



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2017년10월19일
(11) 등록번호 10-1787899
(24) 등록일자 2017년10월12일

(51) 국제특허분류(Int. Cl.)
G06F 17/30 (2006.01) H04L 29/12 (2006.01)
(52) CPC특허분류
G06F 17/30324 (2013.01)
G06F 17/30327 (2013.01)
(21) 출원번호 10-2016-0070526
(22) 출원일자 2016년06월07일
심사청구일자 2016년06월07일
(56) 선행기술조사문헌
KR101582050 B1*
KR1020110060428 A*
KR1020140092449 A*
*는 심사관에 의하여 인용된 문헌

(73) 특허권자
이화여자대학교 산학협력단
서울특별시 서대문구 이화여대길 52 (대현동, 이화여자대학교)
(72) 발명자
임혜숙
서울특별시 강남구 남부순환로 2803, 108동 1102호 (도곡동, 삼성래미안아파트)
서지희
서울특별시 노원구 화랑로 564, 106동 801호 (공릉동, 육사아파트)
(74) 대리인
특허법인 무한

전체 청구항 수 : 총 17 항

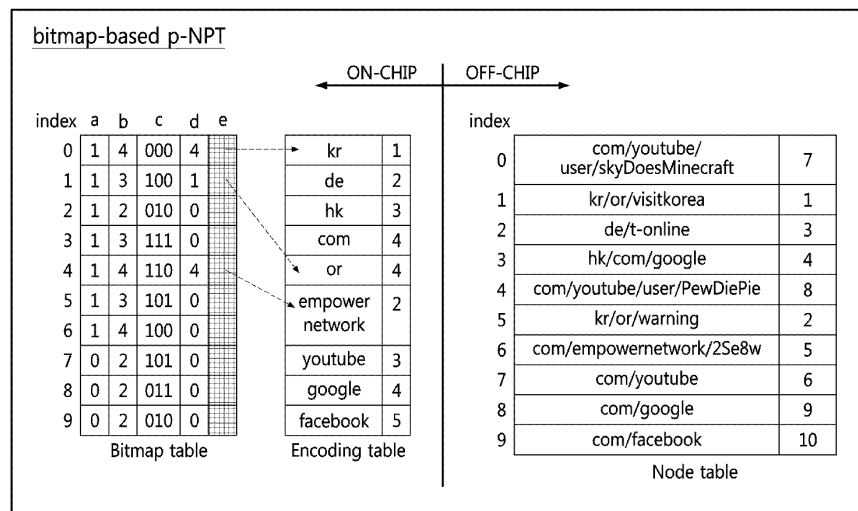
심사관 : 김경완

(54) 발명의 명칭 우선순위 이름 프리픽스 트라이 기반의 비트맵 생성 방법 및 상기 비트맵을 이용한 이름 프리픽스 검색 방법

(57) 요약

우선순위 이름 프리픽스 트라이 기반의 비트맵 생성 방법 및 상기 비트맵을 이용한 이름 프리픽스 검색 방법이 개시된다. 우선순위 이름 프리픽스 트라이 기반의 비트맵 생성 방법은 이름 프리픽스 트라이(Name Prefix Trie, NPT)에서 도출된 우선순위 노드 및 비우선순위 노드를 포함하는 우선순위-이름 프리픽스 트라이(Priority-Name Prefix Trie, P-NPT)를 결정하는 단계; 상기 우선순위-이름 프리픽스 트라이로부터 비트맵 테이블, 인코딩 테이블 및 노드 테이블을 포함하는 비트맵을 생성하는 단계; 및 상기 생성된 비트맵을 저장하는 단계를 포함할 수 있다.

대표도



(52) CPC특허분류

G06F 17/30336 (2013.01)

G06F 17/30339 (2013.01)

G06F 17/30424 (2013.01)

H04L 61/305 (2013.01)

이 발명을 지원한 국가연구개발사업

과제고유번호 2014R1A2A1A11051762

부처명 미래창조과학부

연구관리전문기관 한국연구재단

연구사업명 기초연구사업(학술진흥)-중견연구자지원사업-도약연구지원사업(전략)(상향식)

연구과제명 차세대 콘텐츠 중심 네트워크에서의 패킷 포워딩 기술에 관한 연구

기 여 율 1/2

주관기관 이화여자대학교 산학협력단

연구기간 2014.11.01 ~ 2017.10.31

이 발명을 지원한 국가연구개발사업

과제고유번호 IITP-2016-H8501-16-1007

부처명 미래창조과학부

연구관리전문기관 정보통신기술진흥센터

연구사업명 대학 ICT연구센터 육성지원사업

연구과제명 초고속 무선 네트워크 기반 u- Office 서비스 연구 개발

기 여 율 1/2

주관기관 중앙대학교

연구기간 2016.01.01 ~ 2016.12.31

공지예외적용 : 있음

명세서

청구범위

청구항 1

이름 프리픽스 검색 장치가 수행하는 우선순위 이름 프리픽스 트라이 기반의 비트맵 생성 방법에 있어서,

이름 프리픽스 검색 장치의 프로세서가 이름 프리픽스 트라이(Name Prefix Trie, NPT)에서 도출된 우선순위 노드 및 비우선순위 노드를 포함하는 우선순위-이름 프리픽스 트라이(Priority-Name Prefix Trie, P-NPT)를 결정하는 단계;

이름 프리픽스 검색 장치의 프로세서가 상기 우선순위-이름 프리픽스 트라이로부터 비트맵 테이블, 인코딩 테이블 및 노드 테이블을 포함하는 비트맵을 생성하는 단계; 및

이름 프리픽스 검색 장치의 프로세서가 상기 생성된 비트맵을 저장하는 단계

를 포함하는 우선순위 이름 프리픽스 트라이 기반의 비트맵 생성 방법.

