MPC: 콘텐츠에 대한 인기 기반 캐싱 전략 중심 네트워크

세자르 베르나르디니, 토마스 실버스톤, 페스토르 올리비에

MPC: 인기 기반 캐싱 전략 콘텐츠 중심 네트워크

세자르 베르나르디니 †, Thomas Silverston † 및 Olivier Festor †
Universite de Lorraine, LORIA, UMR 7503, Vandoeuvre-les-Nancy, F-54506, France †
Inria, Villers-les-Nancy, F-54600, France 이
메일: {cesar .bernardini, thomas.silverston, olivier.festor}@inria.fr

개요 - 콘텐츠 중심 네트워킹(CCN)은 최근 콘텐츠를 대규모로 제공하는 유망한 아키텍처로 부상했습니다. 패킷 주소가 콘텐츠의 위치가 아니라 이름을 지정하는 명명된 데이터를 기반으로 합니다. 그런 다음 전제는 전달 경로를 따라 네트워크 노드에 콘텐츠를 캐시하는 것입니다. 따라서 CCN의 중요한 기능은 노드의 캐시를 관리하는 것입니다.

본 논문에서는 CCN 네트워크에 적용된 새로운 캐싱 전략인 MPC(Most-Popular Content)를 제시한다. 인기 있는 콘텐츠만 캐싱함으로써 광범위한 시뮬레이션 실험을 통해 MPC가 더 적은 콘텐츠를 캐시할 수 있는 동시에 더 높은 캐시 히트를 달성하고 CCN의 기존 기본 캐싱 전략보다 성능이 우수함을 보여줍니다.

I. 서론

인터넷은 현재 대부분 콘텐츠에 액세스하는 데 사용됩니다.
실제로 2000년대에는 파일 공유를 위한 P2P 트래픽이 전체 인터넷 트래픽의 약 80%를 차지했습니다. 오늘날 유튜브와 같은 비디오 스트리밍 서비스는 인터넷 트래픽에서 가장 중요한 부분을 차지합니다. 2016년까지 모든 형태의 비디오(TV, VoD 및 P2P)를 합하면 전 세계 소비자 트래픽의 약 86%가 될 것으로 예상됩니다[1].

인터넷은 호스트 간 통신(IP)을 위해 설계되었으며 여전히 호스트 간 통신(IP)에 초점을 맞추고 있지만 사용자는 소스 위치가 아닌 실제 콘텐츠에만 관심이 있습니다. 따라서 CCN[2], NetInf[3] 및 Pursuit/PSIRP[4]와 같은 새로운 ICN(Information-Centric Networking Architecture)이 대규모로 효율적인 콘텐츠 배포에 높은 우선순위를 부여하여 정의되었습니다. 이러한 모든 새로운 아키텍처 중에서 CCN(Content Centric Networking)은 연구 커뮤니티에서 상당한 관심을 끌었습니다[5].

CCN은 패킷 주소가 콘텐츠의 위치가 아니라 이름을 지정하는 명명된 데이터를 기반으로 하는 네트워크 아키텍처입니다. IP에 정의된 호스트라는 개념은 더 이상 존재하지 않습니다. CCN에서는 현재 인터넷의 경우와 같이 전용 서버에서 콘텐츠를 검색하지 않습니다. 전제는 콘텐츠가 네트워크를 통해 배포 경로를 통과할 때 노드당 캐싱 기능을 포함하여 콘텐츠 전달을 향상시킬 수 있다는 것입니다.

따라서 콘텐츠는 복제되고 네트워크의 다른 지점에 위치하므로 들어오는 요청에 대한 가용성이 높아집니다.

CCN의 중요한 기능은 캐시 여부와 캐시가 가득 찬 경우 교체할 요소를 각각 결정하는 캐싱 전략 및 교체 정책으로 노드의 캐시를 관리하는 것입니다. 최근 작업은 주로 바이너리 트리[7] 또는 교육용 ISP[8]와 같은 여러 토폴로지에 대한 캐시 교체 정책[6] [?] (MRU, MFU, LRU, FIFO)에 중점을 두었습니다 . 심지어 개선을 통해 비교

다중 경로 지원[9] 또는 이론적 캐시 속성 조사[10]와 같은 다른 기능. 캐싱 전략과 관련하여 일부 연구에서는 높은 캐싱 성능을 달성하기 위해 엄청난 캐시 메모리가 필요하다는 것을 보여주고[11], 다른 [12] 캐싱을 적게 하면 노드의 하위 집합에만 콘텐츠를 저장하여 유사한 수준의 성능을 달성할 수 있다고 주장합니다. 배달 경로를 따라. 따라서 CCN 네트워크에 적합한 효율적인 캐싱 전략을 설계하는 것이 필수적입니다.

