PICAM에서의 최적 파이프라인 구조

정회원 안 희 일*, 조 태 원**

The Optimal pipelining architecture for PICAM

Hee Il Ahn*, Tae Won Cho** *Regular Members*

요 약

고속 IP 주소 룩업(lookup)은 고속 인터넷 라우터의 성능을 좌우하는 주요 요소이다. LPM(longest prefix matching) 탐색은 IP 주소 룩업에서 가장 시간이 많이 걸리는 부분이다. PICAM[15]은 고속 LPM 탐색을 위한 파이프라인 CAM구조로서, 기존 CAM(content addressable memory, 내용 주소화 메모리)을 이용한 방법보다 룩업 테이블의 갱신속도가 빠르면서도 LPM 탐색율이 높은 CAM구조이다. PICAM은 3단계의 파이프라인으로 구성된다. 단계 1 및 단계2의 키필드분할수 및 매칭점의 분포에 따라 파이프라인의 성능이 좌우되며, LPM 탐색율이 달라질 수 있다. 본 논문에서는 PICAM의 파이프라인 성능모델을 제시하고, 이산사건 시뮬레이션(discrete event simulation)을 수행하여, 최적의 PICAM구조를 도출하였다. IP version 4인 경우 키필드분할수를 8로 하고, 부하가 많이 걸리는 키필드블록을 중복 설치하는 것이 최적구조이며, IP version 6인 경우 키필드블록의 개수를 16으로 하는 것이 최적구조다.

ABSTRACT

Fast IP address lookup is a major factor in high speed Internet router. LPM searching is the most time consuming process in IP address lookup. PICAM is a pipeline CAM architecture for high speed LPM searching. It has faster LMP searching and lookup table update than existing methods have. It has three pipeline stages. It's LPM searching rate is affected by both the number of keyfield blocks in stage 1 and stage 2, and distribution of matching point. In this paper, we propose a performance model for PICAM and have done discrete event simulations on it, and show an optimal architecture of PICAM as follows. In case of IP version 4, a number of keyfield blocks are 8, and the heavy loaded keyfield block is duplicated. In case of IP version 6, a number of keyfield blocks are 16.

I. 서 론

인터넷 패킷 통신에서 패킷의 행선지 주소(destination address)에 따라 패킷이 전달될 출력 포트를 결정하는 것을 IP 주소 룩업(address lookup)이라 한다. IP 주소 룩업은 인터넷 통신에서 패킷 처리 능력을 결정하는 중요한 요소이다. 인터넷의 속도는 통신선로가 아무리 빠르더라도 IP 주소 룩업 처리 능력이 작으면, 고속의 인터넷 통신이 불가능 해진다. 최근 기가 비트급 이상의 초고속 인터넷 통신에서는 IP 주소 룩업 처리 능력이 병목현상을 일으키고 있다.

인터넷 통신의 초기에는 IP 주소에는 3가지 class가 있었고, 각각 8비트, 16비트, 24비트의 고정길이 IP 주소를 사용하였다. 고정길이 IP 주소의 룩업에는 패킷의 목적지 IP 주소와 모든 비트가 정확히 일치하는 엔트리를 룩업 테이블(lookup table)에서 찾아내는 EM(exact matching)탐색[1, 2]을 사용했다. EM탐색은 perfect hashing, binary search 또는 일반적인 CAM을 사용해 빠르게 수행할 수 있다.

* 충북대학교 전기전자공학부
논문번호 : K01082-0227, 접수일자 : 2001년 2월 27일

1990년 초반 CIDR(classless inter-domain routing) [7, 13]의 도입으로 IP 주소는 가변길이를 갖게 되었으며, 동일 link로 도달할 수 있는 IP 주소들을 하나의 그룹으로 묶어 하나의 prefix로 표현하고, 이 prefix를 룩업 테이블의 엔트리로 사용함으로써, 룩업 테이블의 엔트리 개수를 대폭 줄일 수 있었다. 그러나 가변길이의 IP 주소로 인해 EM탐색에 비해 시간이 많이 걸리는 LPM(longest prefix matching) 탐색을 수행하여야 한다는 문제점이 발생했다. LPM 탐색의 어려움은 입력 패킷의 IP 주소의 prefix 길이를 미리 알 수 없다는데 있다. 더구나 입력 패킷에 대해 여러 개의 매치 가능한 prefix중 가장 길게 매치되는 prefix를 찾아내야 하기 때문에 탐색시간이 오래 걸려, 고속 IP 주소 룩업에서 LPM 탐색은 병목현상을 일으키고 있다.

LPM 탐색에는 perfect hashing, binary search 또는 일반적인 CAM과 같은 EM 탐색에 사용하던 방법들을 적용할 수 없게 되었다. LPM 탐색에서 중요한 것은 LPM 탐색율이 높으면서 룩업 테이블의 갱신 속도 또한 빨라야 한다는 점이다. 그러나 기존의 LPM 탐색에 대한 연구들에서는 룩업 테이블의 효율적인 갱신은 고려하지 않고, LPM 탐색율의 향상에만 초점을 맞춰왔다[2, 4, 5, 13].

LPM 탐색을 위해 기존 방법에는 소프트웨어적인 방법과 하드웨어적인 방법들이 있다. 소프트웨어적인 방법[2, 4 - 11]은 binary search를 기반으로 하여 LPM 탐색을 처리하는 방법으로서, 여러 차례 메모리를 참조해야 하기 때문에 처리능력이 낮다. 라우터의 룩업테이블을 압축하여 캐쉬(cache)나 메인메모리에 탑재하여 속도를 올리고 있으나, 룩업테이블의 갱신을 쉽게 할 수 없다는 단점이 있다.

