

# 스마트 분배기를 이용한 효율적인 병렬 IP 주소 검색 구조

## Efficient Parallel IP Address Lookup Architecture with Smart Distributor

김정환, 김진수  
건국대학교 컴퓨터공학과

Junghwan Kim(jhkim@kku.ac.kr), Jinsoo Kim(jinsoo@kku.ac.kr)

### 요약

인터넷에서 초고속 서비스의 제공을 위해서, 라우터는 신속하게 IP 주소 검색을 수행해야 한다. 본 논문에서는 4 단계의 파이프라인으로 구성된 하이브리드 방식의 병렬 IP 주소 검색 구조를 제안한다. 단계 2의 다중 SRAM과 단계 3의 분할 TCAM을 사용하여 저렴한 비용으로 병렬화를 이루고, 파이프라이닝을 통해 처리율을 향상시켰다. 단계 1에서 스마트 분배기는 선행된 IP 주소와 동일한 주소를 다음 단계로 진입시키지 않고 선행의 검색 결과를 그대로 활용하게 한다. 이에 따라, 캐싱 효과에 의해 검색 처리율을 더욱 향상시키고, 단계 3인 TCAM bank로의 접근 충돌도 완화시킬 수 있다. 마지막 단계의 재정렬 버퍼는 처리된 IP 주소 순서를 입력 순서에 맞도록 재조정하는 기능을 수행한다. 또한, 실제 사용되는 라우팅 테이블과 Zipf 법칙에 따라 생성된 트래픽 분포를 사용하여, 병렬 파이프라인 IP 검색 구조의 성능을 기존의 하이브리드 구조와 비교하여 평가한다.

■ 중심어 : | 고성능 인터넷 라우터 | 병렬 IP 주소 검색 | 성능 개선 | 스마트 분배기 |

### Abstract

Routers should perform fast IP address lookup for Internet to provide high-speed service. In this paper, we present a hybrid parallel IP address lookup structure composed of four-stage pipeline. It achieves parallelism at low cost by using multiple SRAMs in stage 2 and partitioned TCAMs in stage 3, and improves the performance through pipelining. The smart distributor in stage 1 does not transfer any IP address identical to previous one toward the next stage, but only uses the result of the previous lookup. So it improves throughput of lookup by caching effects, and decreases the access conflict to TCAM bank in stage 3 as well. In the last stage, the reorder buffer rearranges the completed IP addresses according to the input order. We evaluate the performance of our parallel pipelined IP lookup structure comparing with previous hybrid structure, using the real routing table and traffic distributions generated by Zipf's law.

■ keyword : | High-performance Router | Parallel IP Address Lookup | Performance Improvement | Smart Distributor |

## I. 서론

인터넷은 다수의 라우터와 이들을 연결하여 패킷을

전송하는 전송매체로 구성된다. 다양한 멀티미디어 콘텐츠에 대한 수요 증가와 사용자 수의 증가에 따라, 인터넷 트래픽은 지속적으로 늘어가고 있다. 광섬유와 같

이 높은 대역폭을 제공하는 전송 매체의 발달로 전송매체 자체는 고속의 전송이 가능하게 되었지만, 패킷을 포워딩하는 라우터의 성능 향상 속도는 전송매체를 따라가지 못하고 있다. 따라서 인터넷에서 고성능 라우터의 개발은 초고속 서비스를 제공하기 위한 필수적인 요소이다[1].

라우터의 핵심 기능은 패킷을 최종 목적지 방향으로 포워딩하는 것이다. 라우터는 입력 포트에 들어온 패킷을 어느 출력 포트에 내보낼 지를 결정하여야 한다. 이를 위해 입력 패킷의 목적지 IP 주소를 라우팅 테이블에서 검색해서 대응되는 출력 포트를 구하는데, 이를 라우터의 IP 주소 검색 기능이라고 한다. CIDR(classless interdomain routing)의 도입으로 가변 길이의 프리픽스를 허용하면서, IP 주소의 검색은 목적지 주소에 매치(match)되는 여러 프리픽스 중 최장 매칭 프리픽스(LMP, Longest Matching Prefix)를 가장 적합한 프리픽스로서 선택한다. 이에 따라, IP 주소 검색은 더 복잡하고 처리시간이 오래 걸리게 되었고, IP 주소 검색의 효율성이 라우터 성능을 좌우하는 중요한 요소로 더욱 부각되었다.

