

효율적인 레이블 스위칭을 위한 캐쉬 테이블 관리

(Cache Table Management for Effective Label Switching)

김 남 기 * 윤 현 수 **
(Namgi Kim) (Hyunsoo Yoon)

요 약 인터넷이 급속도로 발전되면서 트래픽이 폭발적으로 증가하여 현재 라우터에는 많은 부하가 걸리고 있으며, 스위칭 기술은 라우팅보다 빠르게 데이터를 전송하면서 라우터 병목 현상을 줄일 수 있다. 따라서 IP 라우팅에 스위칭 기술을 접목한 레이블 스위칭 네트워크가 출현하게 되었다.

레이블 스위칭 기술 중 데이터 기반 레이블 스위칭에서는 흐름 분류와 캐쉬 테이블 관리가 중요하다. 흐름 분류는 패킷을 스위칭으로 전송할 것인지, 그렇지 않으면 기존 라우팅으로 전송할 것인지를 결정하는 부분인데, 이러한 흐름 분류를 위한 정보와 레이블 스위칭을 위한 정보를 저장하는 곳이 캐쉬 테이블이다. 이런 캐쉬 테이블이 크면 클수록 많은 패킷을 스위칭 할 수 있고 스위칭 설정 비용을 줄일 수 있기 때문에 캐쉬 테이블은 클수록 좋으나 라우터 자원에 의해 제약이 받으므로 적절한 캐쉬 테이블 관리가 필요하다. 따라서 효율적인 캐쉬 테이블 관리를 위해 인터넷 트래픽 특성을 고려한 캐쉬 대체 기법에 관한 연구가 요구되어 진다.

본 논문에서는 레이블 스위칭을 위한 여러 캐쉬 대체 기법들을 소개하고 제안한다. 첫째로 스위칭 용량에 한계가 없을 경우 FIFO(First In First Out), LFC(Least Flow Count), LRU(Least Recently Used) 기법을 평가하고 우선순위 LRU, 가중치 우선순위 LRU 기법을 제안한다. 둘째로 스위칭 용량에 한계가 있을 경우 LFC-LFC, LFC-LRU, LRU-LFC, LRU-LRU 기법을 평가하고 LRU-가중치 LRU 기법을 제안한다. 본 논문에서는 한계가 없을 경우 가중치 우선순위 LRU 기법이, 한계가 있을 경우 LRU-가중치 LRU 기법이 가장 나은 성능을 보였다.

Abstract The traffic on the Internet has been growing exponentially for some time. This growth is beginning to stress the current-day routers. However, switching technology offers much higher performance. So the label switching network which combines IP routing with switching technology, is emerged.

Especially in the data driven label switching, flow classification and cache table management are needed. Flow classification is to classify packets into switching and non-switching packets, and cache table management is to maintain the cache table which contains information for flow classification and label switching. However, the cache table management affects the performance of label switching network considerably as well as flow classification because the bigger cache table makes more packet switched and maintains setup cost lower, but cache is restricted by local router resources. For that reason, there is need to study the cache replacement scheme for the efficient cache table management with the Internet traffic characterized by user.

So in this paper, we propose several cache replacement schemes for label switching network. First, without the limitation at switching capacity in the router, we introduce FIFO(First In First Out), LFC(Least Flow Count), LRU(Least Recently Used) scheme and propose priority LRU, weighted priority LRU scheme. Second, with the limitation at switching capacity in the router, we introduce

LFC-LFC, LFC-LRU, LRU-LFC, LRU-LRU scheme and propose LRU-weighted LRU scheme. Without limitation, weighted priority LRU scheme and with limitation, LRU-weighted LRU scheme showed best performance in this paper.

* 본 연구는 첨단정보기술 연구센터를 통하여 과학재단의 지원을 받았음.

† 비 퇴 원 : 한국과학기술원 전자전산학과
ngkim@camars.kaist.ac.kr

** 중 신 퇴 원 : 한국과학기술원 전자전산학과 교수

hyoon@camars.kaist.ac.kr

논문접수 : 2000년 5월 22일
심사완료 : 2001년 4월 3일

1. 서론

웹과 같은 편리한 인터페이스의 개발로 인터넷은 대중화되면서 급속도로 성장하고 있다.

2000년 현재 전 세계적으로 약 6천만개의 컴퓨터가 인터넷과 연결되어 있으며 그 수가 매년 50%씩 증가하는 추세이다 [1]. 또한 컴퓨터 수의 증가와 비례해 인터넷을 통해 전송되는 트래픽의 양도 빠르게 증가하고 있다. 앞으로도 주문형 비디오 오디오 서비스, 인터넷 화상전화와 같은 멀티미디어 서비스와 가상 대학, 전자 도서관 등의 연구 교육 서비스, 그리고 사이버 은행, 인터넷 쇼핑물, 전자 상거래와 같은 인터넷 비즈니스에 의한 요구로 인터넷은 지속적으로 발전되고 트래픽의 양도 급격하게 늘어날 전망이다.

하지만 인터넷 트래픽 증가는 현재의 인터넷 망에 많은 부담을 주고 있다. 특히 앞으로 개발될 인터넷 서비스는 많은 양의 데이터를 좀 더 빨리 처리할 것을 요구하기 때문에 조만간 기존 인터넷망은 한계에 이를 것이다. 그러므로 많은 양의 데이터를 빠르게 처리할 수 있는 차세대 인터넷 망 구성이 시급한 문제로 제기 되고 있다.

차세대 인터넷 망을 구성하기 위해서는 네트워크 망의 속도 향상과 대역폭 확장이 필요하다. 그러기 위해서는 패킷을 어디로 보낼지 결정하는 라우터(router)의 성능 향상이 필수적인데 기존의 전통적인 인터넷 라우터는 라우팅(routing) 과정에서 IP 주소(Internet Protocol address)를 이용한 가장 긴 프리픽스 매칭(longest prefix matching)을 소프트웨어적으로 사용하기 때문에 성능 향상에 한계를 가지고 있다.

이러한 라우터의 성능을 향상시키고자 하는 노력은 크게 두 가지 방향으로 진행되고 있다. 첫번째 방법은 전통적인 라우터의 구조를 그대로 받아 들이고 이를 병렬화하는 것이다. 즉 기존의 가장 긴 프리픽스 매칭을 소프트웨어적으로 그대로 사용하는 대신 이를 병렬적으로 수행하여 성능을 향상시킨 것이다. 대표적인 예로 기가비트 라우터 구조가 여기에 속한다 [2, 3]. 두번째 방법은 레이어 3(layer3)인 IP 계층(IP layer)에서 가변적인 IP 주소를 이용하여 라우팅을 수행하지 않고 ATM과 같은 스위치 계층(레이어2)에서 고정된 길이의 레이블(label)을 이용해 스위칭 하는 것이다 [4]. 이를 레이블 스위칭(label switching)이라고 하는데 레이블 스위칭을 사용하는 라우터는 전송하고자 하는 데이터 패킷에 있는 일정한 크기의 레이블만을 보고서 패킷이 어디로 전송될지 결정할 수 있고 고정된 길이의 레이블을 이용해 하드웨어로 구현이 쉽기 때문에 높은 데이터 처리 속도를 얻을 수 있다 [4].

레이블 스위칭 기법은 데이터 기반(data driven) 방식과 제어 기반(control driven) 방식으로 분류할 수 있다 [5].