이 백서에서는 CCN 네트워크를 위해 설계된 새로운 캐싱 전략인 MPC(Most Popular Content)를 제시합니다. 경로의 모든 노드에 모든 콘텐츠를 저장하는 대신 MPC는 인기 있는 콘텐츠만 캐시합니다. MPC는 CCN 기본 전략보다 적게 캐싱하지만 네트워크 내 캐싱 성능을 향상시키는 동시에 리소스 소비를 줄입니다.

본 논문의 알림은 다음과 같이 구성된다. 섹션 II에서는 CCN을 위한 새로운 캐싱 전략인 MPC에 대해 설명합니다.
섹션 III은 시뮬레이션 환경을 제시하고 섹션 IV는 MPC의 성능을 보여줍니다. 마지막으로 섹션 V에서는 논문을 마무리하고 향후 작업을 제시합니다.

II. 가장 인기 있는 캐싱

이 섹션에서는 먼저 CCN에 대한 간략한 개요를 제공합니다.
그런 다음 새로운 캐싱 전략 MPC를 제시합니다.

가. CCN 개요

CCN 아키텍처는 대부분 Interest와 Data라는 두 가지 프리미티브를 기반으로 합니다. 소비자는 네트워크에서 관심 메시지를 보내 콘텐츠를 요청합니다. 요청을 듣고 데이터가 있는 모든 노드는 데이터 메시지로 응답을 발행할 수 있습니다.

따라서 콘텐츠는 소비자에게 전송되고 전달 경로의 모든 노드는 데이터를 캐시할 수 있습니다. 명확하게 정의된 캐싱 전략이 없는 경우 CCN 기본 캐싱 전략은 항상 전달 경로의 모든 노드에 콘텐츠를 저장합니다(항상 전략). 이 전략은 다른 전략과 비교할 때 최상의 결과를 보여줍니다[8]. 전달 경로를 따라 노드의 하위 집합에만 콘텐츠를 저장하여 캐싱을 적게 해도 비슷한 캐시 성능을 얻을 수 있습니다[12]. 첫 번째 접근 방식은 인기 있는 콘텐츠를 인기 없는 콘텐츠로 대체할 수 있고 두 번째 캐싱 덜 접근 방식을 기반으로 인기 있는 콘텐츠만 캐싱하면 높은 성능을 달성하는 동시에 리소스를 절약할 수 있다고 주장합니다.

따라서 노드가 인기 있는 콘텐츠만 캐시하는 CCN의 새로운 캐싱 전략인 MPC, Most Popular Content를 설계했습니다.

B. 가장 인기 있는 캐싱 전략

MPC에서 모든 노드는 각 콘텐츠 이름에 대한 요청 수를 로컬에서 계산하고 쌍(콘텐츠 이름, 인기도 수)을 Popularity Table에 저장합니다. 콘텐츠 이름이 로컬 인기 임계값에 도달하면 콘텐츠 이름에 인기 있는 것으로 태그가 지정되고 노드가 콘텐츠를 보유하고 있으면 이웃 노드가 새로운 제안 프리미티브를 통해 콘텐츠를 캐시하도록 제안합니다. 이러한 제안 메시지는 리소스 가용성과 같은 로컬 정책에 따라 수락되거나 수락되지 않을 수 있습니다.

제안 과정 이후에는 콘텐츠의 인기도가 떨어질 수 있으므로 동일한 콘텐츠가 이웃에게 플러딩되는 것을 방지하기 위해 재설정 값에 따라 인기도 카운트를 다시 초기화합니다.

MPC 전략은 CCN 노드 요구 사항에 직접 영향을 미칩니다. 필요한 CCN 캐시 공간 외에도 MPC는 Popularity Table을 저장하기 위한 추가 공간이 필요합니다. 예를 들어 테이블에 100만 항목을 유지한다는 것은 콘텐츠 이름당 1023B를 사용하고 인기도 수에 1B를 사용하는 1GB RAM 메모리를 의미합니다(계산을 단순화하기 위해 이름에 고정 길이를 사용함). 공정한 비교를 수행하기 위해 전용 캐싱 메모리를 MPC 테이블과 공유할 것으로 예상합니다.

C. MPC 예시 시나리오

이 섹션에서는 MPC 워크플로를 명확히 하기 위한 예를 제시합니다. 이 예에서 노드는 네트워크 주변에 위치한 콘텐츠를 요청하고 있으며 기본 CCN 전략과 비교하여 MPC 전략이 작동하는 방식을 설명합니다. 이 예는 그림 1에 나와 있으며 그림 1(a)는 콘텐츠 요청의 시나리오와 순서를 설명합니다.

