딩 기법들은 여러 사용자가 동일 콘텐츠를 중복해서 요청하더라도 인코딩 된 content-name이 사용자마다 서로 다르면 중간 네트워크 노드의 CS에 해당 콘텐츠가 임시 저장되어 있어도 네트워킹에 이를 활용할 수 없다. 그러므로 암호화 기반 content-name 인코딩 기법은 CCN의 성능 저하를 초래할 수 있다.

이와 같은 문제점을 해결하기 위하여 해쉬 테이블 기반의 인코딩 기법(Name Anonymity using Hash Table, NAH)이 제안 되었다 [8]. NAH를 구현하기 위하여 각각의 네트워크 도메인은 그림 4처럼 단방향 해쉬 함수를 이용하여 자신의 서브 도메인 이름을 인코딩한 후 저장/관리 한다: 최상위 도메인 D_{11}은 3개의 서브 도메인 D_{21}, D_{22}, D_{23}을 갖고, 각각의 서브 도메인은 2차 서브 도메인을 갖는다. 이 때, 각 도메인 게이트웨이는 서브 도메인 이름 및 그 인코딩 이름을 알고 있다. 또한, 각 도메인은 자신의 도메인 이름과 인코딩 된 도메인 이름을 함께 관리한다. 사용자가 요청하는 콘텐츠 파일(cn)의 계층화된 이름을 $D_{11}/D_{21}/D_{31}/cn/s_0$이라고 할 때, 만약 사용자가 도메인 이름 D_{31}과 콘텐츠 파일 이름 cn이 인코딩 된

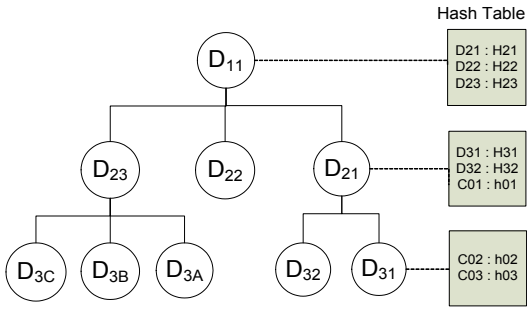


Fig. 4 Hashed CCN Name.

Interest를 생성하려 한다면, 다음과 같이 인코딩 된 content-name을 포함하는 Interest를 생성한 후, 생성된 Interest를 전송 한다:

Interest::/D_{11}/D_{21}/.a.c.H_{31}/.a.c.H_{cn}/s_0

또는,

Interest::/D_{11}/D_{21}/.a.c.H_{31}/.a.c.H_{cn}/s_0/.i.c.H_{on}.

여기서 이름 요소의 의미는 다음과 같으며 H()는 단방향 해쉬 함수를 의미 한다:

- H_{31}=H(D_{31}). 도메인 이름의 해쉬 값.
- H_{cn}=H(cn). 콘텐츠 파일 이름의 해쉬 값.
- H_{on}=H(D_{11}/D_{21}/D_{31}/cn/s_0). 계층화된 CCN content-name의 해쉬 값, optional field.
- .a.c. : Name Maker. 이하 이름 요소(name-component)가 인코딩이 적용됨을 의미한다.
- .i.c. : Name Maker. 이하 name-component가 계층화된 CCN content-name의 해쉬 값임을 의미한다.

D_{21} 노드가 해당 Interest를 수신하면 해쉬 테이블에서 H_{31}에 대응하는 D_{31}을 찾은 후, 해당 Interest를 D_{31} 노드에게 전송한다. D_{31} 노드가 해당 Interest를 수신하면, 인코딩을 위하여 수신된 Interest 이름을 이용하여 Data를 다음과 같이 캡슐화 해서 전송 한다:

```
Data {
  name:=D11/D21/.a.c.H31/.a.c.Hcn/s0 ;
  contentID:=Hon ;
  data :={
    name := D11/D21/D31/cn/s0 ;
    content ;
  }
}
```

그림 5는 NAM에서 정의한 Interest와 Data를 이

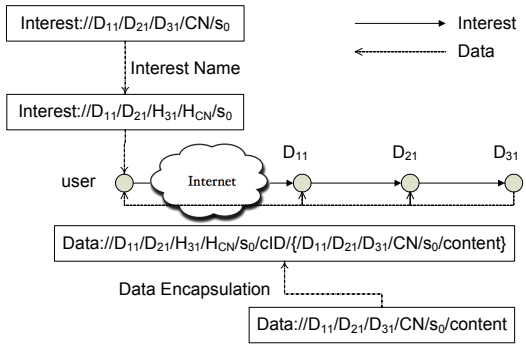


Fig. 5 NAH procedure to handle Interest/Data.