고성능 라우터의 개발을 위해 IP 주소 검색에 대한 많은 기법들이 제안되어 왔고[2-4], 대표적인 방식으로 trie를 기반으로 하는 기법과 TCAM(Ternary Content Addressable Memory)을 기반으로 하는 기법이 있다. Trie는 트리와 유사한 형태의 구조이며 일반적으로 SRAM을 사용하며 구현된다. Trie를 이용한 IP 주소 검색은 한 번의 검색을 위해 다수의 메모리 접근을 해야 한다. 따라서 trie 기반의 IP 주소 검색의 주된 연구 방향은 메모리 접근 횟수를 줄이는 것이었다[5].

TCAM은 프리픽스의 각 비트가 0과 1뿐만 아니라 \*를 표현할 수 있는 연관메모리로서 자체적으로 병렬적 특성을 지니고 있다. 따라서 TCAM 기반의 주소 검색은 한 번의 메모리 접근으로 수행이 가능하여, LMP를 쉽게 찾을 수 있다. TCAM은 SRAM에 비해 접근 시간이 더 길고, 전력소비도 더 크다. 따라서 TCAM 기반 주소검색은 라우팅 테이블의 압축이나 분할 등을 통해 전력 소비를 줄이는 방향으로 진행되었다[6].

TCAM과 SRAM의 장점을 혼합한 하이브리드 방식도 제안되었다[7]. 이 방식에서, 짧은 프리픽스는

SRAM을 사용하여 검색되고, 긴 프리픽스는 SRAM에 의해 인덱싱되는 TCAM을 사용하여 검색된다. 또한 프리픽스 수가 균등하도록 TCAM을 bank 단위로 분할하여 전력 소비를 최소화했다.

본 논문에서는 기존에 제안된 하이브리드 방식의 IP 주소 검색에 병렬성을 부여하고, 내재적인 충돌을 최소화하여 효율성을 높인 검색 기법을 제안한다. 전체 시스템은 네 단계로 구분되며, 단계별 파이프라인 방식을 적용한다. 단계 3에서, 분할된 다수의 TCAM이 동시에 동작하게 함으로써 병렬화하는데, 이를 위해 단계 2에서 여러 개 SRAM을 병렬로 사용한다. 또한 단계 2에서 단계 3으로 진행되는 패킷들 간에 충돌이 발생할 수 있는데, 단계 1에서 스마트 분배기를 개발하여 적용함으로써 성능을 더욱 향상시킨다.

본 논문의 구성은 다음과 같다. 먼저, 2장에서 IP 주소 검색의 병렬화와 프리픽스 분할에 대한 기존 연구 내용에 대해 알아보고, 기존 하이브리드 방식에 대한 기능과 특징에 대해 설명한다. 3장에서는 본 연구에서 제안한 병렬 IP 주소 검색 기법을 서술한다. 그리고 4장에서는 실제 사용되는 포워딩 테이블과 Zipf 분포에 따라 생성된 트래픽을 사용하여, 제안된 기법의 성능을 비교 평가한다. 마지막으로 5장에서 결론을 맺는다.

## II. 관련 연구

### 1. 병렬 IP 주소 검색과 프리픽스 분할

IP 주소 검색의 병렬화는 성능을 향상시키기 위한 방법으로 많이 사용되어 왔다. 일반적으로 TCAM 기반으로 하는 시스템의 경우는 TCAM을 병렬 수행이 가능한 몇 개의 블록으로 나누고, 전단에 인덱스 로직을 통해 연결시킴으로 병렬화하는 방법을 주로 사용한다[8-11]. SRAM의 경우는 독립성이 유지되도록 분리하여 동시에 수행하게 하거나, trie에서 레벨 단위의 검색을 각 단계로 구분하여 파이프라이닝을 통해 병렬화하는 방법을 주로 사용한다[12][13]. 즉 독립적인 수행 부분에는 병렬화 기법을, 순차적인 수행 부분에는 파이프라인 기법을 활용한다.

Zheng 등은 간단한 비트의 선택으로 인덱스를 결정하는 기법을 사용하였고, 자주 사용되는 TCAM 블록을 중복시켜서 부하의 균형을 이루고 있다[8]. Akhbarizadeh 등은 범위 기반의 인덱스 논리를 사용하여 MSMB (Multi-Selector and Multi-Block)라는 병렬 주소검색 구조를 제안했다[9]. 이 구조에서는 선택기와 블록의 개수가 늘어남에 따라, 복잡성이 상당히 높아지는 제한점이 있다. Lin 등은 pre-order splitting라는 trie 기반의 분할 기법을 제안했다[10]. Sun 등은 TCAM에 들어가는 프리픽스가 다른 것들과 중첩되는 것들과 중첩되지 않은 것들로 구분되는 새로운 분할기법을 제안하였다[11]. 대부분의 TCAM 기반 주소검색 기법들은 전단에 인덱스 논리를 사용하고 있다. Kim이 제안한 하이브리드 주소검색 기법[7]은 SRAM과 분할된 TCAM을 사용하는데, 전단의 SRAM은 인덱스 기능 뿐 아니라 주소 검색 자체의 기능을 포함하고 있다.