제어 기반 레이블 스위칭은 데이터가 전송될 수 있는 모든 가상 경로(virtual connection)에 레이블을 할당하고 전체 데이터를 스위칭 기법으로 전송한다. 그러나 가상 경로의 수가 레이블 개수보다 크기 때문에 모든 가상 경로에 서로 다른 레이블을 할당할 수 없으므로 여러 개의 가상 경로들 중 중간 경로에서 같은 출발지와 목적지를 가지는 가상 경로에 대해서는 하나의 레이블로 전송하는 가상 경로 합병(virtual connection merging)이 필요하다. 그러므로 가상 경로 합병은 중간 목적지에 도달하면 데이터가 어느 방향으로 갈지 결정하기 위해 IP 라우팅을 재 수행해야 하는 단점을 지닌다.

그림 1은 데이터 기반 레이블 스위칭을 개략적으로 나타낸 그림이다.

제목:
dataDriven
작성한 사람:
Tgrf-3.0-p13 by William Chia-Wei Cheng (william@cs.UCLA.edu)
미리 보기:
이 EPS 그림은 미리 보기 그림을 포함하지 않고
저장되었습니다.
설명:
이 EPS 그림은 포스트스크립트 프린터를
제외한 다른 프린터에서는
인쇄되지 않습니다.

그림 1 개략적인 데이터 기반 레이블 스위칭

데이터 기반 레이블 스위칭은 레이블 스위칭 네트워크로 패킷이 도달하면 흐름 분류(flow classification)를 통해 각 패킷을 스위칭으로 전송할지 아니면 기존 IP 라우팅으로 전송할지 결정한다. 스위칭으로 전송하기로 결정된 패킷의 IP 주소에는 이에 대응되는 레이블이 할당되고 이 레이블을 통해 데이터를 스위칭 기법으로 전달한다. 이 때 흐름 분류를 위한 정보와 스위칭을 위한 정보를 저장할 필요가 있는데 그 정보를 저장하는 곳이 캐쉬 테이블이다.

데이터 기반 레이블 스위칭에서 흐름 분류와 캐쉬 테이블 관리의 매우 중요하다. 흐름 분류는 어떠한 데이터에 대해 기존의 IP 라우팅으로 전송할지 스위칭으로 전송할지 판별한다. 하나의 레이블을 사용하더라도 많고 중요한 데이터를 전송하는 패킷에 대해 레이블을 할당해 스위칭 하면, 작고 중요하지 않은 데이터에 대해 레이블을 할당해 스위칭 하는 것보다 훨씬 효율적이기 때문에, 어떻게 흐름 분류하는가가 데이터 기반 레이블 스위칭 네트워크의 성능에 매우 큰 영향을 미친다 [6, 7].

데이터 기반 레이블 스위칭에서 흐름 분류와 더불어 중

요한 것이 캐쉬 테이블(cache table) 관리이다. 유입되는 데이터 패킷에 대해 흐름 분류를 수행하기 위해서는 라우터에서 같은 목적지를 가지는 데이터 패킷에 대한 정보를 수집하고 관리해야 한다. 그리고 스위칭 전송을 위해 IP 주소에 대응하는 레이블을 저장하고 있는 스위칭 정보도 필요하다. 이런 흐름 분류 정보와 스위칭 정보를 저장하는 곳이 캐쉬 테이블인데 캐쉬 테이블의 크기는 전체 네트워크의 성능에 매우 중요한 영향을 미친다 [4]. 왜냐하면 캐쉬가 크면 보다 많은 패킷들을 스위칭 시킬 수 있고, 더 많은 가상 연결(virtual connection)을 유지할 수 있어서 가상 연결 설정 횟수를 줄일 수 있기 때문이다.

하지만 캐쉬 테이블은 흐름 분류를 수행하는 라우터의 자원에 의해 제한을 받고 있기 때문에 무한히 커질 수 없다. 그리고 현재 IP 주소의 크기는 32비트로 구성되어 있고 앞으로 사용될 IPv6 [8]에서는 128비트를 사용하기 때문에 모든 정보를 캐쉬에 저장하려면 너무 많은 자원을 낭비하게 된다. 따라서 캐쉬 엔트리가 다 채워졌을 때 새로운 목적지를 가지는 패킷이 들어올 경우 이를 처리하기 위한 적절한 캐쉬 대체 기법(cache replacement scheme)이 필요하다. 즉 앞으로 어떤 캐쉬 엔트리가 플로우가 될지 모르는 상황에서 새로운 데이터를 어느 캐쉬 엔트리(entry)와 대체해야 가장 효율적일 수 있는가에 대한 연구가 필요한 것이다. 하지만 아직까지 데이터 기반 레이블 스위칭에서 캐쉬 대체 알고리즘에 대한 연구와 분석은 거의 전무한 상태이다. 기존의 가상 메모리(virtual memory)를 위해 사용되는 캐쉬 대체 알고리즘을 데이터 기반 레이블 스위칭에 적용할 수도 있겠으나, 프로그램에 의해 요구되는 데이터의 특성을 고려한 가상 메모리 캐쉬 대체 알고리즘을, 사용자에게 의해 요구되어지는 특성을 가진 인터넷 트래픽을 고려하지 않은 채 바로 레이블 스위칭에 적용한다는 것은 적절하지 않다.

본 논문에서는 데이터 기반의 레이블 스위칭에서 캐쉬 테이블을 관리하기 위해 어떤 캐쉬 대체 알고리즘이 사용될 수 있는지 연구하고 새로운 캐쉬 대체 알고리즘을 제안한다. 우선 캐쉬 전체 엔트리를 레이블 스위칭 할 수 있을 만큼 자원이 풍부한 경우, 기본적인 방법으로 FIFO(First Come First Out), LFC(Least Flow Count), LRU(Least Recently Used) 기법에 대해 성능을 평가해 보고 이를 개선한 우선순위 LRU(Priority LRU) 기법과 가중치 우선순위 LRU(weighted priority LRU) 기법을 제안하고 평가한다. 그리고 레이블 스위칭 자원의 제약으로 캐쉬의 일부분만을 레이블 스위칭 할 수 있는 경우 적용될 수 있는 LFC-LFC, LRU-LFC, LRU-LRU 기법에 대해 성능을 평가해 보고 이를 개선한 LRU-가중치 LRU 기법을 제

안하고 평가한다.

본 논문의 구성은 다음과 같다. 2장에서는 보다 구체적으로 데이터 기반 레이블 스위칭에 대해 알아보고 흐름 분류를 위한 캐쉬 관리에 대해 살펴 본다. 3장에서는 레이블 스위칭에 적용할 수 있는 여러 캐쉬 대체 기법을 제시하고 분석해 이를 토대로 보다 개선된 캐쉬 대체 기법을 제안한다. 4장에서는 3장에서 제안된 캐쉬 대체 기법에 대한 성능 평가를 수행하고 그 결과를 분석한다. 마지막으로 5장에서는 결론을 맺고 향후 연구 과제에 대해 기술한다.

2. 관련 연구

본 장에서는 데이터 기반 레이블 스위칭의 동등 모델(peer model)과 겹침 모델(overlay model)에 대해 알아보고 캐쉬 테이블의 중요성을 파악한다.

2.1 데이터 기반의 레이블 스위칭

데이터 기반 레이블 스위칭은 디폴트(default)로 기존 IP 라우팅을 사용하다가 같은 방향으로 가는 데이터가 일정량 이상 된다고 생각되면, 레이블을 할당해 스위칭 하는 방법이다. 따라서 데이터 양에 따라 네트워크를 유연하게 활용할 수 있고 가상 경로 합병에 의한 부가적 비용이 발생하지 않는다.