용하여 네트워크 노드에서 CCN 메시지가 처리되는 과정을 설명한다. 사용자가 콘텐츠 (cn)의 0번째 세그먼트를 요청하는 Interest를 생성할 때, content-name의 일부 요소가 외부에 노출되는 것을 막기 위하여 해쉬 함수를 이용하여 해당 요소에 대한 인코딩 작업을 수행한 후, 생성된 Interest를 CCN을 통해 전송한다. 중간에 권한 없는 노드 또는 사용자가 해당 Interest를 획득하여 분석해도, 단방향 해쉬 함수를 이용하여 인코딩 된 name-component를 디코딩 할 수 없기 때문에 요청된 콘텐츠의 출처와 내용을 예측할 수 없다. 해당 Interest가 콘텐츠 제공자가 속한 네트워크 도메인 게이트웨이를 통해 해당 제공자에게 전송되면, 이에 대응하는 Data를 콘텐츠를 요청한 사용자에게 전송하기 위하여 Interest에 포함된 인코딩 된 content-name을 이용하여 캡슐화를 수행한 후, 캡슐화 된 Data를 CCN을 통하여 사용자에게 전송한다.

그러나 이와 같이 캡슐화 된 Data는 여전히 인코딩 되지 않은 content-name (Data.data.name)과 콘텐츠 파일(Data.data.content)을 포함하고 있다. 그러므로 전송되는 Data를 중간 노드를 통하여 획득/분석하면 요청된 콘텐츠에 대한 정보를 획득할 수 있다. 또한, Interest 생성 시 “.i.c.H_{on}” 항목이 선택적으로 적용되기 때문에 “.i.c.H_{on}”을 포함하지 않는 Interest의 경우, 앞서 제안된 인코딩 기법들처럼 중간 노드에 임시 저장된 콘텐츠를 네트워킹에 활용할 수 없기 때문에 CCN 네트워킹의 효율성이 저하될 수 있다.

5. 개선된 CCN Content-Name 인코딩

앞서 소개된 content-name 인코딩 기술들은 성능

저하 및 불완전한 인코딩으로 인한 정보 유출 등의 문제점들을 갖고 있다. 본 절에서는 NAH의 문제를 개선하기 위한 인코딩 기법 (Improved Name Anonymity Scheme, INA) 제안하고, 제안된 INA를 효과적으로 구현을 위하여 수정된 CCN 운영 방안을 제안한다. 특히, Interest와 Data의 전송 경로를 결정하는 CCN의 방법을 수정하여 성능 저하 없이 보다 안전한 content-name 인코딩 서비스를 제공할 수 있도록 설계되었다.

5.1 Content-name 인코딩을 위한 메시지 구조

제안하는 content-name 인코딩 기술은 CCN content-name의 요소들을 해쉬 기반으로 인코딩하고, 이를 효율적이고 안전하게 처리 위하여, Interest는 기존의 CCN에서와 같이 HCN 구조의 content-name을 사용하고, 추가로 콘텐츠 식별자로 사용될 c_ID를 포함하도록 구성한다. 여기서, c_ID는 인코딩 되지 않은 원래 content-name을 단방향 해쉬 함수를 이용하여 계산한 값이다.

반면에 Data는 FCN 구조의 content-name을 사용한다. 즉, Data는 계층화된 content-name 대신에 c_ID 값만을 content-name으로 사용하도록 수정한다. 수정된 Interest와 Data 구조를 다음과 같이 제안한다:

```
Interest {
  name ;            // 이름 or 인코딩 이름
  c_ID:= Hon ;      // 콘텐츠 unique ID
}

Data {
  name:= Hon ;      // 콘텐츠 unique ID
  data ;
}
```

그림 6은 개선된 Interest/Data 구조를 갖는 CCN 메시지를 네트워크 상에서 처리하는 과정을 설명한다. Interest는 NAH에서 제안한 방식에 의하여 포워딩 된다. 즉, 인코딩 되지 않은 name-component를 기반으로 FIB를 참조하여 Interest는 D_{21} 노드까지 전송된 후, D_{21} 노드에서 인코딩 된 name-component H_{31}을 분석하여, H_{31}에 대응되는 D_{31} 노드에게 Interest를 전송한다.