## 2. 하이브리드 IP 주소 검색 기법

하이브리드 방식의 IP 주소 검색 구조[7]는 [그림 1]과 같다. [그림 1]에서 보는 바와 같이, 각각 SRAM과 TCAM으로 구성되는 최대 2개의 사이클로 검색이 완료된다. 첫 번째 사이클인 SRAM은 다음 사이클에서 사용되는 TCAM bank에 대한 인덱스(bank index) 기능과, 짧은 프리픽스에 대한 주소 검색 자체 기능 등 두 가지 기능을 수행한다. 즉 32비트인 IP 주소에 대한 LMP의 길이가 16비트 이하인 짧은 프리픽스인 경우는 사이클 1에서 검색이 완료된다.

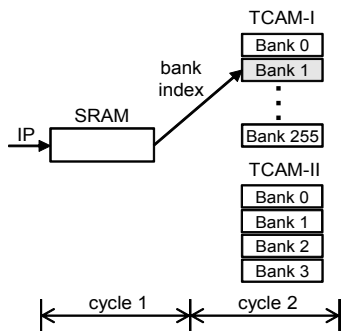


그림 1. 하이브리드 IP 주소 검색 구조

만일 긴 프리픽스인 경우는 사이클 2로 검색이 이어지는데, 길이가 17에서 24인 것은 TCAM-I, 길이가 25에서 32인 것은 TCAM-II에 각각 저장된 프리픽스를 찾기 위해 해당 bank에 접근한다. 예를 들어, LMP가 TCAM-I의 bank 1에 들어있다면, 사이클 1에서 IP 주소의 상위 16비트에 의해 SRAM의 엔트리를 구하고, 그 엔트리의 bank 인덱스 정보에 의해서 사이클 2에서는 TCAM-I의 bank 1만 활성화시키고 접근하여 LMP를 구한다.

TCAM-I과 TCAM-II 내의 bank들은 동일 유형에서 각각 상호 독립적으로 분할된 것이고, bank 내의 엔트리 수 등을 고려하여 각각 256개와 4개의 bank로 이루어져 있다. 하나의 IP 주소를 검색할 때, TCAM-I의 경우 1개의 bank만 활성화되고 나머지 255개의 bank는 비활성화된다. 본 논문에서는 다수의 bank가 동시에 활성화되도록 개선하여 병렬화함으로써, 기존의 방식에 비해 성능을 향상시킨다.

## III. 효율적인 병렬 IP 주소 검색 구조

### 1. 병렬 IP 주소 검색 엔진의 구성

병렬 IP 주소 검색 엔진은 [그림 2]와 같이 전체 4 단계로 구성된다. 시스템의 각 단계는 파이프라인에서 한 개의 단계에 대응된다. 단계 1에서 분배기(distributor)는 연속해서 입력되는 패킷들의 착신 IP주소를 적절한 SRAM으로 전달하도록 한다. 단계 2는 동일한 내용을 갖는 다수의 병렬 SRAM들로 구성된다. 각 SRAM의 엔트리는 출력포트 정보, TCAM-I의 bank 인덱스 정보, TCAM-I과 TCAM-II 내의 프리픽스 존재 여부를 나타내는 플래그(존재 플래그) 각 1비트(존재하는 경우 1) 등 3가지 정보를 갖고 있다. 단계 3은 256개 bank를 갖는 TCAM-I과 4개의 bank를 갖는 TCAM-II로 구성된다. 단계 2에서 단계 3으로 진행시 발생 가능한 내부 충돌에 따른 지연 등으로 처리된 순서가 입력 순서와 다른 상황이 발생할 수 있다. 단계 4에서 재정렬 버퍼(reorder buffer)는 이러한 상황을 해결하기 위해, 패킷의 입력 순서에 맞도록 재정렬하는데 사용된다.