데이터 기반 레이블 스위칭 네트워크는 크게 동등 모델과 겹침 모델로 나눌 수 있다.

동등 모델(peer model)은 라우터끼리 스위치 계층과 IP 계층이 서로 대등하게 연결되어 있다. 동등 모델 라우터는 같은 목적지를 가지는 데이터가 일정량 이상 들어올 경우, 이 데이터 패킷에게 VCI(Virtual Connection Identifier)와 같은 레이블을 할당하고 이를 이용해 IP 계층까지 올라가지 않고 스위치 계층에서 데이터 패킷이 어디로 가야 할지 결정한다 (이를 컷드루(cut-through)라 한다.) 이러한 동등 모델은 각 호스트(host)를 구분하기 위해 기존의 IP 주소를 그대로 사용한다. 그리고 ATM과 같은 스위치 계층의 연결을 위해 기존의 IP 라우팅 프로토콜을 사용하기 때문에 독립적인 연결 신호 프로토콜(connection signaling protocol)과 주소 변환 프로토콜(address resolution protocol)을 필요로 하지 않다. 하지만 기존 IP가 가지고 있는 문제점을 그대로 유지하는 단점이 있다. IP 스위칭(IP switching) [9], CSR(Cell Switch Router) [10]과 같은 프로토콜(protocol)이 이와 같은 동등 모델에 속한다.

겹침 모델(overlay model)은 스위치 기반 네트워크가 IP 네트워크와 겹치면서 서로 독립적으로 존재하는 경우이다. 즉 IP 네트워크에서 ATM과 같은 스위치 네트워크로 데이터가 들어오면 에지 디바이스(edge device)에서

IP 주소를 스위치 네트워크 주소로 변환(address resolution)한 다음 이 데이터를 스위칭 기법으로 전송하는 것이다. 이 방식은 기존의 IP 네트워크와는 별도로 ATM과 같은 스위칭 기반의 네트워크를 새롭게 독립적으로 개발할 수 있다는 장점이 있지만 각각 서로 다른 기반의 주소를 사용하기 때문에 별도의 주소 분석 프로토콜(address resolution protocol)이 필요하다는 단점이 있다. CLIP(Classical IP over ATM) [11], LANE(LAN Emulation) [12], MPOA(Multi-Protocol Over ATM) [13]와 같은 프로토콜이 겹침 모델에 속한다. 특히 겹침 모델에서 CLIP나 LANE는 스위치 기반의 ATM 네트워크를 단순히 여러 개의 LIS(Logical IP Subnet)로 나누고 이를 기반으로 IP를 운용하기 때문에 라우터 병목현상(router bottleneck)이 발생할 수 있는데 MPOA는 이 문제점을 해결하기 위해서 제안되었다.

2.2 흐름 분류

플로우(flow)란 일정한 시간 내에 같은 출발지(source)와 목적지(destination)를 가지는 연속적인 패킷의 집합을 말한다. 플로우에 대해 우선적으로 자원을 할당하면 한정된 자원을 가지고 보다 효율적으로 많은 양의 데이터를 빠르게 전송할 수 있고, 각각의 데이터 플로우에 대해 서로 다른 자원 분배 정책을 사용함으로써 QoS를 위한 서비스의 차별화 등도 쉽게 구현할 수 있다. 그리고 각 플로우에 대해 서로 다른 과금을 책정함으로써 인터넷에서 과금과 서비스질을 향상시키는 등 다양한 인터넷 서비스를 개발할 수도 있다.

흐름 분류(flow classification)란 데이터 패킷을 보고 이것이 어느 플로우에 속하는 알아내는 과정을 말한다. 앞서 서술한 바와 같이 흐름 분류는 데이터 기반 레이블 스위칭 성능에 많은 영향을 미치기 때문에 매우 중요하다. 그러나 IP는 비연결형 패킷 전송을 사용하기 때문에 특정 패킷이 어떤 플로우에 속하는지 알아 내기가 쉽지 않고, 흐름 분류는 인터넷 트래픽의 특성과 네트워크 환경을 고려해서 복잡적으로 이루어져야 하기 때문에 앞으로도 많은 연구가 필요하다.

지금까지 연구된 흐름 분류자(flow classifier)로는 프로토콜 분류자(protocol classifier), 응용프로그램 분류자(application classifier), X/Y 분류자(X/Y classifier), X/Z 분류자(X/Z classifier) 등이 있다 [6, 7]. 이 중 X/Y 분류자는 Y 시간 안에 X개의 패킷이 오면 이를 플로우라고 분류하는 방법으로 간단하고 확장성이 뛰어나 MPOA에서 기본적인 흐름 분류 방법으로 제안되고 있다 [7, 13].

2.3 캐시 테이블 (Cache Table)

데이터 기반 레이블 스위칭에서 흐름 분류와 같이 중요

한 부분이 캐시 테이블 관리이다. 캐시 엔트리(cache entry)는 각 플로우마다 최소한 하나씩 필요로 하는데 캐시 테이블에는 스위칭에 필요한 정보들과 흐름 분류를 수행하기 위한 정보들이 저장된다. 스위칭을 위한 정보로는 IP 주소와 레이블 사이의 매핑 정보와 이 매핑이 언제까지 유효한지에 관한 유효 시간(holding time) 등이 있다. 흐름 분류를 위한 정보는 어떤 흐름 분류자를 사용하는지에 따라 달라지지만 대체적으로 플로우 카운터와 흐름 분류를 위한 시간 스탬프 등이 기록된다. 그림 2는 캐시 테이블 엔트리를 도식적으로 나타낸 것이다.

제목:
cacheEntry
작성한 사람:
Tgit.3.0-p13 by William Chia-Wei Cheng (william@cs.UCLA.edu)
이리 보기:
이 EPS 그림은 미리 보기 그림을 포함하지 않고
저장되었습니다.
설명:
nl.cdc 그림은 포스트스크린샷 포맷으로

그림 2 캐시 테이블 엔트리

캐시 테이블은 크면 클수록 보다 많은 정보를 저장할 수 있고 이를 이용해 더 많은 패킷을 효율적으로 스위칭시킬 수 있으며 많은 가상 연결을 유지해 가상 연결 설정 횟수를 줄여 보다 빠르게 데이터를 전송할 수 있다 [4].

하지만 캐시 테이블은 라우터 자원의 제약에 의해 무한히 커질 수 없다. 흐름 분류를 위해 목적지 주소만을 이용한다 해도 32비트 주소를 사용하는 IPv4에서는 이론적으로 약 4기가(giga) 개수만큼의 캐시 엔트리가 필요하다. 더욱이 128비트 주소를 사용하는 IPv6에서는 데이터를 캐시 엔트리에 모두 저장한다는 것은 거의 불가능하다.

그러므로 캐시 테이블이 모두 채워져 있을 때 새로운 데이터가 입력될 경우, 어떤 데이터를 대체할 것인지에 결정하는 캐시 대체 기법이 필요하다. 하지만 지금까지 데이터 기반 레이블 스위칭에 적용할 수 있는 적절한 캐시 대체 기법에 대한 연구는 거의 없었다.

따라서 다음 장에서는 레이블 스위칭 네트워크 상에서 사용자에게 의해 요구되어지는 인터넷 트래픽의 특성을 고려해 사용될 수 있는 여러 캐시 대체 기법들을 평가해 보고 효과적인 캐시 대체 기법을 제안한다.

