

# TICN 백본망 성능향상을 위한 CCN 적용 캐싱전략

박흥순<sup>\*,1)</sup> · 권태욱<sup>1)</sup>

<sup>1)</sup> 국방대학교 컴퓨터공학과

## Caching Strategies of Content-Centric Networking for the Performance of TICN Backbone

Heungsoon Park<sup>\*,1)</sup> · Taewook Kwon<sup>1)</sup>

<sup>1)</sup> Department of Computer Science and Engineering, Korea National Defense University, Korea

(Received 31 March 2014 / Revised 30 July 2014 / Accepted 15 August 2014)

### ABSTRACT

Military communications have limited resources such as poor bandwidth and insufficient link capacity. Numerous hosts in conventional IP networks, including the Tactical Information Communication Network (TICN), the tactical communication system of the Army, focus heavy traffic on a single server via end-to-end connections and in which data delivery latency is caused by exchanging routing information. Content-Centric Networking (CCN), one framework of Information-Centric Networking (ICN), is a novel Internet paradigm in which data can be accessed by the name of the content instead of the location information. This paper provides content caching strategies for applying CCN to military tactical IP networks based on TICN environment.

Key Words : TICN(전술정보통신체계), Military Communications(군 통신), Information-Centric Networking(정보 중심 네트워킹), Content-Centric Networking(콘텐츠 중심 네트워킹), Caching(캐싱)

### 1. 서론

IT 기술의 발전은 상용통신의 비약적인 발전 뿐 아니라 전쟁수행개념 및 방식 등 군 전반적으로 영향을 주고 있다. 육군은 전술세대 네트워크 시스템인 전술정보통신체계(TICN)를 구축함으로써 모든 단말을 All-IP기반하 음성통신과 대용량의 데이터 통신지원을 계

획하고 있다<sup>[1]</sup>. 하지만, 상용 통신환경과 달리 군 전술 통신환경은 제한적인 환경에서 운용되며, 작전수행에 필요한 다양한 능력을 요구받는다. TICN 또한 과거 통신체계에 비해 다양한 체계(정찰·감시, 지휘통제체계, 복합정밀타격체계 등) 및 각종 C4I체계 연동을 요구받고 있으며, 이에 따라 제한된 환경에서 단말간 네트워크 경로 설정 및 서버에 집중되는 트래픽 부하 등 네트워크 자원소모가 클 것으로 판단된다.

과거 호스트 기반의 통신구조에서 탈피하여 정보 중심의 미래 인터넷 구조를 제안하는 연구를 Information-

\* Corresponding author, E-mail: heungsoon.park@gmail.com  
Copyright © The Korea Institute of Military Science and Technology

Centric Networking(ICN)이라고 한다<sup>[2]</sup>. ICN은 정보의 위치보다는 정보 그 자체에 중점을 갖는 네트워킹 패러다임으로써 장치에 주소를 부여하여 통신하던 기존 방식과는 달리, 정보자체에 이름을 부여 데이터를 전송한다. 이를 통해 현재 인터넷의 IP 네트워크에서의 호스트 대 호스트연결로 인한 라우팅 오버헤드 및 지연, 특정 서버로 집중되는 트래픽 증가 등의 문제점을 해결 수 있다. Content-Centric Networking(CCN)<sup>[3]</sup>은 V. Jacobson에 의해 제안된 대표적인 ICN 모델 중 하나로써 기존 IP네트워크의 라우터와 유사한 동작방식을 사용하고, IP 전송계층과 연동이 용이하다는 특징이 있다. 게다가, 최근에는 군 전송통신과 같은 제한적인 환경에서 CCN을 활용하는 연구가 제안되고 있는 실정이다<sup>[4,5]</sup>.

이에 본 연구에서는 TICN체계 백본망에 CCN 모델을 적용하여 성능향상을 연구했던 기존 연구<sup>[6]</sup>를 바탕으로 군 특성에 맞는 CCN 적용 캐싱전략을 제안한다. 본 논문의 주요 연구내용은 다음과 같다. 첫째, 군 통신체계에 CCN을 적용함으로써 기존 IP 네트워크체계보다 성능이 향상됨을 보였으며, 둘째, CCN을 적용하기 위한 다양한 캐싱전략을 제안하였다.