청구항 2

제1항에 있어서,

상기 결정하는 단계는,

이름 프리픽스 검색 장치의 프로세서가 상기 이름 프리픽스 트라이에서 이름 프리픽스를 가지지 않는 빈 노드가 루트 노드로 설정된 서브 트라이를 결정하는 단계; 및

이름 프리픽스 검색 장치의 프로세서가 상기 서브 트라이에서 이름 프리픽스를 가지는 적어도 하나의 삽입 노드들 중 가장 많은 컴포넌트를 갖는 이름 프리픽스가 저장된 삽입 노드를 우선순위 노드로 결정하여 상기 서브 트라이의 루트 노드로 대체하는 단계

를 포함하는 우선순위 이름 프리픽스 트라이 기반의 비트맵 생성 방법.

청구항 3

제1항에 있어서,

상기 생성하는 단계는,

이름 프리픽스 검색 장치의 프로세서가 상기 우선순위-이름 프리픽스 트라이에 포함된 우선순위 노드 및 상기 우선순위 노드를 제외한 삽입 노드를 나타내는 비우선순위 노드의 노드 번호를 상기 비트맵 테이블의 인덱스 번호 및 노드 테이블의 인덱스 번호 각각에 매칭하는 우선순위 이름 프리픽스 트라이 기반의 비트맵 생성 방법.

청구항 4

제1항에 있어서,

상기 비트맵 테이블은,

(i) 상기 우선순위-이름 프리픽스 트라이에 포함된 노드들이 우선순위 노드 또는 비우선순위 노드인지 여부를 나타내는 노드 타입,

(ii) 우선순위-이름 프리픽스 트라이에 포함된 노드들 각각에 저장된 이름 프리픽스를 구성하는 컴포넌트의 개수를 나타내는 컴포넌트,

(iii) 우선순위-이름 프리픽스 트라이에 포함된 노드들 각각에 저장된 이름 프리픽스에 해시 함수를 적용하여 도출된 핑거프린트(fingerprint)를 나타내는 해시브리프,

(iv) 우선순위-이름 프리픽스 트라이에 포함된 노드들에 대한 자식 노드의 개수를 나타내는 차일드 개수 정보 및

(iv) 우선순위-이름 프리픽스 트라이에 포함된 노드들의 첫번째 자식 노드로의 경로에 해당하는 인코딩 테이블의 주소값을 나타내는 차일드 포인터 정보 중 적어도 하나의 정보를 포함하는 우선순위 이름 프리픽스 트라이 기반의 비트맵 생성 방법.

청구항 5

제1항에 있어서,

상기 인코딩 테이블은,

상기 우선순위-이름 프리픽스 트라이에 포함된 우선순위 노드와 비우선순위 노드 각각의 자식 노드에 대한 에지(edge) 정보 및 상기 우선순위 노드 및 비우선순위 노드와 상기 자식 노드 간의 인덱스 번호 차이를 나타내는 인코딩 번호를 포함하는 우선순위 이름 프리픽스 트라이 기반의 비트맵 생성 방법.

청구항 6

제1항에 있어서,

상기 노드 테이블은,

상기 우선순위-이름 프리픽스 트라이의 우선순위 노드와 비우선순위 노드 각각에 포함된 이름 프리픽스와 상기 이름 프리픽스에 대응하는 출력 페이지를 포함하는 우선순위 이름 프리픽스 트라이 기반의 비트맵 생성 방법.

청구항 7

제1항에 있어서,

상기 저장하는 단계는,

이름 프리픽스 검색 장치의 프로세서가 상기 비트맵 테이블 및 인코딩 테이블은 이름 프리픽스를 검색하는 칩의 내부 메모리에 저장하고,

상기 노드 테이블은 이름 프리픽스를 검색하는 칩의 외부 메모리에 저장하는 우선순위 이름 프리픽스 트라이 기반의 비트맵 생성 방법.

청구항 8

이름 프리픽스 검색 장치가 수행하는 이름 프리픽스 검색 방법은,

이름 프리픽스 검색 장치의 프로세서가 이름 프리픽스 트라이에서 도출된 우선순위 노드 및 비우선순위 노드를 포함하는 우선순위-이름 프리픽스 트라이에 기초하여 생성된 비트맵을 식별하는 단계;

이름 프리픽스 검색 장치의 프로세서가 상기 식별된 비트맵을 이용하여 입력 패킷의 이름 프리픽스와 가장 많은 컴포넌트가 일치하는 우선순위-이름 프리픽스 트라이 내의 이름 프리픽스를 검색하는 단계

를 포함하고,

상기 비트맵은,

비트맵 테이블, 인코딩 테이블 및 노드 테이블로 구성되는 이름 프리픽스 검색 방법.