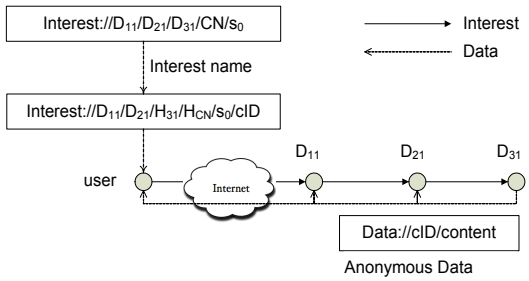


Fig. 6. INA procedure to handle Interest/Data.

Data는 기존 CCN의 계층화된 content-name을 사용하지 않고, content-name의 해쉬 값만을 참조하여 사용자들에게 전송된다. Data가 계층화된 content-name을 포함하고 있지 않기 때문에 중간 노드에서 전송되는 Data를 분석한다 하더라도 해당 Data의 정보를 확보하기 어렵다.

그러나 이와 같은 Data 전송 방식은 기존의 CCN 전송 방식으로는 처리할 수 없기 때문에 CCN 전송 방식을 일부 수정해야만 한다.

5.2 INA의 Interest/Data 처리 절차

INA에서 제안된 Interest와 Data를 처리하기 위하여 CCN 노드는 PIT와 CS의 entry를 계층화된 content-name 기반으로 관리하지 않고, 콘텐츠의 H_{on} 값을 기반으로 관리한다. 즉, PIT의 entry를 {Interest.c_ID=H_{on}, incoming Faces}로 구성한다. 또한, CS entry도 {Data.name=H_{on}, data}로 구성한다. 이와 같이 수정된 PIT와 CS를 기반으로, CCN의 Interest/Data 처리 절차를 다음과 같이 개선한다. 이해를 돕기 위해 수정된 절차는 그림 2의 과정을 바탕으로 설명 한다:

(1) Face 0을 통하여 Interest가 수신된다.

(2) 수신된 Interest에 대응되는 Data가 CS에 저장되어 있는지 여부를 확인 위하여, Interest.c_ID와 CS에 저장되어 있는 entry의 Data.name을 비교 한다. 만약 두 값이 같은 entry가 CS에 저장되어 있다면, Face 0을 통해 대응되는 Data를 전송하고, 수신된 Interest 처리를 완료한다.

(3) 수신된 Interest에 대응되는 Data가 CS에 저장되어 있지 않다면, PIT에 Interest.c_ID에 대응되는 entry가 기록되어 있는지 확인한다. 만약 대응되는 entry가 PIT에 있다면, 해당 정보의 incoming Face

필드에 Face 0을 추가한 후 Interest 처리를 완료한다.

(4) 대응되는 entry가 PIT에 없다면, 수신된 Interest의 name 정보를 기반으로 FIB 테이블을 참조하여 수신된 Interest를 전송할 Face (ex. Face 2)를 선택한다. 이 때, Interest.name과 FIB 테이블의 entry를 비교하여 longest prefix matching 되는 정보에 해당하는 Face를 선택한다. 만약 Interest.name의 이름 요소가 ".a.c."를 포함하고 있다면, 콘텐츠 이름 요소가 해쉬 함수를 이용하여 인코딩 된 것이므로, 하위 도메인의 인코딩 이름 테이블(Encoded Name Table, ENT)을 참조하여 Face를 선택한다. ENT는 하위 네트워크 도메인 노드가 등록될 때, 상위 도메인 노드에 신고한다.

(5) PIT에 수신된 Interest의 c_ID와 incoming Face 정보를 추가한다.

(6) FIB에서 선택한 Face를 통하여 수신된 Interest를 전송한다.

(7) Face 2를 통하여 Data가 수신되면, PIT의 entry 중에서 수신된 Data의 name과 같은 c_ID 정보를 갖는 entry가 있는지 확인한다. 만약 대응되는 entry가 PIT에 기록되어 있지 않다면, 수신된 Data는 폐기처리 되고 Data 처리는 종료된다.