이 논문의 구성은 다음과 같다. 2장에서는 정보중심 네트워킹인 ICN과 대표적인 모델인 CCN을 설명하고, 3장에서는 CCN 적용에 필요한 군 특성에 맞는 캐싱 전략을 제안한다. 4장에서는 실험을 통해 기존 IP기반의 방식과 캐싱전략을 비교 설명하며, 마지막 5장에서 결론을 맺는다.

## 2. 관련 연구

### 2.1 Information-Centric Networking(ICN)

ICN은 기존의 IP기반 네트워킹의 기본구조를 바꾸어, 인터넷의 문제점을 해결하고자 시작된 새로운 통신 패러다임이다<sup>[2]</sup>. IP기반의 네트워킹은 데이터에 대한 송신자와 수신자가 존재하고, 데이터 송수신 이전에 호스트 또는 라우터간 제어정보의 교환으로 경로 정보를 확인해야 한다.

반면, ICN은 송수신자 대신 데이터를 생성한 생성자(publisher)와 데이터를 수신하는 구독자(subscriber)의 개념을 도입한다. 이에 네트워크는 정보를 가지고 있는 호스트가 아닌 정보 자체에 관심을 둔다. 즉, 호스트의 IP 주소 대신 데이터의 고유 이름을 사용하여

라우팅을 수행하며, 라우터는 필요한 경우 특정 데이터를 저장하고, 데이터를 요구하는 인근의 새로운 정보 요청자들에게 데이터를 배포하는 방식을 사용한다. ICN과 관련된 대표적인 연구로는 CCN<sup>[3]</sup>, DONA<sup>[7]</sup>, PURSUIT<sup>[8]</sup> 등이 있다.

### 2.2 Content-Centric Networking(CCN)

CCN은 ICN 이론 중 각광받고 있는 이론 중 하나로써 미국 실리콘벨리를 중심으로 활발히 연구중에 있다.

#### 2.2.1 메시지 구조

CCN은 기본적으로 Fig. 1과 같이 두 개의 메시지 패킷을 가지고 있다. Interest packet은 정보를 원하는 정보 요청자에 의해 발생되며 이에 대한 응답으로 Data packet이 전송된다. 그림에서와 같이 Interest packet에는 정보 생성자의 주소가 없다. 즉, 생성자가 최초 데이터를 생성할 때 만든 이름을 가지고 해당 정보를 요구한다. 이는 IP 네트워크에서 송·수신자의 위치주소를 가지고 정보를 포워딩하고 라우팅하는 것과는 근본적인 차이를 보인다. CCN은 데이터 포워딩에 있어서 콘텐츠 이름을 전적으로 사용하며, 따라서, 콘텐츠 이름은 생성자로 하여금 유일하게 명명되어야 한다.

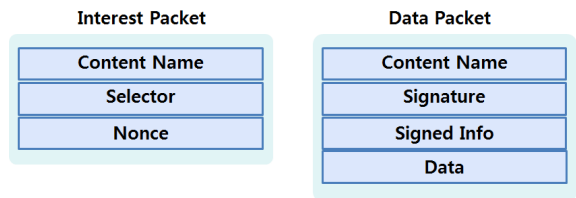


Fig. 1. Two message types of CCN

#### 2.2.2 포워딩 방법

각각의 CCN 라우터는 Content Store(CS), Pending Interest Table(PIT), Forwarding Information Base(FIB)라고 불리는 세 개의 주요 데이터 구조로 구성된다. CCN 노드에서의 포워딩 방법은 Fig. 2와 같이 Interest packet과 Data packet이 CS, PIT, FIB에서 순차적으로 처리되어 이루어진다.

Interest packet이 도착하면, 먼저 CS 내에 해당 콘텐츠의 Data packet이 캐싱되어 있는지 확인한다. 여기서 매칭되는 Data packet이 있다면 Interest packet이 들어