3. 레이블 스위칭에서 캐시 대체 기법

본 장에서는 라우터가 가지는 스위칭 용량의 한계에 따라 캐시 대체 기법을 크게 두 가지로 분류하고 각 기법들에 따른 기본적인 알고리즘을 평가하고 새로운 알고리즘을 제안한다.

패킷이 들어오면 목적지 IP 주소에 해당하는 캐쉬 엔트리가 있는지 조사하고 있으면 정보를 갱신하고 없으면 새로운 엔트리를 생성한다. 그리고 흐름 분류에 의해 플로우라 판단되면 레이블을 할당해 스위칭만으로 전송하고 그렇지 않은 경우에는 기존 IP 라우팅으로 전송하게 된다.

앞서 기술한 바와 같이 캐쉬 엔트리 수는 한정되어 있기 때문에 새로운 목적지를 가진 패킷이 입력되었을 때 빈 캐쉬 엔트리가 없다면 캐쉬 엔트리 대체가 필요하다. 또한 스위칭 할 수 있는 레이블의 수(즉, 스위칭 용량)가 캐쉬 엔트리 수(즉, 캐쉬 용량)보다 작아서 캐쉬 엔트리가 스위칭 되기 원할 때 할당할 수 있는 레이블이 남아 있지 않다면 이 때는 스위칭 엔트리 대체가 필요하다.

그러므로 레이블 스위칭에서 캐쉬 대체 기법은 크게 두 가지로 나누어 생각해 볼 수 있다. 첫째는 캐쉬 엔트리가 스위칭 되기를 원할 때 언제든지 이를 레이블 스위칭 할 수 있을 만큼 라우터의 자원이 충분해서 스위칭 용량에 한계가 없는 경우로 이 때는 캐쉬 엔트리 대체만 있다. 하지만 일반적으로 라우터 자원은 모든 캐쉬 엔트리를 레이블 스위칭 할만큼 충분하지 않기 때문에 둘째로 레이블 스위칭 용량에 한계가 있는 경우에 대해서도 고려해야 하는데 이 때는 캐쉬 엔트리 대체뿐만 아니라 스위칭 엔트리 대체도 필요하다.

그림 3은 스위칭 용량의 한계에 따른 분류를 도식적으로 나타낸 것이다.

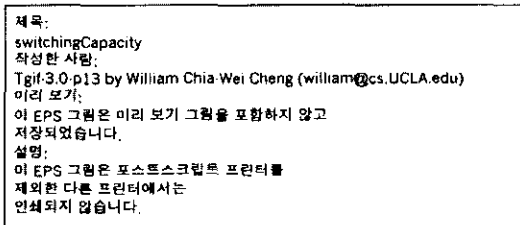


그림 3 스위칭 용량의 한계에 따른 분류

3.1 레이블 스위칭 용량에 한계가 없을 경우

모든 캐쉬 엔트리를 레이블 스위칭 할 수 있을 만큼 라우터 자원이 충분한 경우에 대해 생각해 보자. 이 경우 플로우가 라우터의 자원에 의해 제약을 받지 않으므로 캐쉬 엔트리는 흐름 분류에 의해 플로우가 되지만 하면 모두 레이블 스위칭 할 수 있다. 그러므로 레이블 스위칭 네트워크의 성능은 흐름 분류 방법과 캐쉬 대체 기법에 의해서만 영향을 받는다.

기본적인 알고리즘으로 FIFO(First In First Out) 기법, LFC(Least Flow Count) 기법, LRU(Least Recently

Used) 기법을 성능 평가해 보고 LRU 기법을 확장한 우선순위 LRU(priority LRU) 기법과 가중치 우선순위(weighted priority) LRU 기법을 제안한다.

3.1.1 기본적인 알고리즘

FIFO(First In First Out) 기법 FIFO 기법은 새로운 목적지를 가지는 데이터가 들어 왔을 때 빈 공간이 없으면 캐쉬 엔트리 중 가장 오랫동안 사용한 엔트리와 새로운 데이터를 대체하는 기법이다. FIFO 기법은 가장 오래된 데이터는 자원을 사용할 만큼 충분히 사용했으므로 새로운 데이터에게 자원을 양보해야 한다는 개념에서 나왔다. 따라서 모든 데이터가 자원을 공평(fairness)하게 나누어 쓸 수 있다. 하지만 FIFO 기법은 아직 많은 데이터를 보낼 수 있는 플로우임에도 불구하고 오래 되었다는 이유만으로 캐쉬 엔트리를 삭제할 수 있기 때문에 데이터의 특성을 고려하지 못하고 비효율적인 면이 있다. 특히 지속적으로 데이터를 보내는 플로우가 있을 경우, 추가적으로 캐쉬에서 삭제되었다가 다시 캐쉬로 삽입되는 플로우 인앤아웃(flow in-and-out) 현상이 발생한다. 이런 플로우 인앤아웃 현상은 레이블 스위칭을 위한 설정과 삭제를 반복해서 수행하게 하므로 레이블 스위칭 네트워크의 성능을 크게 저하시킨다.

LFC(Least Flow Count) 기법 LFC 기법은 캐쉬에 빈 공간이 없을 경우 캐쉬 엔트리 중 가장 작은 수의 패킷을 전달한, 즉 플로우 카운트가 가장 작은 엔트리와 새로운 데이터를 대체하는 기법이다. 이 기법은 MPOA에서 기본적으로 사용하기를 권고하는 기법으로 과거의 많은 패킷을 보냈으면 앞으로도 많은 패킷을 보낼 것이라는 예측에서 나왔다. LFC 기법은 캐쉬 엔트리를 대체할 때 과거의 자료를 이용하기 때문에 데이터의 특성을 고려하는 장점이 있고 플로우 인앤아웃 현상이 없기 때문에 레이블 스위칭을 위한 비용을 최소화하는 장점도 있다. 하지만 LFC 기법을 사용하면 과거에 많은 패킷을 보낸 엔트리가 더 이상 사용되지 않더라도 다른 모든 캐쉬 엔트리의 플로우 카운트 수보다 작아지지 않는 한 레이블 유효시간(Holding time)이 끝나기 전에 캐쉬 테이블에서 삭제되지 않는다. 특히 캐쉬 테이블이 가득 차 캐쉬 엔트리 대체가 자주 발생하면 늘 최근에 삽입된 엔트리가 먼저 대체되기 때문에 새로운 엔트리를 플로우로 잘 만들어 내지 못하는 단점을 가진다. 그리고 LFC 기법을 구현하기 위해서는 힙(heap) 구조를 유지할 필요가 있는데 이것은 FIFO 기법이나 LRU 기법이 연결 리스트(linked list) 구조로 구현되는 것에 비해 복잡하고 시간 비용도 많이 소모된다.

LRU(Least Recently Used) 기법 LRU 기법은 캐쉬에 빈 공간이 없으면 캐쉬 엔트리 중에서 가장 오래 동안 사

용하지 않은 엔트리와 새로운 데이터를 대체하는 기법이다. LRU 기법은 가상 메모리에서 좋은 성능을 내는 LRU 대체 기법을 응용한 것으로 데이터의 지역성(locality)을 고려해 가장 오래 전에 사용한 데이터가 앞으로도 가장 사용되지 않으리라는 발상에서 나온 기법이다. LRU 기법을 사용하면 오래 동안 사용하지 않은 데이터보다 최근 데이터에게 우선권을 주는 공평성을 가지면서도 데이터의 특성을 참조해, 오래 되었더라도 최근까지 자주 사용되는 데이터는 계속 유지하는 장점을 가지고 있다. 또 연결 리스트 구조를 사용하기 때문에 LFC 기법의 힙 구조보다 쉽게 구현할 수는 장점이 있다.