하드웨어적인 방법[1,12]에는 CAM을 이용한 3가지 기존 방법이 있다. 첫번째는 일반적인 binary CAM을 이용해 여러 사이클에 LPM탐색을 수행하는 방법으로써, 구조는 비교적 간단하고 룩업테이블의 갱신속도도 빠르나, LPM 탐색율이 매우 낮다는 단점이 있다. 두번째는 prefix 길이별로 일반적인 binary CAM 모듈로 동시에 탐색하고, 그 결과 중 prefix가 제일 긴 것을 LPM탐색의 결과로 하는 방법이다. 이 방법은 LPM 탐색율도 높고 룩업테이블의 갱신속도도 빠르나, 구조의 복잡도가 너무 높은 단점이 있다. 세번째는 ternary CAM을 이용한 방법으로 LPM탐색율도 높고 구조도 비교적 간단하나, 룩업테이블의 갱신속도가 매우 느리다는 단점이 있다.

기존 LPM 탐색방법들의 단점을 개선하고자 3단계 파이프라인 CAM 구조(PICAM)[15]을 이용한 LPM 탐색방법이 제안되었다. PICAM을 이용한 방법은 LPM 탐색율이 높으면서도 룩업테이블의 갱신속도가 빠르며, 또한 구조의 복잡도도 높지않다. PICAM에서는 하나의 패킷에 대해 3단계의 파이프라인 처리를 하고, 또 키필드를 여러 개의 키필드블록으로 나누어 병렬탐색을 수행한다. 따라서 한 패킷에 대한 LPM 탐색시간은 길지만, LPM 탐색율은 높다. 일반적 CAM의 갱신 용이성 때문에 PICAM의 룩업테이블 갱신속도도 빠르며, 구조의 복잡도도 높지않다.

PICAM에는 성능에 영향을 미칠 수 있는 요소들이 있다. 첫째, 파이프라인 처리에서는, 각 단계의 처리 시간이 다르고, 단계1 및 단계2를 몇 개의 키필드블록으로 분할하는가(키필드분할수)에 따라 LPM탐색율이 달라질 수 있다. 둘째, 실제 라우터에서는 패킷의 IP주소의 prefix 길이 분포가 균일하지 않다. IP주소의 prefix 길이 분포가 균일하지 않으면 단계2의 각 키필드블록에 걸리는 부하(load)가 균일하지 않아, 파이프라인 처리의 효율을 떨어트릴 수 있다.

본 논문에서는 PICAM의 성능모델을 제시하고, 이산사건 시뮬레이션(discrete event simulation)을 통해 LPM 탐색율을 최대로 하는 각 단계의 키필드 분할수와 버퍼의 크기를 결정하고, 또한 입력 IP 주소의 prefix 길이 분포가 편향(skew) 되었을 때의 PICAM의 최적구조를 제안한다. II장에서는 PICAM의 구조를 살피고, III장에서는 시뮬레이션을 위한 PICAM의 성능모델을 제시하고, IV장에서는 시뮬레이션 결과에 대해 고찰한 후 PICAM의 최적구조를 제시한다.

Ⅱ. 고속 LPM 탐색을 위한 파이프라인 CAM 구조

PICAM은 binary CAM의 키필드와 데이터필드를 분리해 그림 1과 같이 3단계의 파이프라인 단계로 이루어진다. 파이프라인 처리이기 때문에 하나의 입력 IP주소에 대한 LPM 탐색 시간은 더 오래 걸리지만, 단위시간 당 처리하는 IP주소의 수인 LPM 탐색율(throughput)은 더 올릴 수 있다.

룩업 테이블은 {prefix, port number}로 이루어지는 엔트리로 구성된다. prefix는 동일한 출력링크로 도달될 수 있는 IP주소의 그룹이며, port number는

해당 prefix를 갖는 행선지 IP주소를 갖는 IP 패킷의 출력포트를 나타낸다. LPM 탐색은 입력 IP 패킷의 행선지 IP주소에 대해 가장 길게 일치하는 prefix를 찾아, 이에 해당하는 출력포트로 입력 IP패킷을 출력하는 과정이다.

기존 CAM을 사용하는 LPM방법에서는 룩업테이블의 prefix는 CAM의 키필드에 저장되고, 포트번호는 데이터필드에 저장된다. 이 경우 입력 패킷의 IP주소와 일치하는 키필드의 prefix가 히트되고, 이에 연관된 데이터필드의 포트번호가 출력된다.

기존 방법과는 달리 PICAM에서는 단계1 및 단계2는 각각 키필드가 있으며, 이 키필드는 m개의 키필드블록(KFB: keyfield block)으로 나누어진다. 룩업 테이블의 각 prefix는 단계1 및 단계2의 키필드에 중복 저장된다. 단계 1 및 단계2에서 prefix는 b비트씩 나누어 차례로 키필드블록0, 키필드블록1, ... , 키필드블록m-1의 같은 행에 저장된다. 여기서 b는 키필드블록의 폭으로서 $b = w/m$ 이고, w는 IP주소길이며, m는 키필드분할수로서 2의 멱수 즉 2, 4, 8, 16, 32, 이다. 룩업 테이블의 포트번호는 단계3의 데이터필드에 저장된다. 즉 단계1과 단계2의 키필드는 수평적으로 m개의 키필드블록으로 분리된다. LPM 탐색은 단계1의 블록단위 탐색, 단계2의 비트단위 탐색, 그리고 단계3의 데이터필드 출력의 순으로 진행된다.