은 인터페이스로 해당 Data packet을 전송하고 Interest packet은 버려진다. 만약 CS에 매칭되는 콘텐츠가 없다면 다음 단계인 PIT를 검색하여 사전에 해당 콘텐츠를 요청했던 정보가 있는지 확인한다. 만약 해당 콘텐츠를 요청했던 정보가 있다면, 해당 PIT entry에 Interest packet이 도착한 인터페이스 정보를 추가하고 해당 Interest packet의 임무는 종료된다. 이는 불필요하게 동일한 Interest packet이 재전송되는 것을 방지하여 네트워크 자원을 절약할 수 있게 한다. 만약 PIT에도 콘텐츠를 요청했던 정보가 없다면, 마지막으로 FIB를 검색한다. FIB는 기존의 IP 라우팅 테이블 검색 방식과 동일한 Longest Prefix Matching(LPM)을 통해 해당 콘텐츠 이름 prefix를 찾는다. 이 prefix는 콘텐츠를 생성한 정보생성자와 연결되는 인터페이스 정보와 매칭되며, 기존의 IP라우팅테이블과 흡사하다. 만약 매칭되는 FIB entry가 있다면 해당 인터페이스로 Interest packet을 포워딩하고, PIT에 해당 되는 콘텐츠 정보와 인터페이스 정보를 추가한다. 이는 앞서 설명한 대로 나중에 동일한 이름의 Interest packet이 수신될 경우 중복전송을 방지하기 위한 절차이다. 만약 CS, PIT, FIB 모두 매칭되는 이름 정보가 없다면 해당 Interest packet은 버려진다.

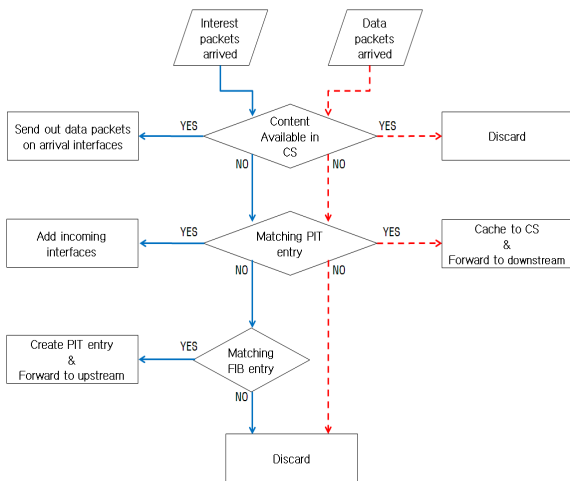


Fig. 2. Packet processing in CCN

Data packet의 처리는 Interest packet보다 간단하다. Data packet은 Interest packet이 포워딩 되면서 생성된 PIT entry를 통해 생성된 인터페이스를 연결한 경로를 따라 정보 요청자에게 전달된다. 최초 Data packet이

CCN 노드에 도착하면, 노드는 해당 CS를 검색하고, 동일한 콘텐츠 정보가 캐싱되어 있다면 해당 Data packet은 버려진다. 이미 동일한 콘텐츠가 있기 때문에 중복해서 저장하지는 않는다. 캐싱되어 있지 않고 PIT entry에 요청된 정보가 있다면, CS에 해당 Data packet을 캐싱하고 PIT에 등록된 인터페이스로 Data packet을 보낸다. Fig. 2는 CCN 노드에서의 패킷처리 개념도를 보여준다.

### 3. 제안하는 CCN 적용 캐싱전략

군의 특수성을 고려하여 군에 적합한 CCN 적용 캐싱전략 방안을 다음과 같이 동적 캐싱(DC : Dynamic Caching)와 통계적 데이터 기반의 캐싱(SC : Statistical Caching)으로 구분하여 제안한다.

#### 3.1 동적 캐싱(DC)

DC는 CCN 라우터가 상급부대로 요청되는 다량의 Interest packet을 탐지하여 일정량 이상의 임계치에 도달하였을 때, 되돌아오는 Data packet을 일시적으로 캐싱하여 경로상의 중간 라우터가 정보를 제공하는 방식을 사용한다. 임계치는 하나의 Interest packet 송신 시 하나의 Data packet이 수신되는 CCN 라우팅 원리를 이용하여 업링크 인터페이스의 대역폭을 감시하여, 회신되는 데이터 스트림의 패킷손실이 발생이 예상되는 시점을 판단하는 방식을 택한다. 링크 대역폭을 탐지하여 패킷손실을 예상하는 스키마는 C. Yi의 연구<sup>[9]</sup>를 활용하여 임계치를 판단하였다.  $L_i$ 를 해당 업링크의 인터페이스에서의 Interest 전송률이라고 한다면, 임계치를 판단하는 공식은 식 (1)과 같다.

$$L_i = \alpha \times C_i / \bar{S}_i \tag{1}$$

이 때,  $C_i$ 는 해당 인터페이스의 업링크 용량을 말하며,  $\bar{S}_i$ 는 데이터 패킷 스트림에서 데이터 패킷의 사이즈를 측정할 평균 데이터 사이즈를 의미한다. ( $\alpha$ 는 보정상수) DC에서 해당 라우터는 상급부대로 가는 업링크 대역폭과 수신되는 데이터 량을 실시간으로 측정하여 임계치에 도달하면, 서버로부터 해당 정보를 캐싱한다. Fig. 3은 DC라우터의 캐싱 알고리즘을 보여준다.