청구항 9

제8항에 있어서,

상기 검색하는 단계는,

이름 프리픽스 검색 장치의 프로세서가 상기 입력 패킷의 이름 프리픽스를 검색하기 위해 접근한 상기 비트맵 테이블의 노드 타입에 기초하여 상기 우선순위-이름 프리픽스 트라이에 포함된 노드가 우선순위-노드인지 또는 비우선순위 노드인지 확인하는 단계; 및

이름 프리픽스 검색 장치의 프로세서가 상기 우선순위-이름 프리픽스 트라이에 포함된 노드가 비우선순위 노드로 확인되면, 상기 확인된 비우선순위 노드의 인덱스 번호와 동일한 노드 테이블의 인덱스 번호에 접근하여 출

력 페이지를 추출하는 단계
를 포함하는 이름 프리픽스 검색 방법.

청구항 10

제9항에 있어서,

상기 추출하는 단계 이후에,

이름 프리픽스 검색 장치의 프로세서가 상기 입력 패킷의 이름 프리픽스를 검색하기 위해 접근한 상기 비트맵 테이블의 비우선순위 노드가 자식 노드를 가지는지 여부를 확인하는 단계;

이름 프리픽스 검색 장치의 프로세서가 상기 비우선순위 노드가 자식 노드를 가지는 것으로 확인된 경우, 상기 비트맵 테이블의 차일드 포인터 정보를 이용하여 인코딩 테이블에 접근하고, 상기 인코딩 테이블을 이용하여 다음에 검색을 진행할 경로와 일치하는 자식 노드에 대한 예지 정보를 확인하는 단계; 및

이름 프리픽스 검색 장치의 프로세서가 상기 일치하는 자식 노드에 대한 예지 정보가 확인된 경우, 상기 입력 패킷의 이름 프리픽스를 검색하기 위해 접근한 상기 비트맵 테이블의 비우선순위 노드에 대한 인덱스 번호와 상기 확인된 예지 정보에 대응하는 인코딩 번호를 이용하여 다음에 검색을 진행할 비트맵 테이블의 인덱스를 결정하는 단계

를 더 포함하는 이름 프리픽스 검색 방법.

청구항 11

제8항에 있어서,

상기 검색하는 단계는,

이름 프리픽스 검색 장치의 프로세서가 상기 입력 패킷의 이름 프리픽스를 검색하기 위해 접근한 상기 비트맵 테이블의 노드 타입에 기초하여 상기 우선순위-이름 프리픽스 트라이에 포함된 노드가 우선순위 노드인지 비우선순위 노드인지 확인하는 단계;

이름 프리픽스 검색 장치의 프로세서가 상기 우선순위-이름 프리픽스 트라이에 포함된 노드가 우선순위 노드로 확인되면, 상기 입력 패킷의 이름 프리픽스에 대한 해시 값과 상기 비트맵 테이블에 포함된 해시브리프가 일치하는지 비교하는 단계;

이름 프리픽스 검색 장치의 프로세서가 상기 입력 패킷의 이름 프리픽스에 대한 해시 값과 상기 비트맵 테이블에 포함된 해시브리프가 일치하는 경우, 상기 입력 패킷의 이름 프리픽스와 상기 노드 테이블에 저장된 이름 프리픽스와 일치하는지 비교하는 단계; 및

이름 프리픽스 검색 장치의 프로세서가 상기 입력 패킷의 이름 프리픽스와 상기 노드 테이블에 저장된 이름 프리픽스와 일치하는 경우, 상기 확인된 우선순위-이름 프리픽스 트라이에 포함된 노드의 인덱스 번호와 동일한 노드 테이블의 인덱스 번호에 접근하여 출력 페이지를 추출하는 단계

를 포함하는 이름 프리픽스 검색 방법.

청구항 12

제8항에 있어서,

상기 우선순위-이름 프리픽스 트라이는,

상기 이름 프리픽스 트라이에서 이름 프리픽스를 가지지 않는 빈 노드를 루트 노드로 설정하는 서브 트라이를 결정하고, 상기 서브 트라이에서 이름 프리픽스를 가지는 적어도 하나의 삽입 노드들 중 가장 많은 컴포넌트를 갖는 이름 프리픽스가 저장된 삽입 노드를 우선순위 노드로 결정하여 상기 서브 트라이의 루트 노드로 대체함으로써 생성되는 이름 프리픽스 검색 방법.

청구항 13

제8항에 있어서,

상기 비트맵은,

상기 우선순위-이름 프리픽스 트라이에 포함된 우선순위 노드 및 상기 우선순위 노드를 제외한 삽입 노드를 나타내는 비우선순위 노드의 노드 번호를 상기 비트맵 테이블의 인덱스 번호 및 인코딩 테이블의 인덱스 번호 각각에 매칭함으로써 생성되는 이름 프리픽스 검색 방법.