3.1.2 우선순위 LRU(priority LRU) 기법

앞서 본 바와 같이 LRU 기법은 FIFO 기법에 비해 과거 자료를 이용한 데이터 특성 고려하고 플로우 인엔아웃 현상이 없으며 LFC 기법에 비해 공평성, 구현 측면에서 우수하다. 그러나 LRU 기법은 캐쉬 테이블에서 삭제될 대상을 고를 때 삭제 대상이 이미 레이블 자원과 설정 시간과 같은 레이블 스위칭 비용을 소모해 스위칭 되고 있음을 고려하지 않는다. 즉 같은 조건이라면 이미 레이블 스위칭 되고 있는 데이터에게 새로이 스위칭 되려 하는 데이터 보다 우선권을 줄 필요가 있는데 LRU 기법은 이를 고려하지 못하는 단점을 가지고 있다.

따라서 LRU기법이 스위칭 되고 있는 엔트리의 비용가치를 고려하지 못하는 단점을 극복하고자 우선순위 LRU 기법을 제안했다. 우선순위 LRU 기법은 삭제 대상을 찾을 때 전체 캐쉬 엔트리에서 찾지 않고 레이블 스위칭 되지 않는 엔트리 중 가장 오랫동안 사용하지 않은 엔트리를 찾는다. 그리고 모든 캐쉬 엔트리가 레이블 스위칭 되고 있다면 그 때셔야 스위칭 되고 있는 엔트리 중에서 가장 오랫동안 사용하지 않은 엔트리를 찾는다. 그림 4는 우선 순위 LRU 기법을 도식적으로 나타낸 것이다.

우선순위 LRU 기법은 캐쉬 테이블에서 데이터를 삭제할 때 레이블 스위칭 되지 않는 것을 먼저 고려하기 때문에 플로우 인엔아웃 현상이 없고 레이블 스위칭 설정 비용

제목:
pLRU
작성한 사람:
Tgif.3.0.p13 by William Chia-Wei Cheng (william@cs.UCLA.edu)
미리 보기:
이 EPS 그림은 미리 보기 그림을 포함하지 않고 저장되었습니다.
설명:
이 EPS 그림은 포스트스크립트 프린터를 제외한 다른 프린터에서는 인쇄되지 않습니다.

그림 4 우선순위 LRU 기법

이 최소화된다. 하지만 캐쉬 테이블에 있는 대부분의 데이터가 플로우가 되어 레이블 스위칭 되고 있다면 최근에 삽입되어 아직 스위칭 되고 있지 않은 데이터가 항상 먼저 삭제됨으로 새로운 플로우를 제대로 만들어 내지 못하게 된다. 그리고 이로 인해 시간이 지남에 따라 라우터의 성능이 저하되며 공평성 결여(unfairness) 현상이 발생한다.

3.1.3 가중치 우선순위 LRU(weighted priority LRU) 기법

우선순위 LRU 기법은 대부분의 데이터가 레이블 스위칭 되고 있으면 새로운 플로우를 제대로 만들어 내지 못하는데 이는 이미 스위칭 되고 있는 데이터의 우선순위가 지나치게 높기 때문이다. 따라서 임계 값(threshold)을 두어 최근 데이터에 대한 우선순위를 적절히 높여 주는 가중치 우선순위 LRU 기법을 제안한다.

가중치 우선순위 LRU 기법은 삭제 대상을 찾을 때 레이블 스위칭 되지 않는 엔트리 중에서 LRU 엔트리를 찾고 그 엔트리의 LRU 시간이 임계 값보다 오래 되었다면 삭제하고 그렇지 않다면 스위칭 되고 있는 엔트리들에서 LRU 엔트리를 찾아 둘 중 오래된 것을 삭제하고 새로운 데이터를 캐쉬 테이블에 삽입한다. 임계 값은 흐름 분류자의 종류와 성격에 따라 달라질 수 있다. 그림 5는 가중치 우선순위 LRU 기법을 도식적으로 나타낸 것이다.

제목:
wpLRUScheme
작성한 사람:
Tgif.3.0.p13 by William Chia-Wei Cheng (william@cs.UCLA.edu)
미리 보기:
이 EPS 그림은 미리 보기 그림을 포함하지 않고 저장되었습니다.
설명:
이 EPS 그림은 포스트스크립트 프린터를 제외한 다른 프린터에서는 인쇄되지 않습니다.

그림 5 가중치 우선순위 LRU 기법

가중치 우선순위 LRU 기법은 기존의 우선순위 LRU 기법이 가지는 장점을 유지하면서 스위칭 데이터가 가진 더 지나친 우선권을 줄이고 최근 데이터에 대한 우선순위를 적절히 향상시켰다. 따라서 캐쉬 테이블에 있는 대부분의 데이터가 레이블 스위칭 되더라도 스위칭 되는 데이터를 적절히 삭제해 시간에 관계없이 꾸준히 최근 데이터가 새로운 플로우가 될 수 있다.

3.2 레이블 스위칭 용량에 한계가 있을 경우

일반적으로 라우터의 자원은 모든 캐쉬 엔트리를 레이블 스위칭 할 수 있을 만큼 충분하지 않으므로 캐쉬 엔트리가 플로우가 되더라도 레이블 스위칭 자원의 한계에 의

해 스위칭 하지 못할 수 있다. 그러므로 레이블 스위칭 용량에 한계가 있다면, 캐쉬 대체 기법을 첫째 단계로 새로운 데이터가 들어왔을 때 캐쉬 엔트리 중 하나를 대체하는 캐쉬 엔트리 대체와 둘째 단계로 새로운 플로우가 생겨 이를 스위칭 해야 할 때 스위칭 되고 있는 엔트리 중 하나를 대체하는 스위칭 엔트리 대체로 나누어 적용하는 것이 합리적이다. 그림 6은 이러한 2단계 캐쉬 대체를 도식적으로 나타낸 것이다.

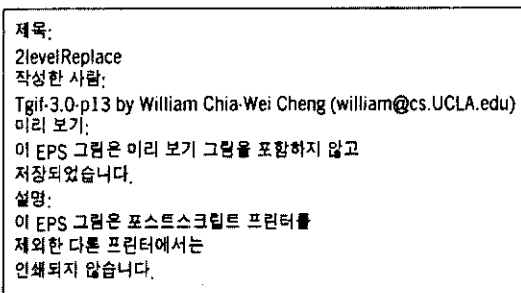


그림 6 레이블 스위칭 용량에 한계가 있을 경우 2단계 캐쉬 대체

본 논문에서는 기본적인 2단계 캐쉬 대체 기법으로 LFC-LFC 기법, LFC-LRU 기법, LRU-LFC 기법, LRU-LRU 기법을 성능 평가하고 LRU-LRU 기법을 확장한 LRU-가중치 LRU(LRU-weighted LRU) 기법을 제안한다.

3.2.1 기본적인 알고리즘

LFC-LFC 기법 LFC-LFC 기법은 첫째, 둘째 단계 캐쉬 대체 모두 LFC 기법을 사용한다.