**Algorithm 1.** DC algorithm

```

1: if (pktType == Interest) {
2:   Integer requestCount = reqCount.get(data);
3:   if (requestCount == null) {
4:     reqCount.put(data, 1);
5:   }
6:   else {
7:     requestCount++;
8:     reqCount.put(data, requestCount);
9:   }
10: if (findData(data) == true)
11:   createDataPacket(data);
12: else
13:   prevNode.RequestContent(data);
14: }
15: else if (pktType == Data) {
16:   if (findData() == true)
17:     Discard(data);
18:   else
19:     if (count.get(data)/t > threshold) {
20:       insertData(data);
21:       forwardDataPacket(data);
22:     }
23: }
    
```

Fig. 3. Dynamic caching algorithm

### 3.2 통계적 캐싱(SC)

SC는 군 특성상 상급부대가 생성하여 전파하는 자료의 빈도와 크기가 하급부대에서 요청 및 보고하는 자료의 건수보다 크다는 실증적인 훈련자료의 통계분석에 기인한다. 즉, 서버가 최초 자료생성과 동시에 상급부대 전파사항에 해당되는 Flag 데이터가 탐지되면 해당 SC라우터에 사전 캐싱되어 정보가 제공된다.

이는 군의 작전 및 훈련 패턴 상 계획된 절차에 의한 자료 생성 및 유통이 이루어진다는 가정 하에 수립된 것으로, 사전에 축적된 데이터를 바탕으로 한다.

따라서, 본 연구에서는 SC의 실증적인 알고리즘 설계를 위해 00년 000훈련간 00000체계의 데이터 처리량을 Table 1과 같이 분석하였다.

훈련 간 00000체계의 000개 업무 중 000개 업무가 사용되었으며, 이 중 A, B, C, D 업무가 전체 자료처리량의 60~70%를 점유하고 있었다. 그 중 상급부대의 전파정보는 A, C, D이며 전체 처리율에 약 43%를 차지하였고, 나머지 정보는 대부분이 하급부대에서 요청하거나 보고하는 정보였다. 이처럼 상급부대 전파정보

를 자료생성과 동시에 SC라우터에 전파하여 사전에 캐싱하면 각 단말은 서버까지 정보를 요청할 필요 없이 가까이 있는 SC라우터에서 정보를 제공받을 수 있어 네트워크 성능을 향상 시킬 수 있다. Fig. 4는 SC 알고리즘을 보여준다.

Table 1. Processing rate of top 5 task categories

Rank	Category	Processing rate	Remarks
1	A	25.15%	Upper
2	B	20.10%	Lower
3	C	9.85%	Upper
4	D	8.35%	Upper
5	E	8.00%	Lower
...	...	...	...

**Algorithm 2.** SC algorithm

```

1: if (init == true) {
2:   cache.put('a', 1);
3:   cache.put('c', 1);
4:   cache.put('d', 1);
5:   init = false;
6: }
7: if (pktType == Interest) {
8:   Integer requestCount = reqCount.get(data);
9:   if (requestCount == null) {
10:    reqCount.put(data,1);
11:   }
12:  else {
13:    requestCount++;
14:    reqCount.put(data, hitCount);
15:  }
16:  if (findData() == true)
17:    createDataPacket(data);
18:  else
19:    prevNode.RequestContent(data);
20: }
21: else if (pktType == Data) {
22:   if (findData() == true)
23:     Discard(data);
24:   else
25:     forwardDataPacket(data);
26: }
    
```

Fig. 4. Statistical caching algorithm

#### 4. 성능평가

##### 4.1 시뮬레이션 환경

실험을 위한 환경구성은 00급이하 TICN백본망을 기준으로 실제 노드구성과 동일하게 토폴로지를 구성하고 실험하였다. 시뮬레이션은 CCNx<sup>[10]</sup>를 기반으로한 가상 테스트베드를 구축하였으며, 실제 훈련자료를 바탕으로 가상의 데이터를 입력하여 실험을 하였다.