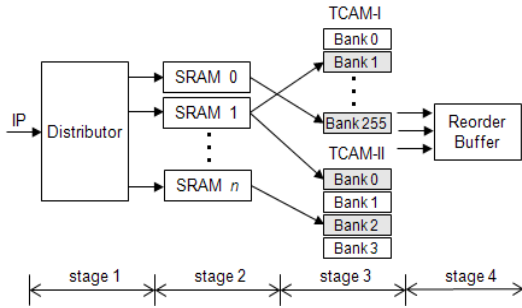


그림 2. 병렬 IP 주소 검색 구조

## 2. 병렬 SRAM과 분할 TCAM을 이용한 병렬 검색

[그림 2]에서 단계 2와 3은 기존 하이브리드 방식의 경우와 유사하게 각각 SRAM과 TCAM으로 이루어지는데, 단계 3에서 TCAM-I은 길이가 17에서 24인, TCAM-II는 25에서 32인, 프리픽스들을 저장한다. 그런데 여러 개의 bank로 분할하여 저장하므로 상호 독립적이며, 동일한 유형의 TCAM에서 다수의 bank들을 동시에 활성화되도록 한다. 이를 위해 단계 2에서 동일한 내용의 SRAM을 여러 개 중복하여, 다수 TCAM bank를 선택할 수 있도록 한다. [그림 2]의 예에서 보는 바와 같이, 다수의 SRAM을 병렬로 사용하여, TCAM-I은 bank 1과 bank 255가, TCAM-II는 bank 0과 bank 2가 동시에 활성화된다.

단계 2의 모든 SRAM은 동일한 정보를 저장하고 있으며, 하나의 SRAM은 길이가 16이하인 모든 프리픽스를 16비트로 확장하여, 전체  $2^{16}$ 개의 엔트리를 갖고 있다. LMP의 길이가 16이하인 경우 이 단계에서 검색이 완료되고 출력포트 정보를 구한다. 이 경우는 TCAM-I과 TCAM-II 존재 플래그 비트가 모두 0이다. 만일 LMP의 길이가 17이상인 경우는 단계 3의 TCAM에서 주소 검색을 계속해야 한다. TCAM-I 존재 플래그가 1인 경우, SRAM의 엔트리에 포함되어 있는 TCAM-I의 bank 인덱스 정보를 사용하여 검색을 계속 진행한다. 또한 TCAM-II 존재 플래그가 1인 경우는, IP 주소의 특정한 두 비트를 이용해 TCAM-II의 bank 번호를 선택한다. 여기서, TCAM-I 존재 플래그와 TCAM-II 존재 플래그가 모두 1인 경우는 TCAM-I과 TCAM-II를

모두 접근하여 검색 결과를 구한다. 단계 3의 TCAM-I, TCAM-II에서 모두 매칭이 일어난다면, TCAM-II, TCAM-I, SRAM의 순서에 따라 최종 LMP를 구한다.

단계 3의 TCAM-I에서 주소 검색은 SRAM에서 선택된 엔트리의 bank 인덱스에서 지정된 bank들만이 활성화되어 이루어진다. 그런데 두 개 이상의 SRAM에서 지정된 bank 인덱스가 같아 하나의 bank로 접근을 시도하면, 충돌이 발생하게 된다. 이러한 경우, 모두 처리하는 것은 불가능하므로, 한 개만 접근이 허용되고 나머지는 일시 중단(stall)했다가 다음 사이클에서 실행의 기회를 갖는다. 충돌이 발생한 것들에 대해 기회는 형평성을 제공하기 위해, 다음 사이클에서 라운드 로빈(round-robin) 방식의 선택을 한다. 동일한 bank로의 접근 시도가 많을수록 충돌로 인한 성능 저하가 심각해지므로, 동일한 주소들이 연속해서 또는 인접해서 단계 2의 SRAM으로 진행되는 경우를 최대한 피하는 것이 바람직하다.

## 3. 분배기와 재정렬 버퍼

분배기와 재정렬 버퍼는 각각 첫 단계와 마지막 단계에 해당한다. 단계 1의 분배기는 IP 주소들을 단계 2의 SRAM들로 분배하는 기능을 수행하고, 단계 4의 재정렬 버퍼는 패킷이 출력되는 순서를 입력된 순서에 맞추도록 순서를 재정렬한다.