그림 1(b)와 1(c)는 각각 MPC와 CCN 캐싱 전략을 적용한 후의 최종 상태를 보여준다. 우리의 시나리오에서 3개의 노드 A, B, C는 처음에 노드 D에 저장된 인기 콘텐츠 d1을 요청할 것입니다 . 노드 A는 또한 처음에 노드 E에 저장된 인기 없는 콘텐츠 e1을 요청할 것입니다 그림 1(b)의 MPC 전략을 사용하여 노드가 A가 콘텐츠 e1 에 대한 관심 메시지를 전송

하면 경로 [A, C, D, E]를 따라 모든 노드의 인기도 테이블에서 e1의 인기도가 증가합니다. 즉, 노드 A, C, D 및 E에서 e1의 인기도가 1로 설정됩니다. 마찬가지로 B가 콘텐츠 d1 에 대한 관심 메시지를 보낼 때 경로 [B, C, D]를 따라 d1 의 인기도는 1로 설정됩니다.

콘텐츠 d1이 인기 있는 콘텐츠이므로 A와 C는 차례로 d1에 대한 관심 메시지를 보냅니다. A의 관심 메시지는 C와 D에서 d1의 인기도를 2로 높일 것입니다(경로 [C, D]를 통해). 마지막으로 C는 d1을 요청하여 노드 C와 D에서 d1의 인기도를 3으로 높입니다 . 이 시점에서 D는 콘텐츠 d1을 보유하는 유일한 노드입니다. d1의 인기도가 인기 임계값에 도달했다고 가정하면 D는 이웃 C와 E에게 제안 메시지를 전파합니다. 따라서 C와 E는 콘텐츠를 캐시하고 다음 요청에 응답하고 콘텐츠 d1을 전송할 수 있습니다. 예를 들어 A에 가까운 관심 있는 노드가 있는 경우 해당 요청은 D가 아닌 C로 리디렉션되어 콘텐츠를 얻기 위한 홉 수를 줄입니다. d1이 이웃에게 전파된 후 D 는 동일한 콘텐츠가 이웃에게 지속적으로 플러딩되는 것을 방지하기 위해 재설정 값에서 d1의 인기도를 재설정합니다 .

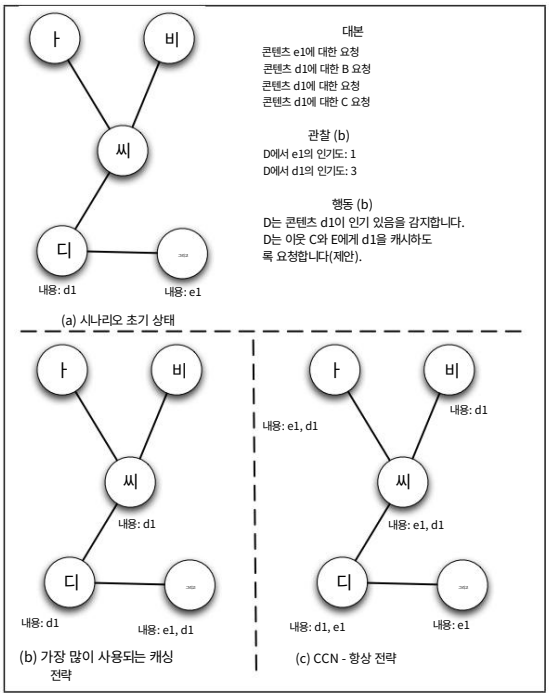


그림 1: MPC 작업 흐름 예 . (a) 네트워크 및 시나리오의 초기 상태 (b) MPC 최종 상태; (c) 항상 전략을 사용하는 CCN의 최종 상태.

그림 1(c)는 기본 항상 캐싱 전략을 사용한 후 CCN 최종 상태를 설명합니다. A가 첫 번째 요청을 보내면 콘텐츠 e1이 경로 [A, C, D, E]를 따라 모든 노드에 캐시됩니다. 그런 다음 시나리오에 따르면 콘텐츠 d1은 B의 첫 번째 요청 후 경로 [B, C, D]를 따라 캐시됩니다. A의 요청은 C를 통해서만 전달되고 d1은 A에 캐시 됩니다. 마지막으로 노드 C는 이전 요청에서 캐시되었으므로 이미 콘텐츠를 저장합니다.