실험 비교 대상은 기존 TICN체계에서 운용되는 IP 환경을 그대로 활용하였으며, DC와 SC알고리즘을 라우터에 적용하여 비교 측정하였다. 성능평가요소로는 서버에 집중되는 부하량과 전체 네트워크에서의 오버헤드를 트래픽의 부하를 변화시키면서 측정하였다. Table 2는 시뮬레이션 환경설정을 보여준다.

Table 2. Simulation parameter

Parameters	Settings
Number of routers	00
Number of servers	0
Number of hosts	00
Interest packet size	48 bytes
Data packet size	200 Kbytes
Average request rate	76 requests/min

##### 4.2 실험결과 및 성능분석

###### 4.2.1 서버의 부하량

서버의 부하량은 각 단말이 서버로 요청하는 Interest Packet에 대한 응답으로 제공 되어지는 데이터의 처리 건수로 상대적 부하량을 측정하였다. Fig. 5에서 보는 바와 같이, 동일한 양의 자료를 요청했을 경우 각 단말에서 요청하는 자료에 대한 서버에 집중되는 부하량이 기존 IP기반의 통신방식에 비해 DC가 약 37% 정도, SC가 약 83% 정도 감소했다. 또한 요청자료의 양을 상대적으로 증가시켜 보았을 때, 그 격차는 조금씩 늘어가는 양상을 보인다. 이는 기존 IP네트워크 라우팅 방식이 end-to-end 연결을 하여 각 단말들의 요청이 하나의 단일 서버로 집중되기 때문이다. 반면에, DC와 SC는 서버로 집중되는 정보요청을 중간 라우터가 대신 자료를 캐싱하여 각 단말들에게 제공함으로써 그만큼의 서버 부하량을 줄일 수 있었다. DC는 서버로

가는 상향링크로 가는 대역폭을 감시하여 임계치 이상의 요청이 수신되었을 때 동작되므로, 사전에 축적된 데이터를 가지고 먼저 작동되는 SC보다는 서버에서 제공되는 데이터의 부하가 크다고 할 수 있다.

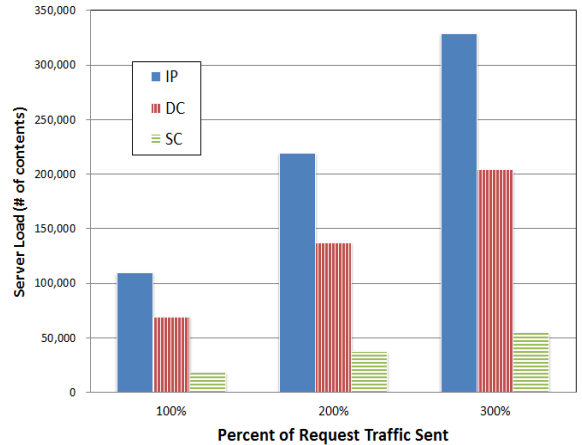


Fig. 5. Sever load vs request traffic

###### 4.2.2 전체 네트워크 통신량

전체 네트워크 통신량은 각 단말에서 요청하는 Interest Packet에 대해 발생되어지는 중간 라우터의 네트워크 부하량 및 서버에서 네트워크로 전송되어지는 자료의 양을 합친 총량으로 산출하였다. Fig. 6에서와 같이 DC는 IP 네트워크 대비 약 21% 정도의 전체 네트워크 통신량을 감소시켰지만, SC에서는 약 42% 정도로 통신량을 감소시켰다. 이는 IP네트워크의 경우 Interest Packet의 포워딩이 서버까지 연결되는 경로를 따라서 이루어지므로 DC나 SC보다 상대적으로 많은 제어정보발생을 초래한 것이다. DC는 SC보다는 상대적으로 임계치가 적용되는 대역폭의 한계점까지 Interest Packet을 포워딩함으로써 네트워크 트래픽을 더 많이 발생시켰다. 이는 대역폭 자원을 더 많이 활용할 것인가 아니면 메모리자원을 더 많이 활용할 것인가하는 trade-off 문제로서, 또 다른 비용분석이 요구 될 것이다.