단계1은 m개의 키필드블록과 1개의 제어모듈로 이루어진다. 키필드블록은 b n개의 CAM 메모리 셀로 이루어진다. 키필드블록이 m개 있으므로, 단계1은 m b $n = w$ n개의 CAM 메모리 셀로 이루어진다. 여기서 n는 룩업테이블의 엔트리 개수이다. 여기서 입력된 2진수로 표현된 IP주소를 왼쪽에서부터 b비트씩 나누어, 각 키필드블록은 블록단위의 매치여부를 탐지하고, 모든 키필드블록이 동시에 동작한다. 제어모듈은 각 키필드블록에서의 매치여부를 기반으로 어느 키필드블록까지 연속적으로 매치가 되는지 탐지하여 매치되지 않는 첫번째 키필드블록의 정보와 그때까지의 매치벡터를 다음 단계로 보낸다. 매치벡터는 처음에서 연속적으로 블록매치 되는 마지막 키필드블록까지의 각 엔트리별로 매치여부를 나타내는 어레이를 말한다. 단계1에서는

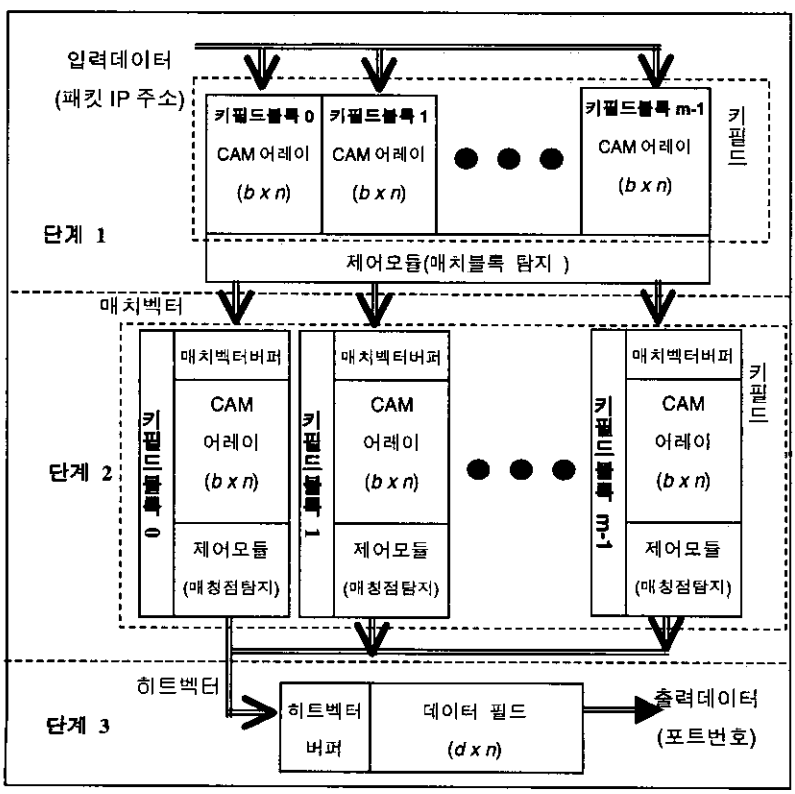

그림 1. PICAM의 구조

한 파이프라인 클록 주기 즉 한 사이클 동안에 한 개의 입력 IP 주소만 처리하며, 한번의 CAM 어레이 접근이 발생한다.

단계2는 m개의 키필드블록으로 이루어지며, 각 키필드블록은 매치벡터버퍼, CAM어레이 및 제어모듈로 이루어진다. 앞 단계에서 지정해준 키필드블록에 대해서만 어느 비트까지 매치되는지 정확한 매칭점(matching point)을 탐지한다. 매칭점 탐지에는 $\log_2 b$의 회수만큼의 CAM 어레이 접근이 일어난다. 키필드블록이 하나의 입력 IP주소에 대해 처리하는데 $\log_2 b$의 사이클이 소요된다. 그러나 키필드블록들이 m개가 있으며, 연속되어 같은 키필드블록이 지정되지 않는 한, 병렬로 동시에 작동될 수 있다. 최대 m개의 키필드블록이 동시에 매칭점 탐지를 수행할 수 있다. 따라서 한 사이클 당 처리되는 최대 입력 IP주소의 수는 m / ($\log_2 b$) 이다. 하나의 키필드블록에서 IP주소를 처리하고 있는데 중간에 동일한 키필드블록으로 그 다음 IP주소가 입력되는 패킷충돌(packet conflict)이 발생될 수 있다. 이 경우에 원활한 처리를 위해, 각 키필드블록의 입력단에는 매치벡터버퍼가 있다. 각 키필드블록의 제어모듈은 앞 단계에서 입력된 매치벡터와 현재의 키필드블록에서 탐지한 매칭점을 기반으로 히트벡터를 생성한다. 히트벡터는 각 엔트리별로 제일 길게 매치되는지를 나타낸다. 히트벡터는 LPM 매치되는 엔트리의 위치를 나타내며, 이는 다음 단계로 출력된다.

단계3은 히트벡터버퍼와 데이터필드로 구성되어 있다. 바로 앞 단계에서 입력된 히트벡터를 기반으로 데이터필드에서 해당되는 출력데이터를 외부로 출력한다. 데이터 필드의 접근이 한번 일어나므로 한 사이클에 하나의 IP주소를 처리된다. 앞 단계의 키필드블록들이 병렬로 동작하므로 한번에 하나 이상의 IP주소가 입력되는 패킷충돌이 발생될 수 있다. 이 경우에 원활한 처리를 위해, 입력단에 히트벡터버퍼가 있다.

그림 2에서는 IP version 4, m=8인 경우의 LPM 과정을 보여주고 있다. 입력데이터 128.32.195.1과 LPM되는 엔트리는 그림 2에서 굵은선으로 둘러싸인 행이다. 단계1의 빗금친 부분은 블록매치되는 키필드블록들을 나타내며, 단계2의 빗금친 부분은 매칭점을 찾아내야 하는 키필드블록을 보여주고 있다. 그림 2에서는 입력데이터와 LPM되는 엔트리는 비트0에서 비트21까지 일치하고 있으며, 매칭점은 비트21이다.