즉 새로운 데이터가 들어와서 캐쉬 엔트리를 대체해야 할 때 스위칭 되고 있지 않는 엔트리 중 플로우 카운터가 가장 작은 것을 삭제하고, 엔트리 중 하나가 플로우가 되어 스위칭 엔트리를 대체해야 할 때도 플로우 카운터가 가장 작은 엔트리를 삭제하는 기법이다. 하지만 LFC 기법은 특성 상 과거 자료에 너무 많은 우선권을 주기 때문에 첫째 단계에서 새로운 플로우를 제대로 생성해 내지 못하고 둘째 단계에서도 최근에 스위칭 되기 시작한 엔트리가 먼저 삭제되어 플로우 인엔아웃 현상이 발생해 불필요한 스위칭 설정 비율(switching setup rate)이 높아지는 문제점이 있다.

LFC-LRU 기법 LFC-LRU 기법은 첫째 캐쉬 대체 기법으로 LFC 기법을 사용하고, 둘째 캐쉬 대체 기법으로 LRU 기법을 사용한다. 즉 캐쉬 엔트리를 대체해야 할 때에는 플로우 카운터가 가장 작은 엔트리를 삭제하고, 스

칭 엔트리를 대체해야 할 때는 가장 오랫동안 사용하지 않은 엔트리를 삭제하는 기법이다. LFC-LRU 기법 역시 첫째 단계 캐쉬 대체로 LFC 기법을 사용하기 때문에 새로운 플로우를 제대로 생성해 내지 못한다. 하지만 둘째 단계 캐쉬 대체는 LRU 기법을 사용하기 때문에 플로우 인엔아웃 현상이 줄어들 불필요한 스위칭 설정 비율이 없다.

LRU-LFC 기법 LRU-LFC 기법은 첫째 캐쉬 대체 기법으로 LRU 기법을, 둘째 캐쉬 대체 기법으로 LFC 기법을 사용한다. 즉 캐쉬 엔트리를 대체해야 할 때는 가장 오랫동안 사용되지 않은 엔트리를 삭제하고, 스위칭 엔트리를 대체해야 할 때는 가장 작은 플로우 카운터를 가진 엔트리를 삭제하는 기법이다. LRU-LFC 기법은 첫째 단계 캐쉬 대체로 LRU 기법을 사용하기 때문에 새로운 플로우를 LFC 기법 보다 많이 생성해 낸다. 하지만 둘째 단계 캐쉬 대체로 LFC 기법을 사용하기 때문에 플로우 인엔아웃 현상이 발생해서 스위칭 설정 비율을 불필요하게 높여 스위칭 비용을 증가시킨다.

LRU-LRU 기법 LRU-LRU 기법은 첫째 둘째 캐쉬 대체 기법 모두 LRU 기법을 사용한다.

즉 캐쉬 엔트리를 대체해야 할 때 가장 오랫동안 사용하지 않은 엔트리를 삭제하고, 스위칭 엔트리를 대체해야 할 때도 가장 오랫동안 사용하지 않은 엔트리를 삭제하는 기법이다. 이 기법은 첫째 캐쉬 대체 기법으로 LRU 기법을 사용하므로 LFC 기법에 비해 많은 플로우를 만들어 낼 수 있고 둘째 캐쉬 대체 기법 역시 LRU 기법을 사용하므로 플로우 인엔아웃 현상을 줄여 불필요한 스위칭 설정 비율을 줄인다. 하지만 새로운 플로우가 만들어지면 무조건 스위칭 되던 엔트리와 대체하므로 기존의 스위칭 엔트리가 만들어지기 위해 사용했던 시간 비용을 고려하지 못하는 단점이 있다.

3.2.2 LRU-가중치 LRU 기법

LRU-LRU 기법은 새로운 플로우에 의해 기존의 스위칭 엔트리를 삭제할 때, 스위칭 엔트리를 만들기 위해 사용된 시간 비용을 고려하지 못하는 단점을 가지고 있는데 이를 보완하고자 하는 것이 LRU-가중치 LRU 기법이다. 즉 새로운 플로우와 기존의 스위칭 엔트리를 대체하고자 할 때, 스위칭 엔트리가 최근에 만들어 졌다면 그 엔트리를 만들때 사용된 시간 비용을 고려해 기존의 스위칭 엔트리를 새로운 플로우와 대체하지 않고 계속 유지하자는 것이다.

LRU-가중치 LRU 기법은 우선 캐쉬 엔트리 대체로 LRU 기법을 사용한다. 그리고 스위칭 엔트리 대체에서 LRU 기법을 이용해 스위칭 되고 있는 엔트리 중 가장 오랫동안 사용하지 않은 엔트리를 찾는다. 이 때 찾은 스

칭 엔트리의 마지막 사용 시간이 새로이 플로우가 된 엔트리
가 플로우가 되기 전 마지막으로 사용한 시간보다 최근
이라면 스위칭 엔트리를 대체하지 않고 기존의 스위칭 엔
트리를 계속 유지하는 것이다. 그림 7은 LRU-가중치
LRU 기법을 도식적으로 나타낸 것이다.

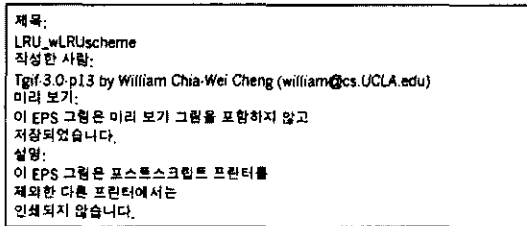


그림 7 LRU-가중치 LRU 기법

LRU-가중치 LRU 기법은 플로우가 된 데이터를 바로
스위칭 하지 않고 지연시킴으로 LRU-LRU 기법보다 스
위칭 시키는 데이터 양이 조금 줄어들 수 있다. 하지만
LRU-LRU 기법의 장점들을 대부분 수용면서 스위칭을
위해 사용되는 시간 비용을 크게 줄일 수 있다. 특히 캐쉬
크기가 작아 많은 플로우가 인엔아웃 되는 스위칭 네트워
크에서 LRU-가중치 LRU 기법은 좋은 효과를 나타낼 수
있다.

4. 성능 평가

본 장에서는 앞서 소개한 여러 캐쉬 대체 기법들을 실
제 인터넷 트래픽을 이용해 실험해 보고 그 성능을 평가한
다.

성능 평가를 위해 가상적인 레이블 스위칭 라우터를 고
안하고 이 라우터에 서로 다른 캐쉬 대체 기법을 적용해
그 성능을 측정했다.

흐름 분류는 어떤 패킷을 플로우로 보고 레이블 스위칭
할 것인지 결정하기 위해 필요하다. 실험에서는 X/Y 흐름
분류자를 사용하고 같은 목적지를 가지는 패킷이 1초안에
10개 이상 입력되면 이를 플로우로 보고 레이블 스위칭 시
킨다. 이것은 ATM 포럼에서 MPOA를 위해 제안한 기본
흐름 분류자와 다프트 값이다.

4.1 실험에 사용된 인터넷 트래픽 특성

실험에 사용한 인터넷 트래픽은 NLANR(Network
Laboratory for Applied Network Research) [14]에서
제공한 것으로 내용은 다음과 같다.