종합적으로 판단하였을 때, IP 환경에서 나타나는 서버의 트래픽 집중과 전체 네트워크의 부하를 CCN을 적용함으로써 감소시킬 수가 있었다. 사전에 축적된 데이터를 가지고 판단 적용한 SC가 일반적인 상황에서의 대역폭을 감시하는 DC보다는 좋은 성능을 보일 수 있으나, 이는 어디까지나 균 특수 환경에서 기

인한 정보예측에 의한 실증적 자료에 근거한 것으로, 일반적인 상황에서는 DC를 적용해야 할 것이다.

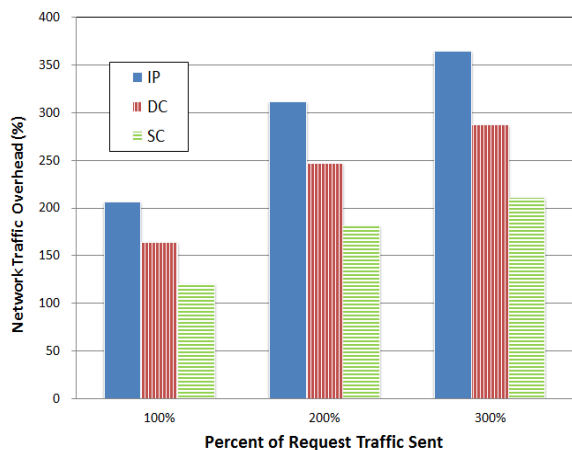


Fig. 6. Network overhead vs request traffic

## 5. 결론

본 논문에서는 미래 네트워크 개념으로 주목받고 있는 ICN 모델 중 대표적인 CCN 모델을 기존 IP 네트워크체인 TICN에 적용하기 위한 캐싱전략을 수립하였다. 서버로가는 상향링크의 부하를 탐지하여 임계치에 따른 동적 캐싱전략인 DC를 수립하였고, 기존 훈련결과에 의한 데이터 분석을 통해 상급부대 전과 정보를 사전에 캐싱하는 SC를 설계하였으며, TICN에 근거한 모델링과 실험을 실시하였다. 실험결과에서 보듯이 군 특수환경을 고려한 2가지 CCN 적용방안은 군 TICN체계의 제한된 대역폭과 전송용량의 문제를 해결하고 서버의 집중되는 트래픽을 분산처리 함으로써 전체 네트워크 자원을 효율적으로 운용할 수 있을 것으로 기대된다.

CCN은 IP 네트워크가 가지고 있지 않은 다양한 이

점을 갖고 있지만, 현재 전통적으로 사용되어지고 있는 IP네트워크체계를 CCN으로 전환하기에는 큰 비용 부담이 발생한다. 차후 IP장비에 CCN을 접목하여 승수효과를 누릴 수 있는 연구가 필요할 것이다.

## References

- [1] ROKA Headquarters, "TICN Operational Concept," 2007.
- [2] B. Ahlgre, C. Dannewitz, et al. "A Survey of Information-Centric Networking," Communications Magazine, IEEE Vol. 50, No. 7, pp. 26-36, 2012.
- [3] V. Jacobson, D. K. Smetters, et al., "Networking Named Content," Commun. ACM, Vol. 55, No. 1, pp. 117-124, 2012.
- [4] S. Y. Oh, D. Lau, M. Gerla, "Content Centric Networking in Tactical and Emergency MANETs," in IFIP Wireless Days (WD), Oct. 2010.
- [5] B. Etefia and L. Zhang, "Named Data Networking for Military Communication Systems," Aerospace Conference, IEEE, 2012.
- [6] S. S. Park, H. S. Park, T. W. Kwon, "A Study of TICN Backbone Routing Applying Content Centric Networking," Korea Computer Congress 2013, Jun. 2013.
- [7] T. Koponen, et al., "A Data-Oriented (and beyond) Network Architecture," ACM SIGCOM Comput. Commun. Review, Vol. 37, No. 4, Oct. 2007.
- [8] Publish-Subscribe Internet Technology. <http://www.fp7-pursuit.eu/PursuitWeb>
- [9] C. Yi, et al., "Adaptive Forwarding in Named Data Networking," ACM SIGCOM Comput. Commun. Review, Vol. 42, No. 3, 2012.
- [10] CCNx project, <http://www.ccnx.org>