PICAM에서는 각 단계가 처리시간이 다르다. 단계1과 단계3의 처리시간은 파이프라인 주기 T이다. 단계2에서의 처리시간은 ($\log_2 b$)T이다. 예를 들어 b=4이면 2T이고, b=8이면 3T가 된다. 단계2에서는 처리시간은 길지만, 각 키필드블록이 병렬로 동작함으로써 단계2의 처리율을 단계1의 처리율에 접근시킬 수 있다.

키필드블록의 개수 m에 따라 파이프라인 효율에 영향을 줄 수 있다. m이 커지면 단계2의 처리시간이 짧아지고, 병렬성이 높아지나, 제어모듈의 복잡도가 커진다. 따라서 너무 m이 크지 않은 범위에서

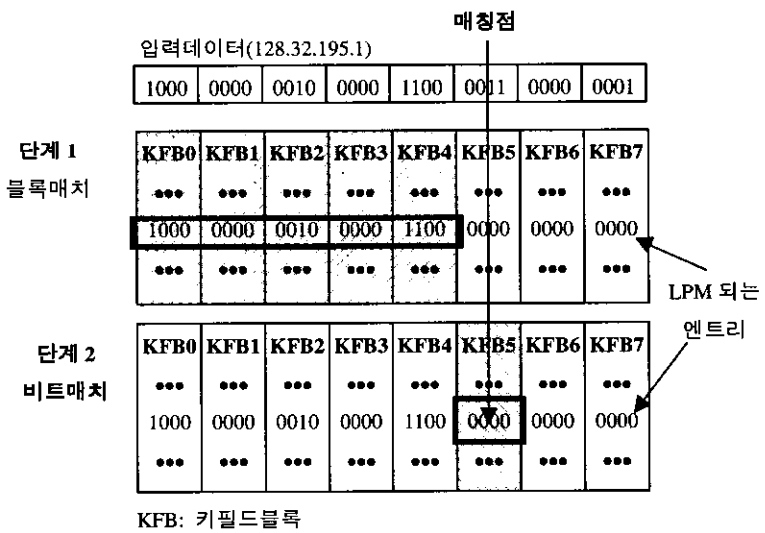

그림 2. PICAM에서의 LPM과정 및 매칭점

파이프라인 효율이 높은 구조가 바람직하다. 또한 단계2와 단계3에서는 패킷충돌로 인해 버퍼가 필요하다. 버퍼의 크기는 버퍼에서의 대기행렬(queue)의 예상 최대 길이로 하면 충분하다.

Ⅲ. 시뮬레이션을 위한 PICAM의 성능모델

PICAM은 단계1, 단계2, 및 단계3의 파이프라인 단계로 이루어진다.

단계1에서는 파이프라인 클록 한 주기 T동안 한 개의 IP주소에 대해 매치블록 탐색을 수행한다. 따라서 단계1의 처리율을 1/T이다. 또한 단계1의 처리시간 μ_1은 T이다. 단계2에서는 한 개의 키필드블록이 하나의 IP주소에 대해 매칭점 탐색을 수행하는데 걸리는 처리시간 μ_2 는 $(\log_2 b)$T이다. 여기서 b는 키필드블록의 폭이다. 단계3에서는 하나의 IP주소를 처리하는데 걸리는 처리시간 μ_3는 T이다. 단계3의 처리율은 1/T이다. 각 단계의 처리시간은 T 또는 $(\log_2 b)$T의 고정길이의 분포를 갖는다.

PICAM의 파이프라인 성능을 위주로 한 성능모델은 그림 3와 같다. 시뮬레이션을 위해 폐쇄회로 시스템(closed loop system)으로 가정한다. 편의상 LPM 탐색 요구를 고객으로 설정하고, 각 단계는 대기행렬(queue)과 서버(server)가 있다고 설정한다. 고객은 단계1이란 서버에서 블록매치 서비스를 받고 단계2에서는 비트매치 서비스를 받은 다음 단계3에서는 출력포트번호를 서비스 받으면, 고객에 대한 시스템의 서비스는 종료된다. 여기서 고객은 단계1에서 서비스를 받고 단계2로 갈 때 매칭점의 위치에 따라 단계2의 어느 키필드블록에 가서 서비스를 받을 것인가가 결정된다. 또한 고객은 단계1, 단계2 및 단계3을 거쳐 나온 후 다시 단계1로 들어간다고 하고, 시스템 내에는 q명의 고객이 있다고 가정한다.

단계 1의 입력단에도 대기행렬을 둔다. 단계1은 μ_1 의 서비스 시간을 갖는다. 단계2는 각 키필드블록을 하나의 대기행렬과 서버(server)로 설정한다. 독립된 m개의 키필드블록이 있으므로 m개의 대기행렬 및 서버가 있다. 단계2의 각 서버는 μ_2의 서비스 시간을 갖는다. 또한 단계3은 μ_3의 서비스 시간을 갖는 대기행렬 및 서버이다. 여기서 편의상 T=1로 놓으면, $\mu_1=\mu_3=1$, $\mu_2=(\log_2 b)$ 이다.

단계1의 처리과정이 끝난 후, 단계2의 어느 키필드블록으로 들어갈 것인가는 입력 IP 주소가 룩업 테이블과 LPM되는 엔트리가 어느 비트까지 매칭되는지 즉 매칭점에 따라 결정된다. 그러나 매칭점의 위치는 미리 알 수 없다. 매칭점에 대해 다음 두 가지 가정을 할 수 있다.