이 시나리오를 요약하기 위해 CCN은 노드 A, C 및 D에서 e1 의 세 복제본과 노드 A, B 및 C에서 d1 의 세 복제본을 계산합니다 (그림 1(c)). MPC는 노드 C와 E에서 d1 의 두 복제본만 계산 하고 e1은 복제된 적이 없습니다(그림 1(b)). 분명히 인기 있는 콘텐츠만 캐싱함으로써 우리가 제안한 전략인 MPC는 CCN Always 전략에 비해 메모리, 대역폭 및 캐싱 작업 수와 같은 리소스를 절약하고 있습니다. 노드 E에서 d1 의 복제는 E가 그것을 요구하지 않기 때문에 기회주의적이었습니다(그림 1(b)). 그러나 노드 E는 인기 있는 콘텐츠이므로 d1 에 대한 향후 요청을 받을 수 있습니다.

III. 시뮬레이션 설정

새로운 전략 MPC를 평가하기 위해 Omnet++ 프레임워크를 통해 C++로 개발된 청크 수준 CCN 시뮬레이터인 ccnSim[8]을 사용합니다. 시뮬레이터는 처음에 다양한 전략과 시나리오로 CCN 성능을 평가하도록 설계되었습니다.

MPC와 CCN 간의 교차 비교를 촉진하기 위해 ccnSim 고유 매개변수 및 네트워크 토폴로지를 포함하여 시뮬레이션 환경을 자세히 설명합니다.

모수	값 10	모수	값
실행/시뮬레이션		요청 비율	50 요청/초
워밍업 단계	진실	청크 크기	10kb
다중 경로	비활성화된 캐싱	교체	LRU
라우팅 전략	가장 가까운	캐싱 전략	언제나

표 I: ccnSim 매개변수 설정

	세그먼트 N δ 코어 11 2.54	σγ Δ[ms] D	
애빌린	0.19 11.13 8		
타이거2	지하철 22 3.60 0.17 0.11 5		
거대한	집계 22 3.40 0.41 2.59 4		
디텔레콤	핵심 68 10.38 1.28 17.21 3		
레벨3	코어 46 11.65 0.86 8.88 4		
나무	참조 15 2.54 0.21 1.00 3		

표 II: 네트워크 토폴로지의 속성[8]

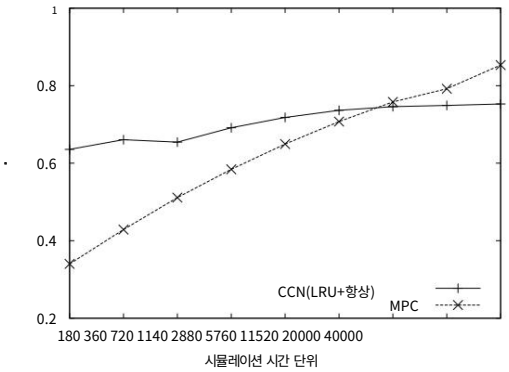


그림 2: 시뮬레이션 시간 평가

ccnSim 구성 가능 매개변수는 표 I에 설명되어 있습니다. 모든 시뮬레이션에서 CCN 노드는 항상 LRU 캐싱 교체와 함께 캐싱 전략 정책. 우리는 이러한 정책을 선택합니다. CCN을 위한 최고의 성능 [8]. 다른 매개 변수가 표시됩니다. 라우팅 정책, 청크 크기 또는 요청 속도와 같은 테이블에서.

ccnSim에서 파일의 인기도는 MZipf 배포 기능[11] [8]에 따라 모델링되었습니다. 우리의 실험에서, 파일을 모델링하기 위해 동일한 분포 함수를 선택합니다. 요청합니다. 108개 파일의 Youtube와 같은 카탈로그의 경우 다음을 의미합니다. 요청의 99%는 6,000개의 가장 인기 있는 파일.

ccnSim에는 여러 네트워크 토폴로지가 포함되어 있으므로 그런 다음 표 II에 설명된 이러한 토폴로지를 사용합니다.

실험 섹션 전체에서 두 가지 시나리오를 사용합니다. 표 III에 정의되어 있습니다. 첫 번째 시나리오 Ssmall 은 시나리오입니다. CCN 평가 [7]에 일반적으로 사용되며 다음으로 구성됩니다. 파일당 평균 102개의 청크가 있는 104개 파일 의 카탈로그 ; 캐시 크기는 노드당 103 청크로 고정됩니다 . 두 번째 Syoutube 는 Youtube와 같은 카탈로그가 있는 대규모 시나리오입니다. 파일당 103개의 청크가 있는 108개의 파일 포함 : 대략 1P 카탈로그 B. 총 청크에 대한 캐시 비율이 로 설정 $\frac{\text{씨}}{\text{|에프 |에프}}$ = 10-5 ; 이 시나리오에서 캐시 크기는 다음으로 설정됩니다. 106 청크(10GB).