청구항 14

제8항에 있어서,

비트맵 테이블은,

- (i) 상기 우선순위-이름 프리픽스 트라이에 포함된 노드들이 우선순위 노드 또는 비우선순위 노드인지 여부를 나타내는 노드 타입,
- (ii) 우선순위-이름 프리픽스 트라이에 포함된 노드들 각각에 저장된 이름 프리픽스를 구성하는 컴포넌트의 개수를 나타내는 컴포넌트,
- (iii) 우선순위-이름 프리픽스 트라이에 포함된 노드들 각각에 저장된 이름 프리픽스에 해시 함수를 적용하여 도출된 핑거프린트(fingerprint)를 나타내는 해시브리프,
- (iv) 우선순위-이름 프리픽스 트라이에 포함된 노드들에 대한 자식 노드의 개수를 나타내는 차일드 개수 정보 및
- (iv) 우선순위-이름 프리픽스 트라이에 포함된 노드들의 첫번째 자식 노드로의 경로에 해당하는 인코딩 테이블의 주소값을 나타내는 차일드 포인터 정보 중 적어도 하나의 정보를 포함하는 이름 프리픽스 검색 방법.

청구항 15

제8항에 있어서,

상기 인코딩 테이블은,

상기 우선순위-이름 프리픽스 트라이에 포함된 우선순위 노드와 비우선순위 노드 각각의 자식 노드에 대한 에지(edge) 정보 및 상기 우선순위 노드 및 비우선순위 노드와 상기 자식 노드 간의 인덱스 번호 차이를 나타내는 인코딩 번호를 포함하는 이름 프리픽스 검색 방법.

청구항 16

제8항에 있어서,

상기 노드 테이블은,

상기 우선순위-이름 프리픽스 트라이의 우선순위 노드와 비우선순위 노드 각각에 포함된 이름 프리픽스와 상기 이름 프리픽스에 대응하는 출력 페이지를 포함하는 이름 프리픽스 검색 방법.

청구항 17

제8항에 있어서,

상기 비트맵 테이블 및 인코딩 테이블은 이름 프리픽스를 검색하는 칩의 내부 메모리에 저장되고,

상기 노드 테이블은 이름 프리픽스를 검색하는 칩의 외부 메모리에 저장되는 이름 프리픽스 검색 방법.

발명의 설명

기술 분야

본 발명은 우선순위 이름 프리픽스 트라이 기반의 비트맵 생성 방법 및 상기 비트맵을 이용한 이름 프리픽스 검색 방법에 관한 것으로, 보다 구체적으로는 이름 프리픽스 트라이(Name Prefix Trie, NPT)에서 도출된 우선순위 노드 및 비우선순위 노드를 포함하는 우선순위-이름 프리픽스 트라이(Priority-Name Prefix Trie, P-NPT)를 결

[0001]

정하고, 결정된 우선순위-이름 프리픽스 트라이로부터 생성된 비트맵(비트맵 테이블, 인코딩 테이블 및 노드 테이블)을 이용하여 입력 패킷에 대한 이름 프리픽스를 검색하는 방법에 관한 것이다.

배경 기술

[0002] 현재의 인터넷은 TCP/IP를 기반으로 하는 자원(resource) 공유를 목적으로 개발되었다. 따라서, 이러한 TCP/IP 기반의 인터넷은 두 종단 시스템(end-system)간의 데이터 전송에 적합하다. 하지만, 오늘날 인터넷 트래픽은 단순한 데이터 전송이 아닌 웹 스트리밍, 비디오 전송, P2P(peer-to-peer)기반의 파일 공유 등 대용량 콘텐츠의 분배 및 응용이 주를 이룬다. 이러한 인터넷 트래픽 양상의 변화로 인하여 기존의 인터넷은 대역폭(bandwidth) 부족, 서비스 품질의 저하 등의 한계점을 드러내고 있다. 이와 같은 인터넷의 문제를 해결하기 위해 미래인터넷의 한 형태로 제안된 것이 NDN(Named Data Network)이다.

[0003] 기존의 인터넷 라우터와 NDN 라우터에서는 가장 길게 일치하는 프리픽스(longest prefix)를 기반으로 패킷 전송이 이루어진다. 그러나 이와 같이 가장 길게 일치하는 프리픽스를 찾는 과정에서 기존의 인터넷 라우터는 IP주소를 사용하지만, NDN 라우터는 콘텐츠의 이름을 사용한다는 점에서 차이가 있다. 콘텐츠의 이름은 고정된 길이를 갖는 IP주소와는 달리 길이와 형태의 제약이 없기 때문에, NDN 라우터의 FIB(Forwarding Information Base)테이블을 구현하기 위해서는 많은 메모리가 필요하다. 이러한 이유로, NDN 라우터의 FIB테이블은 칩-외부메모리(off-chip memory)를 이용하여야만 구현이 가능하다.