매칭점 가정 1: 매칭점은 모든 키필드블록에 고루 발생한다고 가정한다. 즉 키필드블록 k에 매칭점이 있을 확률을 p_k라 하면, 0에서 $m-1$까지의 모든 정수에 대해 $p_k = 1/m$ 이다.

매칭점 가정 2: 매칭점의 위치는 룩업 테이블의

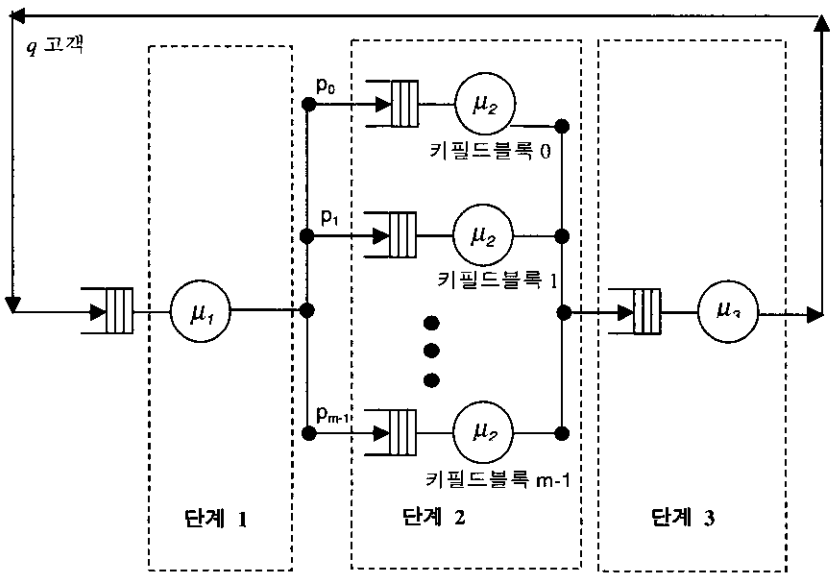

그림 3. PICAM의 성능모델

prefix길이의 분포에 따른다. 즉 키필드블록k에 매칭점이 있을 확률 p_k는 prefix길이가 $bk+1$에서 $b(k+1)$ 범위에 있는 엔트리 수에 비례한다. 여기서 b는 키필드블록의 폭으로서 IP주소길이 w를 m으로 나눈 값이다. 룩업테이블에서 prefix길이 r에 대한 엔트리의 수를 $N(r)$라면 $p_k = \sum_{r=bk+1}^{b(k+1)} N(r) / \sum_{r=1}^{w} N(r)$ 이다.

그림 4은 MAE-EAST 백본 라우터[16]에서의 룩업테이블의 prefix 길이 분포를 보여주고 있다. 8, 16 및 24비트에 엔트리들이 상당히 몰려있는 것을 볼 수 있다. 이는 CIDR이 도입되기 전에 고정길이 IP 주소의 영향 때문이다. 특히 24비트에 엔트리들이 많이 몰려있다. 매칭점 가정 2를 적용하면 매칭점은 계속 같은 블록에 발생하게 되어 단계2의 병렬성을 감소시킬 수 있다.

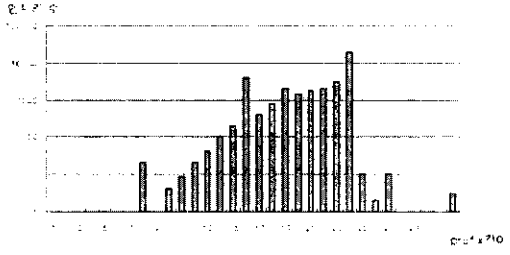

그림 4. 백본라우터에서의 룩업테이블의 prefix길이 분포 (MAE-EAST, 01/02/98)

본 논문에서는 매칭점 가정 1과 매칭점 가정2의 각 경우에 대해 시뮬레이션을 하여 차이점을 살펴보고, 병렬성 감소현상으로 인한 LPM탐색율 저하에 대한 PICAM구조의 대안을 제안하고자 한다. 여기서 IP version 4인 경우 매칭점 가정 1 및 매칭점 가정 2를 모두 적용해 시뮬레이션을 수행했으며, IP version 6인 경우에는 매칭점 가정 1만 적용해 시뮬레이션을 수행하였다. 매칭점 가정 2에 대해서는 그림 4와 같은 백본라우터의 룩업테이블의 prefix길이 분포를 구해 추후에 시뮬레이션 할 계획이다.

Ⅵ. 시뮬레이션 및 PICAM최적구조

앞장에서 기술한 PICAM의 성능모델로 이산사건 시뮬레이션을 수행하였다. 시뮬레이션 프로그램은 SMPL시뮬레이션 언어[17]로 작성하였고 크기는 110라인이다. 성능모델은 폐쇄회로시스템이다. 하나의 고객이 단계1 및 단계2를 거쳐 단계3에서 서비스를 다 받으면 다시 단계1로 돌아가 순환을 계속하며, 폐쇄회로시스템내에 10개의 고객이 존재한다. 단계1에서의 고객의 도착율은 파이프라인 전체의 LPM탐색율과 같게 된다.

시스템의 시간당 처리 고객의 수는 PICAM의 LPM탐색율이 된다. 또한 고객이 단계1에서 서비스를 받기 시작해서 단계3의 서비스를 다 받을 때까지의 시간은 PICAM의 LPM탐색시간이 된다. 시스템의 단계2의 대기행렬들은 매치벡터버퍼, 단계3의 대기행렬은 히트벡터버퍼다. 아래의 각각의 경우는 1,000,000명의 고객에 대해 시뮬레이션을 수행하였다.