그런 다음 각 실험에 대해 무작위로 하나의 카탈로그를 설정합니다. 토폴로지에 8개의 요청자 노드가 있습니다. 우리는 10 공연 시뮬레이션당 실행하고 평균값을 제공합니다.

IV. MPC 의 성능 평가

이 섹션에서는 시뮬레이션 결과를 제시합니다. 우리 다음에 따라 MPC의 성능을 평가합니다. 지표:

- Cache Hit Ratio: 모든 캐시 적중률을 얻을 확률 요청자에서 캐시 노드까지의 경로를 따라;
- 스트레치: 데이터 청크가 가지고 있는 CCN 홉 수 서버 저장과 관련하여 네트워크를 이동했습니다. 원본;
- 캐시된 요소 비율: 캐시된 요소의 비율 관심사의 총 수와 관련하여;
- Diversity[8]: 파일에 저장된 서로 다른 청크의 비율 캐시.

매개변수 설명 F avg. 파일당 청크		스몰 앤아웃튜브	
파일 수 캐시당 저장된 청크 수 MZipf 지		102 103	
|에프|	수 매개변수 MZipf 고	104 108	
씨	원 매개변수	103 106	
ㅏ		1.5 1.5	
큐		0 0	

표 III: 시뮬레이션 실험 시나리오

MPC가 Popularity Table과 같은 새로운 매개변수를 도입함에 따라 인기 임계값 및 재설정 값, 이의 첫 번째 부분 섹션은 추론하기 위해 이러한 매개변수의 조정을 수반합니다. 적절한 기준. 시뮬레이션 시간도 공부했다. 두 번째로, 우리는 항상 전략을 사용하는 MPC 전략 및 CCN.

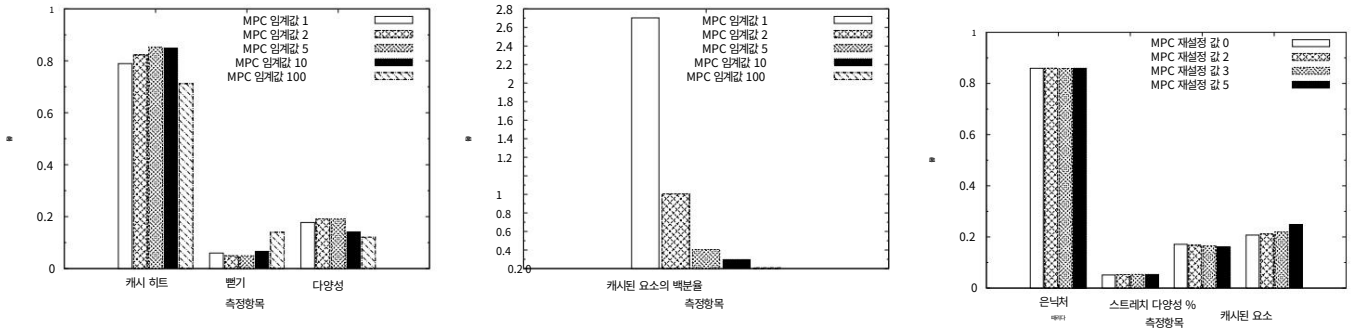
A. MPC 파라미터 조정

MPC의 매개변수 선택은 다음에 따라 수행되었습니다. Syoutube 시나리오를 통한 광범위한 시뮬레이션 프로세스 그리고 트리 토폴로지.

시뮬레이션 시간은 다양한 전략을 정확하게 시뮬레이션하기 위한 중요한 매개변수입니다. ccnSim 시뮬레이터는 다음에 따라 캐시에 청크를 배치하여 워밍업합니다. 전략의 통계적 시뮬레이션. 인기도 표로 워밍업되지 않은 MPC는 캐싱을 시작할 시간도 필요합니다. 대중적인 콘텐츠. 따라서 MPC에 필요한 시간을 평가합니다. CCN을 능가합니다. 시뮬레이션 시간을 최대 40,000까지 다양화했습니다. 시뮬레이션 시간 단위(STU). CCN 및 MPC 캐시 적중 결과는 그림 2에 나와 있습니다.