### 3.1 단순 분배기

단순 분배기는 IP 주소를 입력된 순서에 따라 SRAM에 라운드 로빈 방식으로 분배하는 구조이다. 가장 간단한 방식으로 구현이 용이하지만, 동일하거나 유사한 IP 주소들이 연속해서 입력될 경우, 단계 2에서 단계 3으로의 충돌을 완화시킬 방법이 전혀 없다. 따라서 시간 및 공간 지역성을 갖는 인터넷 트래픽 특성상, 단순 분배기는 병렬화에 의한 성능 개선 효과를 감소시킬 수 있다. 단순 분배기를 갖는 시스템의 경우, 재정렬 버퍼도 간단하게 구성된다. 즉 주소 검색이 완료된 패킷은 재정렬 버퍼에 임시 저장되는데, 그것의 입력 순서번호와 일치하는 위치에 저장하면 된다.

### 3.2 스마트 분배기와 재정렬 버퍼

동일 IP 주소 여러 개가 일정한 시간 지역성을 갖고 들어오는 경우, 파이프라인에서 불필요하게 동일한 정보를 검색해야 할 뿐 아니라, 단계 3의 TCAM 접근 시 동일 bank에서 충돌이 발생하게 된다. 동일 IP 주소에 대한 검색의 경우 파이프라인 단계 2로 진입시키지 않고 선행 검색 결과를 그대로 활용한다면, 검색 처리를 향상과 함께 TCAM bank의 접근 충돌도 줄일 수 있다.

[그림 3]은 동일 IP 주소의 진입을 제어하는 스마트 분배기의 구성을 보여준다. 순차적으로 들어오는 IP 주소에 대해 선행하는 동일 IP 주소가 이미 파이프라인에 들어와 있는 지 탐색을 수행하기 위해 CAM (Content Addressable Memory)을 사용한다. CAM은 수십 개의 IP 주소를 저장할 수 있는 엔트리들로 구성되며, 병렬로 탐색 가능하다. CAM 엔트리 개수는 최대 32개까지 구성되므로 단계 1의 접근 시간은 파이프라인에서 가장 지연시간이 큰 단계 3의 TCAM 접근 시간에 비해 충분히 작다. 따라서 스마트 분배기는 단계 2의 8개 SRAM을 원활하게 동작시킬 만큼의 충분한 속도를 제공할 수 있다.

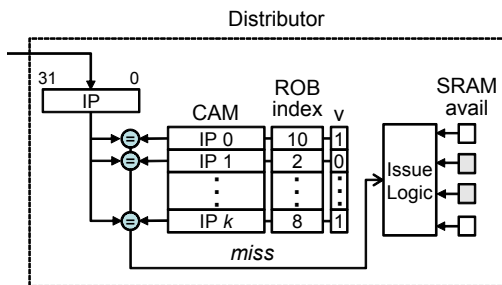


그림 3. 스마트 분배기

스마트 분배기에서 v비트는 각 CAM 엔트리의 유효성을 나타내는 것으로써, 초기에 0의 값을 가지며 엔트리에 IP주소가 저장되면 1로 기록된다. 검색이 완료된 IP 주소가 파이프라인을 빠져나가고 해당 IP 주소와 동일한 IP 주소들이 모두 처리되고 나면, 스마트 분배기의 해당 v비트를 0으로 표시한다.

입력 IP 주소와 동일한 IP 주소가 CAM 엔트리들 중

에서 발견되는 경우, 파이프라인 상에 진행 중인 선행 IP 주소가 이미 있다는 것을 의미하므로 파이프라인에 진입하지 않는다. 대신 선행 IP 주소의 검색 결과를 공유할 수 있도록 한다. 선행 IP 주소의 검색 결과가 저장될 재정렬 버퍼에는 해당 값이 복사되어 저장될 위치가 기록된다. [그림 4]는 재정렬 버퍼의 구성을 보여준다. 재정렬 버퍼의 각 엔트리는 검색 결과(output port), 검색이 완료되어 값이 저장되어 있는 지 여부(valid), 동일 IP가 존재하는 경우 결과 값을 복사할 위치(duplicate)로 구성된다.

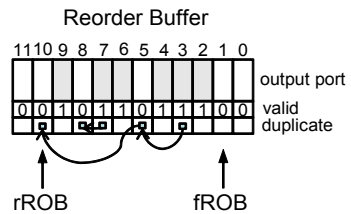


그림 4. 재정렬 버퍼

예를 들어, 3개의 패킷 A, B, C가 모두 동일 IP 주소를 갖고 있다고 해보자. 첫 번째 패킷 A가 진입하였을 때 해당 IP 주소는 CAM에서 발견되지 않으며(miss), 이러한 경우 IP 주소는 CAM에 저장되며 SRAM의 이용 가능 여부에 따라 파이프라인 단계 2로 진입하게 된다. 또한 최종 검색 결과를 저장하기 위한 재정렬 버퍼 엔트리 한 개가 할당되며, 해당 엔트리 인덱스(ROB index)가 CAM 엔트리와 연동된 위치에 기록된다. 두 번째 패킷 B가 진입하면, 해당 IP는 CAM에서 발견되며(hit), 이 경우 ROB index 값만 패킷 B에 대해 할당한 재정렬 버퍼의 인덱스 값으로 갱신된다. 또한 재정렬 버퍼에서는 duplicate 포인터를 통해 두 엔트리들이 연결된다. 세 번째 패킷 C도 동일한 방식으로 처리되며, [그림 4]는 이들 패킷들의 ROB index가 각각 3, 5, 10인 경우를 보여준다.