4.1 매칭점 가정1인 경우 시뮬레이션 결과

이 경우는 단계1에서 처리가 끝나고 단계2의 각 키필드블록으로 들어가는 확률 p_0, p_1, ..., p_{m-1}은 1/m으로 균일하다. IP version 4와 IP version 6의 경우에 대해 m에 대한 LPM탐색율, LPM탐색시간, 및 최대 대기행렬 길이는 표 1과 표 2와 같다. 여기서 LPM 탐색율은 PICAM의 throughput을 나타내는 지표로서 가능한 최대치는 1이다. 1인 경우에는 파이프라인흐름이 매우 순조로움을 뜻하고, 작아질수록 파이프라인이 순조롭지 않아 어느 한쪽에서 정체되고 있음을 뜻한다. LPM탐색시간은 하나의 LPM탐색요구가 단계1에서 처리되기 시작해서 단계3의 처리가 끝날 때까지 걸린 시간을 말한다. 시간이 길어질수록 시스템 내에서 정체되고 있음을 의미한다. 최대 대기행렬 길이는 단계2 및 단계3의 입력단에 있는 버퍼의 크기를 결정할 수 있는 지표가 된다. 대기행렬의 길이가 이보다 커질 수 없으므로 버퍼의 크기는 최대 대기행렬 길이로 놓을 수 있다.

표 1. IP version 4및 매칭점 가정1인 경우 PICAM의 파이프라인 성능

m		2	4	8
LPM탐색율(1/T)		0.4513	0.8598	0.9998
LPM탐색시간(T)		13.2959	6.9787	6.0010
최대 대기 행렬 길이	단계2	5	5	3
	단계3	1	2	2

T: 파이프라인 클록 1 주기 m: 키필드 분할 개수

표 2. IP version 6 및 매칭점가정 1인 경우 PICAM의 파이프라인 성능

m		4	8	16	32
LPM탐색율(1/T)		0.5611	0.8220	0.9934	1.0000
LPM탐색시간(T)		10.6934	7.2992	6.0396	5.9915
최대 대기 행렬 길이	단계2	5	5	3	2
	단계3	2	2	2	2

T: 파이프라인 클록 1 주기

시뮬레이션 결과는 표1 및 표2와 같다. m이 커질수록 LPM탐색율은 높아지고, LPM탐색시간은 짧아지고 있다. IP version 4인 경우 m이 8, IP version 6인 경우 m이 16이면, LPM탐색율이 충분히 1에 가까워져 있다.

m이 커지면 비용이 증가하므로 가능한 m를 작게 할 필요가 있다. IP version 4인 경우 m는 8, IP version 6인 경우 m이 16이면 LPM 탐색율을 저하시키지 않는 구조로 볼 수 있다.

4.2 매칭점 가정2인 경우의 시뮬레이션 결과

이 경우는 단계1에서 단계2로 넘어갈 때 각 키필드블록으로 들어갈 확률 p_0, p_1, ..., p_{m-1}은 그림 3의 prefix길이분포와 연관되어있다. 즉 키필드블록k로 들어갈 확률 p_k라면, $p_k = \sum_{r=bk+1}^{b(k+1)} N(r) / \sum_{r=1}^{w} N(r)$ 이다. 예를 들면, IP version 4이고 m=8이면, 그림 3에서 키필드블록 3으로 들어갈 확률은 prefix길이가 13에서 16의 범위에 있는 엔트리 수의 합을 전체 엔트리 수의 개수로 나눈 수가 된다.

매칭점 가정2는 IP version 4인 경우에 대해서만 적용하였다. 매칭점 가정2를 IP version 4에 적용하면, 단계2에서 특정 키필드블록에 다른 키필드블록보다 부하가 많이 걸릴 수 있다. 특히 prefix길이가 24비트가 되는 IP주소를 처리하게 되는 키필드블록에 부하가 많이 걸린다. 이런 편중된 부하가 단계2의 병렬성을 감소시키게 된다. 매칭점 가정 2를 적용한 결과는 표3과 같다.

매칭점 가정2를 적용한 경우는 매칭점 가정 1을 적용한 경우보다 LPM탐색율은 저하되고, LPM탐색시간은 증가하고 있다. 이는 편중된 부하로 인한 병렬성 감소로 발생한 것이다. LPM 탐색율에 대한 매칭점 가정 1 및 매칭점 가정 2의 비교는 표 4와 같다.

표 3. IP version 4및 매칭점 가정2인 경우 PICAM의 파이프라인 성능

m		2	4	8
LPM탐색율(1/T)		0.2846	0.3798	0.6699
LPM탐색시간(T)		21.0857	15.7993	8.9565
최대 대기 행렬 길이	단계2	5	5	4
	단계3	1	1	2

T: 파이프라인 클록 1 주기 m: 키필드 분할 개수

표 4. IP version 4에서의 매칭점 가정 1과 매칭점 가정 2 경우의 LPM 탐색율 비교

m	2	4	8
매칭점 가정 1	0.4513	0.8598	0.9998
매칭점 가정 2	0.2846	0.3798	0.6699

부하 편중현상으로 인한 성능저하에 대한 보완책으로 단계2에 부하가 많이 걸리는 키필드블록을 한 개 이상 증설한다. 특정 키필드블록 즉 과부하가 걸리는 키필드블록의 개수를 늘리는데 따른 PICAM의 성능은 표5, 표6, 표7과 같다. 여기서 특정 키필드블록은 과부하가 걸리는 키필드블록을 말한다. 증설 키필드블록 개수가 0인 경우는 증설하지 않고 시뮬레이션한 경우이고, 1인 경우는 특정 키필드블록을 1개 더 증설한 경우이며, 2인 경우는 2개 더 증설한 경우이고, 3인 경우는 3개 더 증설한 경우이다.