- 측정 장소 : FIX-WEST (미국 서부)
- 측정 시간 : 1997년 11월 20일 19시 45분부터 1155
초 동안 측정

- 패킷 수 : 13,676,730 패킷 (평균 초당 11.8 K 패킷)
- 데이터량 : 약 5 기가 바이트 (평균 34.9 Mbps)
- 사용된 IP 주소 수 : 114,433 개 (평균 초당 99.0 개)
- 재사용된 IP 주소 수 : 13,561,432 개 (평균 초당
11,741.5 개)

4.2 측정 기준

레이블 스위칭 네트워크에서 중요한 성능 평가 측정 기
준에는 스위칭 패킷 수, 스위칭 설정 비율(setup rate) 등
이 있다.

스위칭 패킷 수 단위 시간 당 레이블 스위칭 되는 패킷
수를 나타낸다. 레이블 스위칭 네트워크는 스위칭 되는 패
킷 수가 많을 수록 좋은 성능을 나타냄으로 캐쉬 대체 기
법에 따라 얼마나 많은 수의 패킷을 스위칭 하느냐가 성능
측정에서 가장 중요한 요소가 된다.

스위칭 설정 비율 단위 시간 당 스위칭 하기 위해 설정
과정을 필요로 하는 캐쉬 엔트리의 개수를 나타낸다. 캐쉬
대체 기법에 따라 플로우를 제대로 만들어 내지 못할 경우
스위칭 설정 비율이 지나치게 낮아진다. 이렇게 되면 스위
칭 되는 데이터 양이 줄어 레이블 스위칭 네트워크 성능이
나빠진다. 한편 플로우 인엔아웃 현상에 의해 설정 비율이
지나치게 높아지면 스위칭을 위한 시간 비용이 많이 소모
됨으로 좋지 않다. 따라서 스위칭 설정 비율은 적절히 유
지하는 것이 좋다.

4.3 레이블 스위칭 용량에 한계가 없을 경우

레이블 스위칭 용량에 한계가 없을 때 기본적인 알고리
즘으로 FIFO 기법, LFC 기법, LRU 기법에 대해 성능 평
가해 보고 우선순위 LRU 기법과 가중치 LRU 기법에 대
해 성능 평가 한다.

그림 8, 9는 캐쉬 크기에 따른 각 캐쉬 대체 기법들의
성능을 보여 주고 있다. 전체적으로 스위칭 패킷 수는 캐
쉬가 커질 수록 늘어나고 스위칭 설정 비율은 캐쉬가 커질
수록 줄어 든다.

FIFO 기법은 평균 초당 스위칭 되는 패킷 수가 다른 기
법 중 가장 작다. 그리고 스위칭 설정 비율이 가장 높다.
이는 FIFO 기법의 플로우 인엔아웃 현상이 가장 심하기
때문이다.

LFC 기법은 캐쉬가 아주 작을 경우 스위칭 패킷 수가
크지만 캐쉬가 커질 수록 그 수가 LRU 기법보다 작아 진
다. 그리고 스위칭 설정 비율도 아주 낮다. 이는 LFC 기법
이 플로우가 되어 스위칭하고 있는 엔트리를 내보내지 않
고 유지하려는 특징이 강하기 때문에 캐쉬가 아주 작은 경
우 다른 기법에 비해 많은 스위칭 패킷 수를 가질 수 있지
만 캐쉬가 커질 수록 새로운 플로우를 만들어 내지 못하기
때문이다.

제목:
l1avgfPkt.ps
작성한 사람:
gnuplot 3.7 patchlevel 1
미리 보기:
이 EPS 그림은 미리 보기 그림을 포함하지 않고
저장되었습니다.
설명:
이 EPS 그림은 포스트스크립트 프린터를
제외한 다른 프린터에서는
인쇄되지 않습니다.

그림 8 캐쉬 크기에 따른 평균 초당 스위칭 패킷 수

제목:
l1avgfPkt.ps
작성한 사람:
gnuplot 3.7 patchlevel 1
미리 보기:
이 EPS 그림은 미리 보기 그림을 포함하지 않고
저장되었습니다.
설명:
이 EPS 그림은 포스트스크립트 프린터를
제외한 다른 프린터에서는
인쇄되지 않습니다.

그림 9 캐쉬 크기에 따른 평균 초당 스위칭 설정 비율

LRU 기법은 캐쉬가 아주 작을 때 LFC 기법보다 작은 스위칭 패킷 수를 가지다 캐쉬가 커질 수록 LFC 기법보다 많은 스위칭 패킷 수를 가진다. 그리고 LRU 기법은 스위칭 설정 비율면에서 FIFO 기법과 LFC 기법 중간 정도 값을 유지한다. 캐쉬 크기가 아주 작을 때 LFC 기법보다 LRU 기법이 더 작은 스위칭 패킷 수를 나타내는 것은 캐쉬 크기가 아주 작을 경우 LRU 기법에서도 플로우 인엔아웃 현상이 발생하기 때문이다. 하지만 캐쉬 크기가 적정 수준 이상으로 커지면 FIFO 기법, LFC 기법 보다 좋은 성능을 나타낸다. 따라서 기본 알고리즘 중에서는 LRU 기법이 가장 좋은 성능을 보였다.

우선순위 LRU 기법은 스위칭 패킷 수에서 LRU 기법 보다 나은 성능을 보인다. 하지만 스위칭 설정 비율이 LFC 기법처럼 아주 낮음을 알 수 있다. 이처럼 우선순위 LRU 기법이 스위칭 설정 비율이 아주 낮은 이유는 과거의 데이터에 너무 많은 우선권을 부여해 시간이 지날 수록 새로운 플로우를 잘 만들어 내지 못하기 때문이다.

가중치 우선순위 LRU 기법은 우선순위 LRU 기법 보다 높은 스위칭 패킷 수를 보여 주고 적절한 스위칭 설정 비율을 보여 준다.

4.4 레이블 스위칭 용량에 한계가 있을 경우

레이블 스위칭 용량에 한계가 있을 경우, 기본적인 2단

계 캐쉬 대체 기법으로 LFC-LFC 기법, LFC-LRU 기법, LRU-LFC 기법, LRU-LRU 기법에 대해 성능 평가하고 LRU-가중치 LRU 기법에 대해 성능 평가한다.

레이블 스위칭 용량은 전체 캐쉬 엔트리의 10%로 가정한다.

그림 10, 11은 캐쉬 크기에 따른 각 캐쉬 대체 기법들의 성능을 보여 주고 있다. 전체적으로 스위칭 패킷 수는 캐쉬가 커질 수록 늘어나고 스위칭 설정 비율은 캐쉬가 커질 수록 줄어 든다.

제목:
l1avgfSet.ps
작성한 사람:
gnuplot 3.7 patchlevel 1
미리 보기:
이 EPS 그림은 미리 보기 그림을 포함하지 않고
저장되었습니다.
설명:
이 EPS 그림은 포스트스크립트 프린터를
제외한 다른 프린터에서는
인쇄되지 않습니다.

그림 10 캐쉬 크기에 따른 평균 초당 스위칭 패킷 수

제목:
l2avgfPkt.ps
작성한 사람:
gnuplot 3.7 patchlevel 1
미리 보기:
이 EPS 그림은 미리 보기 그림을 포함하지 않고
저장되었습니다.
설명:
이 EPS 그림은 포스트스크립트 프린터를
제외한 다른 프린터에서는
인쇄되지 않습니다.