[그림 4]에서 3개의 재정렬 버퍼의 엔트리들이 duplicate 포인터로 연결되어 있으며, 파이프라인에서 실제 검색이 수행되어 검색 결과가 저장되는 엔트리는 ROB index 값이 3인 경우 뿐이다. 파이프라인 단계 4

표 1. 처리율 (packets / cycle)

SRAM	a = 0.7			a = 1.0			a = 1.2		
	<i>npp</i>	<i>pp</i>	<i>spp</i>	<i>npp</i>	<i>pp</i>	<i>spp</i>	<i>npp</i>	<i>pp</i>	<i>spp</i>
1	0.494	1.000	1.001	0.492	1.000	1.023	0.489	1.000	1.112
2	0.494	1.998	2.001	0.492	1.991	2.097	0.489	1.973	2.496
3	0.494	2.995	3.001	0.492	2.971	3.215	0.489	2.909	4.109
4	0.494	3.991	4.002	0.492	3.937	4.374	0.489	3.790	5.885
5	0.494	4.985	5.002	0.492	4.888	5.567	0.489	4.596	7.711
6	0.494	5.977	6.002	0.492	5.822	6.792	0.489	5.301	9.188
7	0.494	6.966	6.999	0.492	6.731	8.034	0.489	5.899	9.963
8	0.494	7.954	7.995	0.492	7.620	9.144	0.489	6.394	10.260

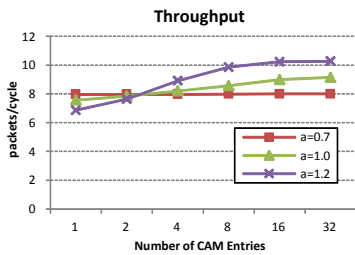


그림 5. 처리율

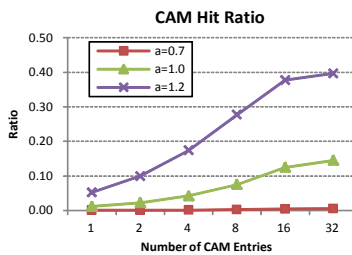


그림 6. CAM 적응율

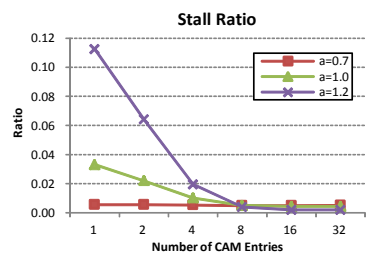


그림 7. 파이프라인 지연 비율

에서는 재정렬 버퍼의 앞쪽(rROB)에서부터 차례대로 저장된 검색 결과를 내보냄으로써 순서를 유지한다. 모든 IP 주소는 파이프라인 단계 1에서 재정렬 버퍼의 뒤쪽(rROB)부터 순차적으로 엔트리를 할당받았기 때문에 순서가 유지될 수 있다. 저장된 검색 결과를 내보낼 때 duplicate 포인터가 가리키는 엔트리가 있다면 결과를 해당 위치로 복사함으로써, 동일 IP 주소들에 대한 결과 값을 전파할 수 있다.

킷 트레이스는 무작위적으로 발생시키되 Zipf의 법칙을 따르도록 하였다. 실제 IP 주소들의 분포는 Zipf와 유사한 분포를 따른다는 것이 알려져 있다[15].

$$P(\text{Rank}) \sim 1 / \text{Rank}^a$$

Zipf의 법칙에 따르면, IP 주소들의 빈도별 순위 (Rank)에서 각 빈도값은 지수함수의 역수에 비례한다. 지수값 a는 네트워크 응용 또는 흐름에 따라 달라질 수 있는 값이지만, 가장 흔한 경우로 예상되는 0.7 ~ 1.2의 범위의 값을 실험에 사용하였다.