표 5. 매칭점 가정2인 경우 증설 키필드블록의 개수에 따른 PICAM의 LPM 탐색율

m		2	4	8
증설 키필드블록 개수	0	0.2846	0.3798	0.6699
	1	0.5689	0.7584	1.0000
	2	0.8286	0.9974	
	3	0.9629		

단위 : 1/T, T는 파이프라인 클록 1 주기

특정 키필드블록을 증설함으로써 LPM탐색율을 올릴 수 있다. 표3에서 보면 m=8이고, 특정 키필드블록을 1개 증설한 경우 LPM탐색율은 1이다. 따라서 과부하가 걸리는 특정키필드블록을 증설함으로써 IP 주소의 편중현상이 발생해도 LPM탐색율의 저하없이 PICAM을 동작시킬 수 있다. m=8인 경우 이로 인한 CAM메모리 셀은 단계2에 필요한 CAM메모리 셀의 1/16에 불과하다. 또한 그림 4에서 보면 엔트리가 없는 prefix길이 영역이 있으므로 이들에 해당하는 키필드블록의 CAM메모리 셀을 특정 키필드블록에 배정함으로써 실질적인 CAM메모리 셀 증가를 없앨 수 있다.

표 6. 매칭점 가정 2인 경우 증설 키필드블록의 개수에 따른 PICAM의 LPM 탐색시간

m		2	4	8
증설 키필드 블록 개수	0	21.0857	15.7993	8.9565
	1	10.5469	7.9116	5.9956
	2	7.2414	6.0155	
	3	6.2313		

단위 : T, T는 파이프라인 클록 1 주기

표6에서 키필드블록을 증설함으로써 LPM 탐색시간이 단축됨을 알 수 있다. m=8인 경우 증설을 하지 않은 경우에는 8.9565로서 표1에서의 6.0010에 비해 LPM탐색시간이 길다. 이는 단계2의 병렬성 감소로 인한 것이며, 키필드블록을 증설해 이를 단축시킬 수 있다. PICAM에서는 하나의 패킷에 대한 LPM탐색시간은 6이지만 단위시간당 LPM 탐색을 하는 패킷의 개수인 LPM탐색율은 1에 접근시킬 수 있다.

표 7. 매칭점 가정2인 경우 증설 키필드블록의 개수에 따른 PICAM의 최대 대기행렬의 길이

m			2	4	8
증설 키필드 블록 개수	0	단계2	5	5	4
		단계3	1	1	2
	1	단계2	4	3	3
		단계3	1	2	2
	2	단계2	4	3	
		단계3	2	2	
	3	단계2	4		
		단계3	1		

표7에서는 키필드블록을 증설시킴에 따라 최대 대기행렬의 길이가 감소하는 것을 볼 수 있다.

4.3 PICAM의 최적구조

PICAM의 최적구조를 구하기 위해서는 비용 및 성능에 대한 모델이 필요하다. 성능모델은 앞에서 시뮬레이션한 결과이다. 비용모델은 VLSI제작시 소요면적의 지표가 되는 복잡도에 따른 비용모델과 키필드블록증설에 따른 비용모델이 있다.

- 복잡도에 따른 비용모델

비용은 복잡도에 비례한다고 볼 수 있다. PICAM의 복잡도를 나타내는 지표로 셀의 개수와 제어회로의 개수가 있다. 셀의 개수는 m에 무관한 반면, 제어회로의 개수는 m에 비례한다. 비용을 C_m라면, $C_m = \alpha m + \beta$이다. 여기서 α, β는 상수이다. 이들 상수를 정확히 구할 수 있으면, 비용 당 성능을 최대로 하는 m를 정확히 구할 수 있으며, PICAM의 최적구조가 결정될 수 있다. 룩업테이블의 엔트리 n이 커지면 제어회로의 비용 αm 이 CAM 셀의 비용 β보다 상대적으로 매우 작아진다. 따라서 m의 증가에 따른 비용 증가는 작아진다. 이 비용모델은 매칭점 가정 1 및 매칭점 가정 2의 모두에 대해 적용될 수 있다.

- 키필드블록 증설에 따른 비용모델

매칭점 가정 2인 경우 키필드블록 증설에 따른 비용은 다음 표 8과 같다.

표 8. 매칭점 가정2인 경우 증설 키필드블록의 개수에 따른 비용

m		2	4	8
증설 키필드블록 개수	0	1	1	1
	1	$1\frac{1}{4}$	$1\frac{1}{8}$	$1\frac{1}{16}$
	2	$1\frac{1}{2}$	$1\frac{1}{4}$	
	3	$1\frac{3}{4}$		

증설하지 않았을 때의 비용은 1임.

- 최적구조

매칭점 가정 1인 경우에는 복잡도에 따른 비용만

고려하면 된다. 룩업테이블의 크기가 클 경우에는 m의 증가에 따른 비용의 증가가 상대적으로 적다. 따라서 최적구조를 구할 때 비용보다는 성능에 비중을 두는 것이 타당하다. 따라서 이 경우의 최적구조는 IP version 4인 경우 키필드분할수 m은 8이고, IP version 6인 경우는 m이 16이다.
매칭점 가정 2인 경우에는 키필드블록 증설에 따른 비용은 복잡도에 따른 비용(m의 증가에 따른 비용)에 비해 상대적으로 매우 크다. 매칭점 가정 2이고 룩업테이블의 크기가 클 때 즉 백본 라우터의 경우에는 증설 키필드블록에 대한 비용만 고려하면 된다. 표 8에서 보면 m = 8인 경우 1/16 비용증가로 LPM 탐색율이 1로 증가한 반면, m = 2 인 경우에는 3/4의 비용증가로도 LPM 탐색율이 1에 미치지 못하고 있다. 따라서 m이 8이고 키필드블록을 하나 증설하는 경우가 비용증가를 최소로 하면서도 LPM탐색율이 저하되지 않는 최적의 PICAM 구조이다. IP version 4 의 최적구조 및 IP version 6의 최적구조 모두 버퍼의 크기는 단계2는 3, 단계3은 2이다.