[0004] 하지만, DRAM과 같은 칩-외부메모리를 사용하여 프로세싱할 때 걸리는 시간은 SRAM과 같은 칩-내부메모리를 사용하여 프로세싱할 때 걸리는 시간의 약 10~20배에 달한다. 칩-외부메모리에 저장된 FIB테이블을 사용하여 가장 길게 일치하는 프리픽스를 찾는 과정은 전체 패킷 전송 속도를 저하시키는 큰 요인이 되고 있다. 라우터에서는 패킷 선-속도(wire-speed)에 맞추어 패킷 전송이 이루어져야 하기 때문에, FIB테이블에서의 검색 속도 저하는 라우터의 성능저하로 직결된다.

발명의 내용

해결하려는 과제

[0005] 본 발명은 이름 프리픽스 트라이(Name Prefix Trie, NPT)에서 도출된 우선순위 노드 및 비우선순위 노드를 포함하는 우선순위-이름 프리픽스 트라이(Priority-Name Prefix Trie, P-NPT)를 결정하고, 결정된 우선순위-이름 프리픽스 트라이로부터 생성된 비트맵(비트맵 테이블, 인코딩 테이블 및 노드 테이블)을 이용하여 입력 패킷에 대한 이름 프리픽스를 검색함으로써 이름 프리픽스 검색 능력을 향상시킬 수 있는 장치 및 방법을 제공한다.

과제의 해결 수단

[0006] 본 발명의 일실시예에 따른 우선순위 이름 프리픽스 트라이 기반의 비트맵 생성 방법은 이름 프리픽스 트라이(Name Prefix Trie, NPT)에서 도출된 우선순위 노드 및 비우선순위 노드를 포함하는 우선순위-이름 프리픽스 트라이(Priority-Name Prefix Trie, P-NPT)를 결정하는 단계; 상기 우선순위-이름 프리픽스 트라이로부터 비트맵 테이블, 인코딩 테이블 및 노드 테이블을 포함하는 비트맵을 생성하는 단계; 및 상기 생성된 비트맵을 저장하는 단계를 포함할 수 있다.

[0007] 상기 결정하는 단계는 상기 이름 프리픽스 트라이에서 이름 프리픽스를 가지지 않는 빈 노드가 루트 노드로 설정된 서브 트라이를 결정하는 단계; 상기 서브 트라이에서 이름 프리픽스를 가지는 적어도 하나의 삽입 노드들 중 가장 많은 컴포넌트를 갖는 이름 프리픽스가 저장된 삽입 노드를 우선순위 노드로 결정하여 상기 서브 트라이의 루트 노드로 대체하는 단계를 포함할 수 있다.

[0008] 상기 생성하는 단계는 상기 우선순위-이름 프리픽스 트라이에 포함된 우선순위 노드 및 상기 우선순위 노드를 제외한 삽입 노드를 나타내는 비우선순위 노드의 노드 번호를 상기 비트맵 테이블의 인덱스 번호 및 노드 테이블의 인덱스 번호 각각에 매칭할 수 있다.

[0009] 상기 비트맵 테이블은 (i) 상기 우선순위-이름 프리픽스 트라이에 포함된 노드들이 우선순위 노드 또는 비우선순위 노드인지 여부를 나타내는 노드 타입, (ii) 우선순위-이름 프리픽스 트라이에 포함된 노드들 각각에 저장된 이름 프리픽스를 구성하는 컴포넌트의 개수를 나타내는 컴포넌트, (iii) 우선순위-이름 프리픽스 트라이에 포함된 노드들 각각에 저장된 이름 프리픽스에 해시 함수를 적용하여 도출된 핑거프린트(fingerprint)를 나타내는 해시브리프, (iv) 우선순위-이름 프리픽스 트라이에 포함된 노드들에 대한 자식 노드의 개수를 나타내는 차

일드 개수 정보 및 (iv) 우선순위-이름 프리픽스 트라이에 포함된 노드들의 첫번째 자식 노드로의 경로에 해당하는 인코딩 테이블의 주소값을 나타내는 차일드 포인터 정보 중 적어도 하나의 정보를 포함할 수 있다.