그림 11 캐쉬 크기에 따른 평균 초당 스위칭 설정 비율

전체적으로 분석해 보면 캐쉬 엔트리 대체로 LFC를 쓰면 LFC 특성으로 인해 새로운 플로우를 제대로 못 만들고, 스위칭 엔트리 대체로 LFC를 사용하면 최근 스위칭 플로우가 먼저 삭제되어 플로우 인엔아웃 현상이 발생할 수 있다. 따라서 기본적인 알고리즘에서는 LRU-LRU 기법이 나은 성능을 보였고, LRU-가중치 LRU 기법은 플로우를 스위칭 하기 위한 설정 시간 비용까지 고려해 가장 좋은 성능을 보이고 있다.

5. 결론

본 논문에서는 데이터 기반 레이블 스위칭 네트워크에서 사용될 수 있는 기본적인 여러 캐쉬 대체 기법에 대해

평가해 보았고 새로운 캐쉬 대체 기법으로 우선순위 LRU 기법, 가중치 우선순위 LRU 기법과 LRU-가중치 LRU 기법을 제안하고 그 성능을 평가하였다.

캐쉬 대체 기법은 크게 라우터에서 스위칭 용량에 한계가 있는 경우와 없는 경우로 나누어 고려해 볼 수 있었는데, 스위칭 용량에 한계가 없을 경우엔 기본 알고리즘으로 FIFO 기법, LFC 기법, LRU 기법을 성능 평가했고 이를 확장한 기법으로 우선순위 LRU 기법과 가중치 우선순위 LRU 기법을 제안했다. 기본적인 알고리즘에서 FIFO 기법은 플로우 인앤아웃 현상이 나타나는 문제점이 있었고, LFC 기법은 과거의 데이터에 지나친 우선권을 주어 새로운 플로우를 제대로 만들어 내지 못하는 문제점이 있었다. 그리고 LRU 기법이 가장 나은 성능을 보였는데 이를 개선해서 스위칭 설정 비용을 고려하여 현재 스위칭 되고 있는 데이터에게 우선권을 주는 우선순위 LRU 기법을 제안했다. 하지만 우선순위 LRU 기법은 스위칭 된 데이터에 지나친 우선권을 주어 시간이 지남에 따라 새로운 플로우를 제대로 만들어 내지 못했기 때문에 시간이 지날수록 성능이 떨어졌다. 따라서 이를 다시 개선한 가중치 우선순위 LRU 기법을 제안했다. 가중치 우선순위 LRU 기법은 스위칭 되고 있는 데이터의 스위칭 설정 비용을 고려하면서도 지나치게 오래된 데이터를 적절히 삭제함으로써 스위칭 라우터의 성능을 향상시켜 가장 좋은 성능을 나타냈다.

스위칭 용량에 한계가 있을 경우에는 두 단계 캐쉬 대체 기법을 적용하는 것이 합리적인데 기본 알고리즘으로 LFC-LFC 기법, LFC-LRU 기법, LRU-LFC 기법, LRU-LRU 기법을 성능 평가했고 LRU-LRU 기법을 개선한 LRU-가중치 LRU 기법을 제안했다. 기본 알고리즘에서는 LRU-LRU 기법이 가장 나은 성능을 보였다. 왜냐하면 캐쉬 엔트리 대체로 LFC를 쓰면 LFC 특성으로 인해 새로운 플로우를 제대로 만들어 내지 못하고, 스위칭 엔트리 대체로 LFC를 사용하면 최근 스위칭 플로우가 먼저 삭제되어 플로우 인앤아웃 현상이 발생하기 때문이다. 그리고 LRU-LRU 기법을 개선해 LRU-LRU 기법과 거의 같은 성능을 내면서도 스위칭 설정 비용을 줄여 스위칭 될 때 소모되는 시간 비용을 최소화하는 LRU-가중치 LRU 기법을 제안했다.

이상의 연구 결과를 통해, 레이블 스위칭 네트워크에서 어떠한 캐쉬 대체 기법을 적용하는가에 따라 전체 레이블 스위칭 네트워크의 성능이 크게 좌우됨을 알 수 있었다.

특히 레이블 스위칭 네트워크에 전송되는 데이터량이 백본망(back bone network)처럼 매우 크다면 캐쉬 대체 기법의 중요성은 더욱 커짐을 알 수 있다.

본 연구에서는 주로 LRU 기법을 기본으로 이를 확장한

여러 캐쉬 대체 기법을 제안하고 성능 평가 했다. 하지만 LRU 기법 외에도 FIFO 기법, LFC 기법 모두 나름대로의 특징과 장점을 가지고 있다. 특히 LFC 기법은 스위칭 엔트리에 대해 특별한 우선권을 주지 않더라도 스위칭 되고 있는 엔트리를 쉽게 삭제하지 않는 특징을 가지고 있어서 스위칭 해제 비율을 낮게 유지하는 장점을 가지고 있다. 따라서 앞으로는 LFC 기법에 적절한 가중치를 적용한 가중치 LFC(weighted LFC) 기법에 대해서 연구해 볼 계획 이다.

참 고 문 헌

- [1] <http://www.nw.com>
- [2] Nexabit Inc., Architecture White Paper, White Paper in <http://www.nexabit.com>, Mar. 1999.
- [3] Pluris Inc., Competitive Study, White Paper in <http://www.pluris.com>, Apr. 1999.
- [4] Widjaja, H. Wang, S. Wright and A. Chatterjee, Scalability Evolution of Multi-Protocol Over ATM (MPOA), in IEEE INFOCOM'99, Mar. 1999.
- [5] P. Boustead, J. Chicharo, and G. Anido, Scalability and Routing Performance of Label Switching Networks, in IEEE GROBECOM'98, Nov. 1998.
- [6] H. Che and S. qi Li, MPOA Flow Classification Design and Analysis, in IEEE INFOCOM'99, Mar. 1999.
- [7] S. Lin and N. McKeown, A Simulation Study of IP Switching, in ACM SIGCOMM'97, Sept. 1997.
- [8] S. Deering and R. Hinden, Internet Protocol, Version 6 (IPv6) Specification, RFC 2460, Dec. 1998.
- [9] P. Newman, G. Minshall, and T. L. Lyon, IP Switching: ATM Under IP, IEEE/ACM Transactions on Networking, vol. 6, Apr. 1998.
- [10] Toshiba Corp., Cell Switch Router White Paper Version 1.0, White Paper, Nov. 1996.
- [11] M. Laubach, Classical IP and ARP over ATM, RFC 1577, Jan. 1994.
- [12] ATM Forum, LAN Emulation Over ATM Version 1.0, af-lane-0021.000, Jan. 1995.
- [13] ATM Forum, Multi-Protocol Over ATM Version 1.0, af-mpoa-0087.000, July 1997.
- [14] <http://www.nlanr.net>



김 남 기

1997년 2월 서강대학교 컴퓨터학과 학사.
2000년 2월 한국과학기술원 전자전산학과 석사. 2000년 3월 ~ 현재 한국과학기술원 전자전산학과 박사 과정 재학중. 관심분야는 컴퓨터 네트워크, 인터넷, 이동통신, adhoc, ATM.



윤 현 수

1979년 서울대학교 전자공학과 학사.
1981년 한국과학기술원 전산학과 석사.
1981년 ~ 1984년 삼성전자 연구원.
1988년 오하이오 주립대학 전산학 박사.
1988년 ~ 1989년 AT&T Bell Labs. 연구원. 1989년 ~ 현재 한국과학기술원 전산학과 교수. 관심분야는 상호연결 네트워크, ATM switch, 병렬 컴퓨터 구조임