V. 결 론

PICAM은 고속 LPM 탐색을 위한 파이프라인 CAM구조이다. PICAM은 3단계의 파이프라인으로 구성되고, 단계2 및 단계3의 입력단에는 패킷충돌(packet conflict)의 경우 원활한 동작을 위해 버퍼가 있다. 매칭점 위치의 확률분포는 IP주소의 prefix 발생 분포에 따라 달라진다. 본 논문에서는 PICAM의 파이프라인 성능모델을 제시하고, IP주소의 prefix길이 발생분포를 기반으로 이산사건 시뮬레이션을 수행하였다. IP version 4인 경우 키필드를 8개의 키필드블록으로 분할하고 단계2에서는 입력 IP 패킷 발생이 많은 키필드블록을 중복 설치하는 구조가 최적이고, IP version 6인 경우에는 키필드를 16개의 키필드블록으로 나누어 구성하는 구조가 최적이다. 또한 이때 필요한 단계2 및 단계3의 버퍼의 크기는 3 및 2이다.

참 고 문 헌

[1] Anthony J. McAuley and Paul Francis, "Fast Routing Table Lookup Using CAMs", IEEE INFOCOM'93, vol. 3, pp 1392-1392, March 1993

[2] Nen-Fu Huang and Shi-Ming Zhao, "A Novel IP-Routing Lookup Scheme and Hardware Architecture for Multigigabit Switching Routers", IEEE Journal on Selected Areas in Communications, Vol. 17, No. 6, June 1999, pp. 1093 - 1104

[3] Y. Rekhter and T. Li. "An Architecture for IP Address Allocation with CIDR." RFC 1518, Sept. 1993

[4] Henry Hong-Yi Tzeng and Tony Przygienda, "On Fast Address-Lookup Algorithms", IEEE Journal on Selected Areas in Communications, Vol. 17, No. 6, June 1999, pp. 1067-1082

[5] Pankaj Gupta, Steven Lin, and Nick McKeown, "Routing Lookups in Hardware at Memory Speeds", Proc. of INFOCOM '98, Session 10B-1, San Francisco, CA, pp. 1240-1247

[6] Andreas Moestedt and Peter Sjodin, "IP Address Lookup in Hardware for High-Speed Routing", Proceeding of Hot Interconnects, August 1998, pp. 1-9

[7] Marcel W. Waldvogel, George Varghese, Jon Turner, Bernhard Platner, "Scalable High Speed IP Routing Lookups", Proc. of ACM SIGCOM '97, France, pp. 25-36

[8] Mikael Degermark, Andrew Brodnik, Svante Carlsson, and Stephen Pink, "Small Forwarding Tables for Fast Routing Lookups", Proc. of ACM SIGCOMM'97, France, pp. 3-14

[9] Wen-Shyen E. Chen and Chung-Ting Justine Tsai, "A Fast and Scalable IP Lookup Scheme for High-Speed Networks", Proc. of IEEE International Conference on Networks(ICON '99), pp. 211-218

[10] Stefan Nilsson and Gunnar Karlsson, "Fast address lookup for Internet routers", http://www.nada.kth.se/~snilsson/public/papers.html

[11] Butler Lampson, Venkatachary Srinivasan, and George Varghese, "IP Lookups Using Multiway and Multicolumn Search", IEEE Transaction on Networking, Vol. 7, No. 3, June 1999, pp. 324-334

[12] Anthony J. McAuley, Paul F. Tsuchiya, and Daniel V. Wilson. "Fast multilevel hierarchical routing table using content-addressable

memory", U. S. Patent serial number 034444. Assignee Bell Communications research Inc Livingston NJ, January 1995.

[13] Daxiao Yu, Brandon C. Smith, and Belle Wei, "Forwarding Engine For Fast Routing Lookups and Updates", IEEE Global Telecommunications Conference, proceeding on GLOBECOM '99, vol. 2, pp. 1556-1564, 1999

[14] Scott Bradner, "Next Generation routers Overview," proceeding of Networld Interop 97, 1997

[15] 안희일, 조태원, "고속 LPM 탐색을 위한 파이프라인 CAM 구조(PICAM)", 한국통신학회 논문지 제 26권 제 4호, pp 650-661, 2001

[16] Merit Networks, Inc. See http://www.merit.edu

[17] M. H. MacDougall, "Simulating Computer Systems", The MIT Press, 1987

안 희 일(Hee Il Ahn) 정회원

1973년 2월 : 서울대학교 전자공학과 졸업(공학사)
1996년 2월 : 충북대학교 대학원 전자공학과 졸업 (공학석사)
1976년 11월~1978년 3월 : 한국과학기술연구원 연구원
1978년 1월~1999년 4월 : 한국전자통신연구원 책임연구원/실장
1999년 9월~현재 : 충북대학교 전기전자공학부 객원교수, 강사
<주관심 분야> 컴퓨터구조, 라우터, 유전자알고리즘

조 태 원(Tae Won Cho) 정회원

1973년 2월 : 서울대학교 전자공학과 졸업(공학사)
1986년 5월 : 미국 루이빌대 전자공학과 졸업 (공학석사)
1992년 5월 : 미국 켄터키 주립대 전자공학과 졸업 (공학박사)
1973년 8월~1983년 10월 : 금성전선㈜
1977년 1월~1977년 3월 : 영국 및 프랑스의 ITT제열사 연수
1992년 9월~현재 : 충북대학교 전기전자공학부 부교수
<주관심 분야> 집적회로설계, 컴퓨터구조, 저전력회로설계, DSP core 설계