- [0010] 상기 인코딩 테이블은 상기 우선순위-이름 프리픽스 트라이에 포함된 우선순위 노드와 비우선순위 노드 각각의 자식 노드에 대한 에지(edge) 정보 및 상기 우선순위 노드 및 비우선순위 노드와 상기 자식 노드 간의 인덱스 번호 차이를 나타내는 인코딩 번호를 포함할 수 있다.
- [0011] 상기 노드 테이블은 상기 우선순위-이름 프리픽스 트라이의 우선순위 노드와 비우선순위 노드 각각에 포함된 이름 프리픽스와 상기 이름 프리픽스에 대응하는 출력 페이지를 포함할 수 있다.
- [0012] 상기 저장하는 단계는 상기 비트맵 테이블 및 인코딩 테이블은 이름 프리픽스를 검색하는 칩의 내부 메모리에 저장하고, 상기 노드 테이블은 이름 프리픽스를 검색하는 칩의 외부 메모리에 저장할 수 있다.
- [0013] 본 발명의 일실시예에 따른 이름 프리픽스 검색 방법은 이름 프리픽스 트라이에서 도출된 우선순위 노드 및 비우선순위 노드를 포함하는 우선순위-이름 프리픽스 트라이에 기초하여 생성된 비트맵을 식별하는 단계; 및 상기 식별된 비트맵을 이용하여 입력 패킷의 이름 프리픽스와 가장 많은 컴포넌트가 일치하는 우선순위-이름 프리픽스 트라이 내의 이름 프리픽스를 검색하는 단계;를 포함하고, 상기 비트맵은 비트맵 테이블, 인코딩 테이블 및 노드 테이블로 구성될 수 있다.
- [0014] 상기 검색하는 단계는 상기 입력 패킷의 이름 프리픽스를 검색하기 위해 접근한 상기 비트맵 테이블의 노드 타입에 기초하여 상기 우선순위-이름 프리픽스 트라이에 포함된 노드가 우선순위-노드인지 또는 비우선순위 노드인지 확인하는 단계; 및 상기 우선순위-이름 프리픽스 트라이에 포함된 노드가 비우선순위 노드로 확인되면, 상기 확인된 비우선순위 노드의 인덱스 번호와 동일한 노드 테이블의 인덱스 번호에 접근하여 출력 페이지를 추출하는 단계를 포함할 수 있다.
- [0015] 상기 추출하는 단계 이후에 상기 입력 패킷의 이름 프리픽스를 검색하기 위해 접근한 상기 비트맵 테이블의 비우선순위 노드가 자식 노드를 가지는지 여부를 확인하는 단계; 상기 비우선순위 노드가 자식 노드를 가지는 것으로 확인된 경우, 상기 비트맵 테이블의 차일드 포인터 정보를 이용하여 인코딩 테이블에 접근하고, 상기 인코딩 테이블을 이용하여 다음에 검색을 진행할 경로와 일치하는 자식 노드에 대한 에지 정보를 확인하는 단계; 상기 일치하는 자식 노드에 대한 에지 정보가 확인된 경우, 상기 입력 패킷의 이름 프리픽스를 검색하기 위해 접근한 상기 비트맵 테이블의 비우선순위 노드에 대한 인덱스 번호와 상기 확인된 에지 정보에 대응하는 인코딩 번호를 이용하여 다음에 검색을 진행할 비트맵 테이블의 인덱스를 결정하는 단계를 더 포함할 수 있다.
- [0016] 상기 검색하는 단계는 상기 입력 패킷의 이름 프리픽스를 검색하기 위해 접근한 상기 비트맵 테이블의 노드 타입에 기초하여 상기 우선순위-이름 프리픽스 트라이에 포함된 노드가 우선순위 노드인지 비우선순위 노드인지 확인하는 단계; 상기 우선순위-이름 프리픽스 트라이에 포함된 노드가 우선순위 노드로 확인되면, 상기 입력 패킷의 이름 프리픽스에 대한 해시 값과 상기 비트맵 테이블에 포함된 해시브리프가 일치하는지 비교하는 단계; 상기 입력 패킷의 이름 프리픽스에 대한 해시 값과 상기 비트맵 테이블에 포함된 해시브리프가 일치하는 경우, 비트맵 테이블의 인덱스 번호와 동일한 노드 테이블의 인덱스 번호에 접근하여 상기 입력 패킷의 이름 프리픽스와 상기 노드 테이블에 저장된 이름 프리픽스와 일치하는지 비교하는 단계; 및 상기 입력 패킷의 이름 프리픽스와 상기 노드 테이블에 저장된 이름 프리픽스와 일치하는 경우, 출력 페이지를 추출하는 단계를 포함할 수 있다.
- [0017] 상기 우선순위-이름 프리픽스 트라이는 상기 이름 프리픽스 트라이에서 이름 프리픽스를 가지지 않는 빈 노드를 루트 노드로 설정하는 서브 트라이를 결정하고, 상기 서브 트라이에서 이름 프리픽스를 가지는 적어도 하나의 삽입 노드들 중 가장 많은 컴포넌트를 갖는 이름 프리픽스가 저장된 삽입 노드를 우선순위 노드로 결정하여 상기 서브 트라이의 루트 노드로 대체함으로써 생성될 수 있다.
- [0018] 상기 비트맵은 상기 우선순위-이름 프리픽스 트라이에 포함된 우선순위 노드 및 상기 우선순위 노드를 제외한 삽입 노드를 나타내는 비우선순위 노드의 노드 번호를 상기 비트맵 테이블의 인덱스 번호 및 노드 테이블의 인덱스 번호 각각에 매칭함으로써 생성될 수 있다.
- [0019] 비트맵 테이블은 (i) 상기 우선순위-이름 프리픽스 트라이에 포함된 노드들이 우선순위 노드 또는 비우선순위 노드인지 여부를 나타내는 노드 타입, (ii) 우선순위-이름 프리픽스 트라이에 포함된 노드들 각각에 저장된 이름 프리픽스를 구성하는 컴포넌트의 개수를 나타내는 컴포넌트, (iii) 우선순위-이름 프리픽스 트라이에 포함된 노드들 각각에 저장된 이름 프리픽스에 해시 함수를 적용하여 도출된 핑거프린트(fingerprint)를 나타내는 해시 브리프, (iv) 우선순위-이름 프리픽스 트라이에 포함된 노드들에 대한 자식 노드의 개수를 나타내는 차일드 개

수 정보 및 (iv) 우선순위-이름 프리픽스 트라이에 포함된 노드들의 첫번째 자식 노드로의 경로에 해당하는 인코딩 테이블의 주소값을 나타내는 차일드 포인터 정보 중 적어도 하나의 정보를 포함할 수 있다.