(8) PIT에서 대응하는 entry를 찾으면, CS에 수신된 Data를 저장한 후, 해당 entry의 incoming Face들을 통해서 Data를 전송한다. 전송이 완료되면 해당 entry를 PIT에서 삭제한다.

6. 성능분석

6.1 Security 분석

표 1은 CCN의 content-name 인코딩 기법들의 보안 기능을 비교하여 설명한다.

공개키 암호화를 기반으로 하는 FNE, CNE, FEO는 도메인 호스트의 공개키를 안전하게 활용하기 위하여 키 관리 시스템을 별도로 운영해야 한다. 특히, FEO의 경우 키 관리 시스템 외에도 Onion Router와 이를 관리하는 ORM을 추가로 운영해야 한다.

또한, Data는 콘텐츠 생성자의 전자서명을 포함해야만 한다. 그러므로 만약 중간 호스트들에 의해서 콘텐츠 이름이 변경 된다면, 반드시 Data를 새로 캡슐화 해야 한다. 이와 같은 경우, 인코딩 된 이름을 사용하는 Data는 여전히 인코딩 되지 않은 원래 이름을 포함하게 된다. 그러므로 Session Key를 사용하는 CNE를 제외한 FNE와 FEO는 Data 분석을 통해 인코딩 되지 않은 content-name을 분석해 낼 수 있다.

NAH와 INA는 Interest 인코딩을 위하여 단방향 해쉬 함수를 사용하기 때문에 별도의 시스템 구성을 추가로 요구하지 않는다. 그러나 NAH도 앞서 설명된 FNE/FEO와 같이 Data 캡슐화 절차가 필요하다. 그러므로 캡슐화 된 Data 분석을 통해 인코딩 되지 않은 content-name을 알아 낼 수 있다.

이에 반하여, INA는 Data에 FCN을 사용하기 때문에 Data 분석을 통하여 content-name 및 제공자에 대한 정보를 분석하기 어렵다.

6.2 성능 분석

이 절에서는 제안된 기법들의 성능을 비교 분석하기 위하여 Interest 전송량과 Interest에 대한 Data의 응답 시간을 비교한다. 이와 같은 분석은 다음과 같은 의미를 갖는다:

- Interest 전송량은 전체 네트워크 트래픽에 대한 분석을 의미하지만, CCN에서는 CS의 활용도와도 밀접한 관계가 있다.
- Interest/Data의 응답 시간은 CS의 활용도, 인코딩 기법의 프로세스 성능과 관계가 있다.

성능 분석을 위하여 4개의 최상위 도메인으로 구

Table 1. The Comparison of Encoding Function and Operation

Scheme	Name Illegibility		Operation
	Interest	Data	
FNE	○	×	• Name encapsulation to forward Data
CNE	○	○	
FEO	○	×	• The name encapsulation of onion routers to forward Data • Onion router modify PIT to handle en/decapsulation Interest
NAH	○	×	• Name encapsulation to forward Data
INA	○	○	• none

성된 네트워크 환경을 가정하고, 각각의 최상위 도메인은 다음과 같은 트리 형태의 네트워크 구조를 갖는다 : 트리를 구성하는 각각의 노드는 4개의 서브 노드를 가지고 트리의 깊이(depth)는 5로 한다. 또한, 각각의 노드는 계층화된 도메인 이름을 갖고 있으며 네트워크를 구성하는 전체 최단 노드 (leaf Node)들 중 12개의 노드에 각각 1개씩의 서로 다른 콘텐츠가 존재한다고 가정한다. 4개의 최상위 도메인 중 1개를 선택하여 선택된 도메인에 속한 최단 노드들이 랜덤하게 콘텐츠를 요청하는 372개의 Interest를 생성하도록 설정한다. 이와 같은 설정 하여서는 전체 Interest 중 평균 1/4은 내부 도메인에 저장된 3개의 콘텐츠를 요청하고, 나머지 3/4은 외부 도메인에 저장된 9개의 콘텐츠를 요청한다.

콘텐츠는 중간 노드에 임지 저장되지만, 안전성을 위하여 키와 Onion Router List는 중간노드에 임시 저장되지 않도록 설정한다.