분명히 짧은 시간 시뮬레이션의 경우 11,520 STU까지, CCN은 MPC보다 높은 캐시 적중률에 도달했습니다. 그런 다음 CCN의 캐시 적중률은 안정적으로 유지되는 반면 MPC의 캐시 적중률은 증가하고 CCN을 능가합니다. 예를 들어 40,000 STU에서 캐시 적중률은 MPC의 경우 85%이고 CCN의 경우 75%에 불과합니다. ~ 안에 이 백서의 나머지 부분에서는 장시간 시뮬레이션에 중점을 두고 시뮬레이션 시간을 40,000 시뮬레이션 시간 단위로 설정합니다.

Popularity Threshold 매개변수와 관련하여 그림 3a와 그림 3b는 임계값이 다른 결과를 보여줍니다. 가치. 분명히 5로 설정된 Popularity Threshold는 가장 높은 Cache Hit, Diversity 및 다른 제품보다 낮은 Stretch



(ㅏ) (비)

그림 3: 인기 임계값 매개변수

그림 4: 재설정 값 매개변수

테스트 값(그림 3a). 동시에 캐시된 요소의 비율은 20%로 매우 낮게 유지됩니다 (그림 3b). 이 임계값이 더 낮을수록(예: 1) CCN 기본 전략보다 최대 2.7배 더 많이 캐시할 수 있습니다.

앞에서 설명한 것처럼 콘텐츠 이름이 인기 임계값에 도달하고 이웃을 통해 확산되면 이 콘텐츠에 대한 향후 요청이 있을 경우 이 값을 재설정해야 합니다. 그림 4에서 볼 수 있듯이 캐시 적중률이 테스트된 모든 값에 대해 동일한 비율이지만 재설정 값을 0으로 설정하는 것이 캐시에서 동일한 콘텐츠의 다양성이 약간 더 높고 확장 및 복제가 적기 때문에 더 적합합니다.

MPC는 콘텐츠 이름의 인기도를 추적하기 위해 테이블을 사용하므로 캐시 노드의 메모리를 낭비하지 않고 적절한 테이블 크기를 선택해야 합니다. 그림 5는 항목이 2,500만 개인 인기 테이블이 캐시 히트, 다양성이 가장 높고 스트레치를 낮은 값으로 유지함을 보여줍니다. 콘텐츠 이름을 1KB로 가정하면 2.5GB의 메모리를 의미합니다. 그런 다음 캐시 크기 공간은 캐싱 리소스의 25%를 나타내는 MPC 요구 사항(Popularity Table)과 공유됩니다(SY outube 시나리오의 경우 캐시의 총 10GB인 경우 2.5GB).

이 평가 부분을 요약하기 위해 다음 실험을 위해 MPC 매개 변수를 조정합니다. 시뮬레이션 시간은 40,000으로 설정되고 Popularity Threshold, Reset Value 및 Popularity Table 크기는 각각 5, 0 및 2.5GB로 설정됩니다.

B. MPC 대 CCN 항상 전략

공정한 MPC와 CCN/Always를 비교하기 위해 먼저 [8]에서와 같이 Ssmall 시나리오로 동일한 실험을 수행했습니다. 다양한 인기도 분포 함수(MZipf 지수를 0.5에서 2.5로 변경)로 총 청크 수에 대한 캐시의 비율을 변경하고 캐시 적중률을 관찰합니다. 결과는 각각 CCN과 MPC에 대한 그림 6a와 그림 6b에 나와 있습니다. 차트의 유사성은 모든 구성에 대해 유사하게 작동하는 방식을 보여줍니다. 그러나 표 IV는 MPC에 대한 캐시된 요소의 비율을 보여줍니다. CCN에서는 항상 정책을 사용하여 콘텐츠가 통과하는 모든 노드에서 콘텐츠를 캐시합니다. CCN의 비율은 항상 1입니다. MPC의 경우 비율은 38%를 초과하지 않습니다. 이는 MPC가 CCN보다 훨씬 적은 수의 요소를 캐시한다는 의미입니다.

	캐시 비율			
MZipf 지수 1/10 1/100 1/1000 1/10000				
0.5				
1	0.22	0.09		
	0.33	0.31	0.04	
1.5 2 2.5	0.37	0.38	0.15	

표 IV: MPC에 대한 캐시된 요소의 비율. 이 비율은 CCN/항상 정책의 경우 1입니다. (는 0에 가까운 값입니다.)