[0020] 상기 인코딩 테이블은 상기 우선순위-이름 프리픽스 트라이에 포함된 우선순위 노드와 비우선순위 노드 각각의 자식 노드에 대한 에지(edge) 정보 및 상기 우선순위 노드 및 비우선순위 노드와 상기 자식 노드 간의 인덱스 번호 차이를 나타내는 인코딩 번호를 포함할 수 있다.

[0021] 상기 노드 테이블은 상기 우선순위-이름 프리픽스 트라이의 우선순위 노드와 비우선순위 노드 각각에 포함된 이름 프리픽스와 상기 이름 프리픽스에 대응하는 출력 페이지를 포함할 수 있다.

[0022] 상기 비트맵 테이블 및 인코딩 테이블은 이름 프리픽스를 검색하는 칩의 내부 메모리에 저장되고, 상기 노드 테이블은 이름 프리픽스를 검색하는 칩의 외부 메모리에 저장될 수 있다.

발명의 효과

[0023] 본 발명의 일실시예에 의하면, 이름 프리픽스 트라이(Name Prefix Trie, NPT)에서 도출된 우선순위 노드 및 비우선순위 노드를 포함하는 우선순위-이름 프리픽스 트라이(Priority-Name Prefix Trie, P-NPT)를 결정하고, 결정된 우선순위-이름 프리픽스 트라이로부터 생성된 비트맵(비트맵 테이블, 인코딩 테이블 및 노드 테이블)을 이용하여 입력 패킷에 대한 이름 프리픽스를 검색함으로써 이름 프리픽스 검색 능력을 향상시킬 수 있다.

[0024] 본 발명의 일실시예에 의하면, 우선순위-이름 프리픽스 트라이로부터 생성된 비트맵(비트맵 테이블, 인코딩 테이블 및 노드 테이블)을 이름 프리픽스를 검색하는 칩의 내부 메모리 및 외부 메모리에 나누어 저장함으로써 이름 프리픽스 검색 능력을 향상시킬 수 있다.

도면의 간단한 설명

[0025] 도 1은 본 발명의 일실시예에 따른 이름 프리픽스 검색 장치를 도시한 도면이다.

도 2는 본 발명의 일실시예에 따른 이름 프리픽스의 예시 세트 및 상기 예시 세트를 이용하여 구축된 이름 프리픽스 트라이의 예를 도시한 도면이다.

도 3은 본 발명의 일실시예에 따른 우선순위 이름 프리픽스 트라이의 예를 도시한 도면이다.

도 4는 본 발명의 일실시예에 따른 우선순위 이름 프리픽스 트라이 기반의 비트맵 예를 도시한 도면이다.

도 5는 본 발명의 일실시예에 따른 비우선순위로 판단된 우선순위 이름 프리픽스 트라이의 노드에 대한 이름 프리픽스 검색 방법을 순서대로 도시한 도면이다.

도 6은 본 발명의 일실시예에 따른 우선순위로 판단된 우선순위 이름 프리픽스 트라이의 노드에 대한 이름 프리픽스 검색 방법을 순서대로 도시한 도면이다.

발명을 실시하기 위한 구체적인 내용

[0026] 본 명세서에 개시되어 있는 본 발명의 개념에 따른 실시예들에 대해서 특정한 구조적 또는 기능적 설명들은 단지 본 발명의 개념에 따른 실시예들을 설명하기 위한 목적으로 예시된 것으로서, 본 발명의 개념에 따른 실시예들은 다양한 형태로 실시될 수 있으며 본 명세서에 설명된 실시예들에 한정되지 않는다.

[0027] 본 발명의 개념에 따른 실시예들은 다양한 변경들을 가할 수 있고 여러 가지 형태들을 가질 수 있으므로 실시예들을 도면에 예시하고 본 명세서에 상세하게 설명하고자 한다. 그러나, 이는 본 발명의 개념에 따른 실시예들을 특정한 개시형태들에 대해 한정하려는 것이 아니며, 본 발명의 사상 및 기술 범위에 포함되는 변경, 균등물, 또는 대체물을 포함한다.

[0028] 제1 또는 제2 등의 용어를 다양한 구성요소들을 설명하는데 사용될 수 있지만, 상기 구성요소들은 상기 용어들에 의해 한정되어서는 안 된다. 상기 용어들은 하나의 구성요소를 다른 구성요소로부터 구별하는 목적으로만, 예를 들어 본 발명의 개념에 따른 권리 범위로부터 이탈되지 않은 채, 제1 구성요소는 제2 구성요소로 명명될 수 있고, 유사하게 제2 구성요소는 제1 구성요소로도 명명될 수 있다.