## IV. 성능 평가

### 1. 실험환경

성능 평가를 위해 4단계 병렬 파이프라인 구조를 시뮬레이터로 구현한 후, 이를 통해 사이클 단위의 시뮬레이션을 수행하였다. 시뮬레이션 입력에 필요한 라우팅 테이블은 [14]로부터 얻은 실제 테이블을 이용하였으며, 총 프리픽스 개수는 234,153으로 16비트 이하, 17 이상 24비트 이하, 25비트 이상 프리픽스 개수는 각각 11,419(4.9%), 218,930(93.5%), 3,804(1.6%)이다. 또한 패

### 2. 실험 결과 및 토의

[표 1]은 처리율(throughput)을 비교한 결과이다. 표에서 *npp*(non parallel and pipelined)는 단순 하이브리드 IP 주소 검색기로서 병렬처리를 하지 않은 경우를 나타낸다. *pp*(parallel and pipelined)는 병렬처리와 파이프라이닝을 사용하지만, 동일 TCAM bank에서 접근 충돌이 생길 수 있으며, 이로 인해 파이프라인 지연이 발생할 수 있는 구조로서 단순분배기를 사용하는 시스

템의 경우이다. *spp*(smart parallel and pipelined)는 본 논문에서 제안한 스마트 분배기를 사용한 병렬처리와 파이프라이닝을 하는 구조이다.

실험 결과 제안한 *spp*의 처리율은 *npp*에 비해 병렬성에 따라 약 2~21배 향상됨을 보였다. 또한 *pp*에 비해서도 1~1.6배 우수함이 확인되었다. 특히, *pp*의 경우 *a*값이 커질수록 성능은 떨어지는데 반해, *spp*는 오히려 성능이 향상됨을 보여준다. 이는 *spp*가 스마트 분배기를 이용하여 중복 IP 주소들의 TCAM bank 접근 충돌을 효과적으로 회피할 뿐 아니라, 선행 검색 결과를 활용함으로써 성능을 향상시키기 때문이다.

[그림 5]는 스마트 분배기내에 존재하는 CAM 엔트리 개수에 따른 처리율의 변화를 보여준다. 병렬성을 의미하는 SRAM 개수는 8개로 고정하였다. 엔트리 개수가 증가하면 전반적으로 처리율은 향상되며, 특히 *a*값이 클수록 향상 정도가 커진다.

[그림 6]은 입력된 IP 주소가 CAM에서 적중되는 비율을 나타낸다. 엔트리 개수가 늘어남에 따라 적중률은 향상되며, *a*값이 크면 향상 폭이 급격히 커진다.

파이프라인 지연율은 CAM 엔트리 개수가 증가함에 따라 감소하게 된다. [그림 7]은 이러한 변화를 보여준다. *a*값이 크면 감소 폭도 크지만, 이 값이 낮을수록 애초 지연율이 낮다는 점에 주목할 필요가 있다. *a*값은 IP 주소들의 빈도값에서 집중도를 의미하는 것으로서 이 값이 낮다는 것은 TCAM bank 접근 충돌로 인한 파이프라인 지연 발생 가능성도 낮다는 것을 의미한다. [그림 5]에서 CAM 엔트리 수가 1일 때 *a*값이 클수록 오히려 처리율이 낮은 것은 이러한 데에 기인한다.

## V. 결론

IP 주소 검색 성능을 효과적으로 향상시키기 위해서는 메모리의 특성을 잘 활용하도록 검색 자료구조를 구성 배치하여야 한다. 본 논문에서는 접근 시간과 전력 소비에 있어 우수한 SRAM과 가변길이의 프리픽스를 저장하기 용이한 TCAM을 혼합한 하이브리드 형태의 IP 주소 검색기에 대해 성능 향상을 위한 구조 개선을

제안하였다.

다중 SRAM과 분할 TCAM을 사용함으로써, 비교적 저렴한 비용으로 병렬 검색이 가능하도록 하였으며, 또한 파이프라이닝을 사용하여 처리율이 향상되도록 하였다. 그러나, 동일 IP 주소가 복수로 들어오는 경우 동일 TCAM bank에서 접근 충돌이 발생하고, 이로 인해 파이프라인 지연이 생긴다.

본 논문에서 제안한 스마트 분배기와 재정렬 버퍼는 동일 IP 주소의 파이프라인 진입을 효과적으로 막음으로써, 성능 향상과 TCAM bank의 접근 충돌을 감소시킨다. 인터넷 트래픽의 특성을 반영하는 트래픽 분포와 실제 라우팅 테이블을 사용하여 시뮬레이션을 수행하였다. 시뮬레이션 결과를 통해, 병렬 구조의 성능이 기존의 하이브리드 구조에 비해 더욱 우수했으며, 스마트 분배기를 사용하는 병렬 구조의 경우는 향상 정도가 매우 크게 나타났다.