이제 모든 토폴로지에서 SY outube 시나리오를 사용하여 MPC와 CCN을 비교할 것입니다 . 그림 7a에서 분명히 MPC 캐시 적중률은 CCN보다 높고 85% 이상입니다.
CCN이 Level3 또는 DTelecom 토폴로지에서 최고 결과에 도달하더라도 MPC는 여전히 CCN을 능가합니다. 캐시된 요소의 비율은 그림 7b에 나와 있습니다. 항상 정책이 있는 CCN의 경우 콘텐츠는 항상 캐시되며 100% 비율에 도달합니다. MPC 전략은 Tree, Abilene, Geant 및 Tiger 토폴로지(약 20%)에 대해 CCN보다 훨씬 적은 콘텐츠를 캐싱하여 캐싱 작업을 덜 수행하고 메모리를 절약합니다. MPC는 DTelecom 및 Level3 토폴로지(각각 80% 및 60%)로 더 많은 콘텐츠를 캐시하지만 여전히 CCN보다 낮은 속도입니다. 이 두 토폴로지에 대한 캐시된 요소의 증가는 노드의 높은 연결 정도(표 II의 10개 또는 11개 이웃) 때문입니다. 이렇게 고도로 연결된 토폴로지에서 MPC는 더 많은 이웃에게 콘텐츠를 저장하라는 제안 메시지를 보내기 때문에 경로를 벗어난 캐싱입니다.

공간 제한으로 인해 Stretch 메트릭에 대한 Figure는 제공하지 않습니다. MPC와 CCN은 유사한 결과를 나타냅니다. CCN의 스트레치는 모든 토폴로지에서 약 10%이고 MPC는 약간 낮은 8%입니다. 인기 있는 콘텐츠만 캐싱함으로써 MPC 전략은 여전히 요청자와 가까운 콘텐츠를 캐싱할 수 있습니다.

다양성 메트릭은 그림 7c에 나와 있습니다. CCN 다양성은 모든 토폴로지에 대해 28%에서 35% 범위이며 MPC 다양성은 3%에서 18%로 훨씬 낮습니다. 다양성과 관련하여 MPC는 인기 있는 콘텐츠만 캐시하도록 설계되어 노드 캐시에서 청크의 다양성을 제한하기 때문에 MPC가 CCN보다 덜 효율적일 것으로 예상되었습니다. 그러나 CCN과 MPC 모두 높은 다양성을 달성하지 못했으며,

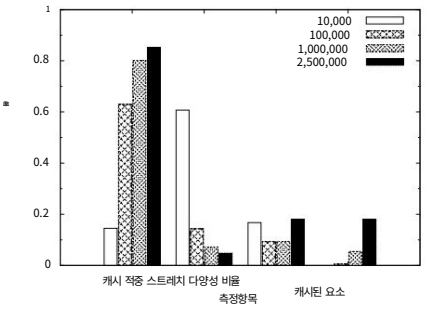
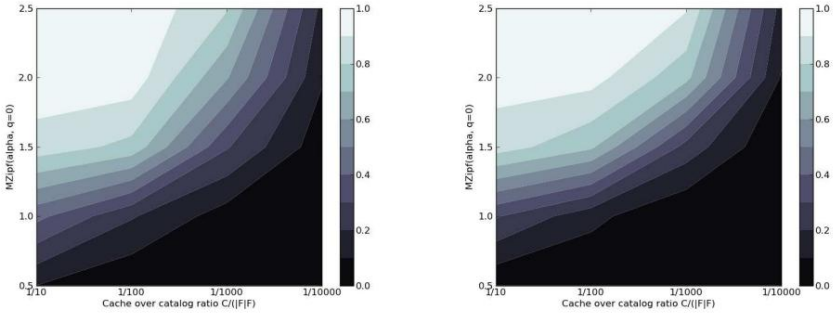
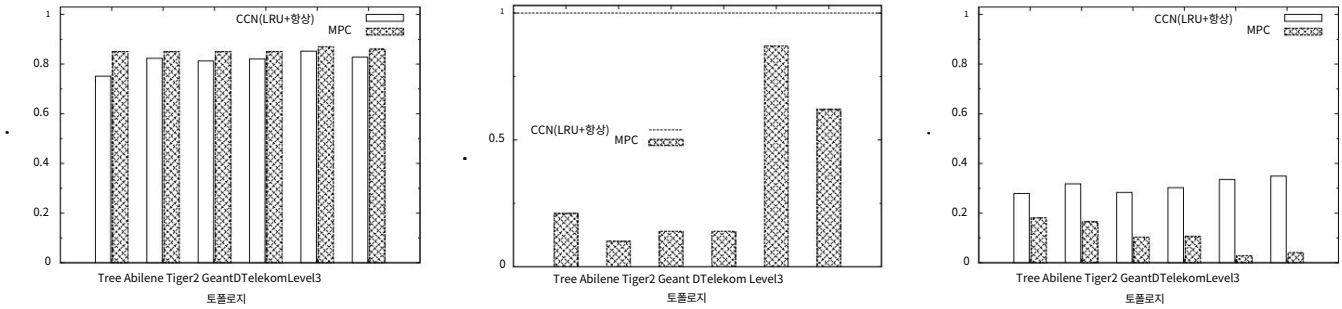


그림 5: 인기도 테이블 크기 매개변수



(a) CCN - LRU/항상 (나) MPC

그림 6: Cache Hit의 등고선 플롯



(a) 캐시 적중률 (b) 캐시된 요소의 비율 (c) 다양성

그림 7: 서로 다른 토폴로지에서의 MPC 대 CCN(LRU/Always)

CCN 네트워크.