그림 7는 콘텐츠 요청을 위해 네트워크에 전송되는 Interest의 수를 각 노드에서 측정하여 그 결과를 총합한 결과이다. “CCN”은 인코딩 기능이 없는 기본적인 CCN에서의 측정 결과이고, “NAH-1”과 “NAH-2”는 각각 NAH에서 Interest에 c_ID가 포함된 상황과 되지 않은 상황을 의미한다. “Content 요청 Interest”, “Key 요청 Interest”, “OR List 요청 Interest”는 각각 콘텐츠, 도메인 공개키, Onion Router List를 요청하는 Interest를 의미한다.

그림 7에서 보이듯이 암호화를 기반으로 인코딩 기술을 구현하는 경우, 네트워크 노드에 임시 저장된 콘텐츠 이용도가 낮아지기 때문에 Interest 전송 수가 CCN에 비하여 증가하는 것을 알 수 있다. 특히, 암호화를 위한 Key와 List를 추가로 요청해야 하기 때문에 Transmission Overhead가 크게 증가할 수 있다. 이에 반하여 해쉬 함수와 c_ID를 기반으로 인코딩 하는 경우, 기존 CCN과 유사한 Transmission Overhead 결과를 나타낸다.

그림 8은 Interest에 대한 Data의 응답시간 분석 결과이다. 그림 8-(A)는 Interest에 대한 Data의 평균 응답 시간을 측정한 결과이다. 공개키 기반의 암호화 기술을 이용하는 기술들이 해쉬 기반의 인코딩 기술에 비하여 노드 내부에서 패킷 처리에 많은 시간이 소요되는 것을 알 수 있다. 특히, 패킷 전송 시간에 비하여 노드 내부에서 패킷을 암/복호화 하고, 이를

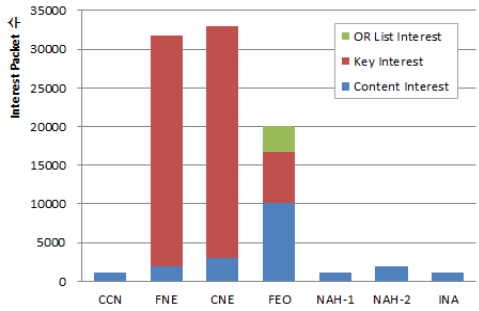
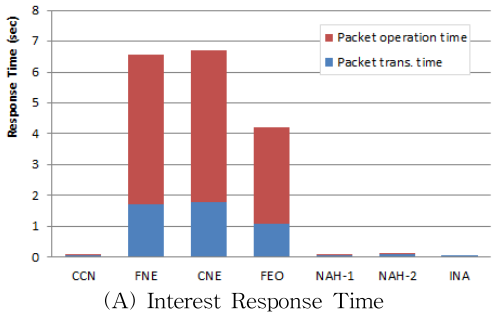
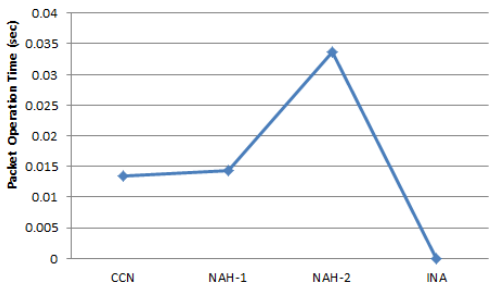


Fig. 7. Transmission Overheads.



(A) Interest Response Time



(B) Interest Operation Time

Fig. 8. Computation Overheads.

다시 캡슐화 하는 시간이 훨씬 많이 소요되는 것을 알 수 있다.

그림 8-(B)는 해쉬 기반의 인코딩 기술을 이용하는 제안들과 CCN의 패킷 처리 시간만을 분리하여 비교한 결과이다. INA는 Data의 name이 c_ID로 구성되어 있기 때문에 별도의 캡슐화 과정을 필요로 하지 않을 뿐만 아니라, CS에서 Data를 탐색하기 위하여 해쉬 값을 추가로 계산하지 않아도 되기 때문에 기존의 다른 기술들에 비하여 추가적인 절차로 인한 성능 저하가 거의 발생하지 않는다.