## 참고 문헌

- [1] H. J. Chao, "Next Generation Routers," Proc. of IEEE, Vol.90, No.9, pp.1518-1558, 2002.
- [2] Mi. A. Ruiz-Sanchez, E. W. Biersack, and W. Dabbous, "Survey and Taxonomy of IP Address Lookup Algorithms," IEEE Network, Vol.15, No.2, pp.8-23, 2001.
- [3] H. J. Chao and B. Liu, *High Performance Switches and Routers*, Wiley-Interscience, 2007.
- [4] W. Wu, *Packet Forwarding Technologies*, Auerbach Pub., 2008.
- [5] 김진수, 김정환, "효율적인 IP 주소 검색을 위한 개선된 LC-trie", 한국콘텐츠학회논문지, 제7권, 제3호, pp.50-59, 2007.
- [6] G. Wang and N.-F. Tzeng, "Exact Forwarding Table Partitioning for Efficient TCAM Power Savings," Proc. of IEEE NCA 2007. pp.249-252, 2007.
- [7] J. H. Kim, M. C. Ko, H. K. Kang, and J. S. Kim,

“A Hybrid IP Forwarding Engine with High Performance and Low Power,” Proc. of the ICCSA 2009, pp.888-899, 2009.

[8] K. Zheng, C. C. Hu, H. B. Lu, and B. Liu, “A TCAM-based Distributed Parallel IP Lookup Scheme and Performance Analysis,” IEEE/ACM Transation on Networking, Vol.14, No.4, pp.863-875, 2006.

[9] Mohammad J. Akhbarizadeh, Mehrdad Nourani, Rina Panigrahy, and Samar Sharma, “A TCAM-based Parallel Architecture for High-speed Packet Forwarding,” IEEE Transactions on Computers, Vol.56, No.1, pp.58-72, 2007.

[10] D. Lin, Y. Zhang, C. Hu, B. Liu, X. Zhang, and D. C. Pao, “Route Table Partitioning and Load Balancing for Parallel Searching with TCAMs,” Proc. of 21st IPDPS, pp.1-10, 2007.

[11] Y. Sun, H. Liu, and M. S. Kim, “Using TCAM Efficiently for IP Route Lookup,” Proc. of 8th IEEE Consumer Communications and Networking Conference, pp.816-817, 2011.

[12] Zoran Cica and Aleksandra Smiljanic, “Frugal IP Lookup Based on Parallel Search,” Proc. of International Conference on High Performance Switching and Routing (HPSR), pp.1-6, 2009.

[13] Weirong Jiang and Viktor K. Prasanna, “Parallel IP Lookup using Multiple SRAM-based Pipelines,” Proc. of 22nd International Parallel and Distributed Processing Symposium (IPDPS), pp.1-14, 2008.

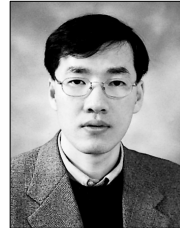
[14] <http://www.routeviews.org/>

[15] W. Shi, M. H. MacGregor, and P. Gburzynski, “An Adaptive Load Balancer for Multiprocessor Routers,” Simulation, Vol.82, No.3, pp.173-192, 2006.

저 자 소 개

김 정 환(Junghwan Kim)

정회원



- 1991년 2월 : 서울대학교 계산통계학과(이학사)
  - 1993년 2월 : 서울대학교 대학원 전산과학과(이학석사)
  - 1999년 2월 : 서울대학교 대학원 전산과학과(이학박사)
  - 1999년 ~ 2000년 : 삼성전자 통신연구소 연구원
  - 2001년 ~ 현재 : 건국대학교 컴퓨터공학과 교수
- <관심분야> : 병렬처리, 분산시스템, RFID 시스템

김 진 수(Jinsoo Kim)

정회원



- 1983년 2월 : 서울대학교 컴퓨터 공학과(공학사)
  - 1985년 2월 : 한국과학기술원 전산학과(공학석사)
  - 1998년 8월 : 한국과학기술원 전산학과(공학박사)
  - 1985년 4월 ~ 2000년 2월 : KT 선임연구원
  - 2000년 3월 ~ 현재 : 건국대학교 컴퓨터공학과 교수
- <관심분야> : 병렬처리, 고속통신망, 센서네트워크