마지막으로 시뮬레이션 실험은 MPC 전략이 더 높은 캐시 적중률을 달성하기 때문에 CCN Always 전략보다 성능이 우수함을 보여줍니다. 또한 MPC는 콘텐츠를 더 적게 캐시하고 메모리와 같은 리소스를 절약하며 캐시 작업 수를 줄입니다.

V. 결론

본 논문에서는 CCN 네트워크를 위한 새로운 캐싱 전략인 MPC를 제시하였다. MPC 전략은 인기 있는 콘텐츠만 캐시하고 각 노드에서 캐시 부하를 줄입니다. 우리의 시뮬레이션 실험은 MPC가 CCN/항상 기본 전략을 능가하는 것으로 나타났습니다. MPC는 더 높은 캐시 적중률을 달성하고 여전히 요소의 복제 수를 크게 줄입니다. 더 적은 데이터를 캐싱하고 캐시 적중률을 개선함으로써 MPC는 네트워크 리소스 소비를 개선합니다.

향후 작업으로 우리의 전략이 이름 기반 라우팅 프로토콜을 연구하는 기반이 될 수 있을 것으로 기대합니다. 제안 기반 메커니즘이므로 노드 간에 콘텐츠를 관리하고 인기도를 예측하며 콘텐츠를 목적지로 라우팅하도록 조정하는 것이 가능합니다.

승인

저자는 재정적 지원을 위해 Conseil Regional de Lorraine에 감사드립니다.

참조

[1] Cisco, "Cisco 비주얼 네트워킹 인덱스: 글로벌 모바일 데이터 트래픽 예측 업데이트, 2011-2016," Cisco, Tech. 2012년 2월 의원.

[2] V. Jacobson, DK Smetters, JD Thornton, MF Plass, NH Briggs 및 RL Braynard, "Networking named content", 신흥 네트워킹 실험 및 기술에 관한 제5회 국제 회의 진행, ser. 코넥스트 '09. 뉴욕, 뉴욕, 미국: ACM, 2009, pp.

1–12. [온라인]. 이용 가능: http://doi.acm.org/10.1145/1658939.1658941 [3] C. Dannewitz, "NetInf: 미래 인터넷을 위한 정보 중심 설계", 2009.

[4] N. Fotiou, GC Polyzos 및 D. Trossen, "게시 구독 인터넷 아키텍처 설명."

[5] [온라인]. 이용 가능: http://www.ccnx.com [6]
K. Katsaros, G. Xylomenos 및 GC Polyzos, "Multicache: 오버레이
정보 중심 네트워킹을 위한 아키텍처입니다."

[7] I. Psaras, RG Clegg, R. Landa, WK Chai, G. Pavlou, "모델링 및 ccn-캐싱 트리 평가," 네트워킹에 관한 제10회 국제 IFIP TC 6 회의 절차 - 제1부, 서. 네트워킹'11. 베를린, 하이델베르크: Springer-Verlag, 2011, pp. 78–91. [온라인]. 이용 가능: http://dl.acm.org/citation.cfm?id=2008780.2008789 [8] D. Rossi 및 G. Rossini, "캐싱 성능 of content centric networksunder multi-path routing (and more)," Telecom ParisTech, 기술

의원, 2011.

[9] , "ccn 다중 경로 관심 전달 전략 평가", in CCNxCon, 2012.

[10] DSM Elisha J. Rosensweig 및 J. Kurose, “정상 상태에서 캐시 네트워크”, IEEE Infocom, 2012.

[11] C. Fricker, P. Robert, J. Roberts 및 N. Sbihi, "콘텐츠 중심 네트워크에서 캐싱 성능에 대한 트래픽 혼합의 영향", 2012년 2월. [온라인]. 이용 가능: http://hal.inria.fr/hal-00666169 [12]
WK Chai, D. He, I. Psaras, and G. Pavlou, “Cache ”less for more” in information-centric networks,” 네트워킹 (1), 2012, pp. 27–40.