7. 결 론

CCN은 계층화된 콘텐츠 이름을 이용하여 Interest와 Data의 전송 경로를 결정하고, 이 계층화된 이름으로 네트워크 노드에 임시 저장된 콘텐츠를 검색하고 전송함으로써 네트워크 효율성을 높이도록 설계되었다. 그러나 전송되는 Interest/Data의 계층화된 콘텐츠 이름을 분석하면 사용자의 콘텐츠 이용에 관한 정보와 콘텐츠 생성자에 관한 정보를 알아낼 수 있다는 단점 때문에 콘텐츠 이름에 대한 인코딩 방안이 요구되고 있다.

본 논문은 전송되는 콘텐츠 이름 정보를 악의적인 사용자로부터 안전하게 보호하기 위하여 CCN의 기본적인 이름 구조를 수정하도록 제안하였다. Interest의 경우, 기존의 CCN 이름 구조인 HCN을 그대로 채택하여 사용하되, 단방향 해쉬 함수를 이용하여 Interest에 포함되어 있는 CCN 콘텐츠 이름을 보호하도록 제안하였다. Data의 경우, HCN 대신에 FCN을 사용하도록 수정하고, 이렇게 수정된 Data를 처리하기 위하여 PIT와 CS 운영 방안을 일부 수정하도록 제안하였다. 제안된 방식은 다음과 같은 두 가지 측면에서 CCN 연구에 기여한다.

첫째, 제안된 인코딩 방법은 전송량의 증가 없이 Interest/Data의 콘텐츠 이름 보호 기능을 제공함으로써 사용자 프라이버시 보호를 구현할 수 있다. 특히, NAH가 Data 분석을 통한 콘텐츠 이름 유출이 가능한 반면에 제안된 인코딩 기법은 Interest/Data에서 모두 안전하게 콘텐츠 이름을 보호할 수 있도록 설계 되었다.

둘째, 제안된 인코딩 기법은 기존 CCN 구조의 변경을 최소화 하도록 설계 되었다. PIT와 CS를 HCN이 아닌 FCN를 기반으로 운영하도록 구현하면 충분하다. 특히, 현재 CCN 오픈 소스에서도 PIT는 c_ID 값을 바탕으로 비교하기 때문에 Data에서 콘텐츠 이름을 분석하는 절차만 수정하면 된다.

REFERENCE

[1] D. Clark, "The Design Philosophy of the DARPA Internet Protocols," *ACM SIGCOMM Computer Communication Review*, Vol. 18, No. 1, pp. 106-114, 1988.

[2] B. Ahlgren, C. Dannewitz, C. Imbrenda, D. Kutscher, and B. Ohlmann, "A Survey of Information-Centric Networking," *IEEE Communications Magazine*, Vol. 50, No. 7, pp. 26-36, 2012.

[3] V. Jacobson, D. Smetters, J. Thornton, M. Plass, N. Briggs, and R. Braynard, "Networking Named Content," *Prceeding of 5th International Conference on Emerging Networking Experiments and Technologies*, pp. 1-12, 2009.

[4] The NDN Project Team, *Named Data Networking Project*, *NDN Technical Report*, NDO-0001, 2010.

[5] G. Bartolomeo and T. Kovacikova, *Identification and Management of Distributed Data*, CRC Press, New York, 2013.

[6] NSF FIA PI meeting: NDN team presentation, http://named-data.net/publications/1105fia-pi-ndn/, April, 2014

[7] S. DiBenedetto, P. Gasti, G. Tsudik, and E. Uzun, "ANDaNA: Anonymous Named Data Networking Application," *ISOC Symposium on Network and Distributed System Security*, pp. 1-18, February 2012.

[8] D. Kim, "Content Centric Networking Naming Scheme for Efficient Data Sharing," *Journal of Korea Multimedia Society*, Vol. 15, No. 9, pp. 1126-1132, 2012.

김 대 엽

1997년 3월~2000년 2월 고려대학교 대학원 수학과 이학박사
1997년 9월~2000년 3월 ㈜텔리맨 CAS팀 연구원
2000년 4월~2002년 8월 시큐아이닷컴 정보보호연구소 차장
2002년 9월~2012년 2월 삼성종합기술원 전문연구원
2012년 3월~현재 수원대 정보보호학과 조교수
관심분야 : 보안프로토콜, 네트워크/시스템 보안, 콘텐츠 보안, 미래 인터넷 보안