[0029] 어떤 구성요소가 다른 구성요소에 "연결되어" 있다거나 "접속되어" 있다고 언급된 때에는, 그 다른 구성요소에 직접적으로 연결되어 있거나 또는 접속되어 있을 수도 있지만, 중간에 다른 구성요소가 존재할 수도 있다고 이해되어야 할 것이다. 반면에, 어떤 구성요소가 다른 구성요소에 "직접 연결되어" 있다거나 "직접 접속되어" 있

다고 언급된 때에는, 중간에 다른 구성요소가 존재하지 않는 것으로 이해되어야 할 것이다. 구성요소들 간의 관계를 설명하는 표현들, 예를 들어 "~사이에"와 "바로~사이에" 또는 "~에 직접 이웃하는" 등도 마찬가지로 해석되어야 한다.

- [0030] 본 명세서에서 사용한 용어는 단지 특정한 실시예들을 설명하기 위해 사용된 것으로, 본 발명을 한정하려는 의도가 아니다. 단수의 표현은 문맥상 명백하게 다르게 뜻하지 않는 한, 복수의 표현을 포함한다. 본 명세서에서, "포함하다" 또는 "가지다" 등의 용어는 실시된 특징, 숫자, 단계, 동작, 구성요소, 부분품 또는 이들을 조합한 것이 존재함으로 지정하려는 것이지, 하나 또는 그 이상의 다른 특징들이나 숫자, 단계, 동작, 구성요소, 부분품 또는 이들을 조합한 것들의 존재 또는 부가 가능성을 미리 배제하지 않는 것으로 이해되어야 한다.
- [0031] 다르게 정의되지 않는 한, 기술적이거나 과학적인 용어를 포함해서 여기서 사용되는 모든 용어들은 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자에 의해 일반적으로 이해되는 것과 동일한 의미를 가진다. 일반적으로 사용되는 사전에 정의되어 있는 것과 같은 용어들은 관련 기술의 문맥상 가지는 의미와 일치하는 의미를 갖는 것으로 해석되어야 하며, 본 명세서에서 명백하게 정의하지 않는 한, 이상적이거나 과도하게 형식적인 의미로 해석되지 않는다.
- [0032] 이하, 실시예들을 첨부된 도면을 참조하여 상세하게 설명한다. 그러나, 특허출원의 범위가 이러한 실시예들에 의해 제한되거나 한정되는 것은 아니다. 각 도면에 제시된 동일한 참조 부호는 동일한 부재를 나타낸다.
- [0033] 도 1은 본 발명의 일실시예에 따른 이름 프리픽스 검색 장치를 도시한 도면이다.
- [0034] 이름 프리픽스 검색 장치(100)는 프로세서(110), 프로세서의 내부에 배치된 내부 메모리(120) 및 프로세서의 외부에 배치된 외부 메모리(130)로 구성될 수 있다. 이때, 프로세서(110)는 하나의 칩으로 구현될 수 있다. 그래서, 프로세서(110)가 입력 패킷에 대한 이름 프리픽스를 검색하기 위해 프로세서(110)의 내부에 배치된 내부 메모리(120)를 이용하는 경우, 이름 프리픽스 검색은 온-칩(On-Chip)에서 진행될 수 있다. 하지만, 프로세서(110)가 이름 프리픽스를 검색하기 위해 프로세서(110)의 외부에 배치된 외부 메모리(130)를 이용하는 경우, 이름 프리픽스 검색은 오프-칩(Off-Chip)에서 진행될 수 있다.
- [0035] 이 때, 오프-칩에서 진행되는 이름 프리픽스 검색은 외부 메모리(130)로의 접근 시간으로 인해 온-칩에서 진행되는 이름 프리픽스 검색보다 검색 속도가 늦다. 즉, 오프-칩에서 이름 프리픽스를 검색하는 것은 소요되는 시간이 매우 길어 이름 프리픽스를 검색하는 속도에 가장 큰 영향을 미치므로 오프-칩의 접근 횟수를 이름 프리픽스 검색에 대한 성능평가의 지표로 삼을 수 있다.
- [0036] 따라서, 이름 프리픽스 검색 장치(100)는 입력 패킷에 대한 이름 프리픽스 검색에 걸리는 시간을 줄이기 위하여 FIB 테이블을 프로세서(110)의 내부 메모리(120) 및 외부 메모리(130)에 분할하여 저장할 수 있다.
- [0037] 구체적으로 이름 프리픽스 검색 장치(100)는 이름 프리픽스 트라이에서 도출된 우선순위 노드 및 비우선순위 노드를 포함하는 우선순위-이름 프리픽스 트라이에 기초하여 생성된 비트맵을 식별하고, 식별된 비트맵에 기초하여 입력 패킷에 대한 이름 프리픽스를 검색할 수 있다. 즉, 이름 프리픽스 검색 장치(100)는 비트맵 중 우선순위-이름 프리픽스 트라이의 윤곽 정보를 포함하는 비트맵 테이블과 인코딩 테이블을 내부 메모리(120)에 저장하고, 우선순위-이름 프리픽스 트라이에 포함된 각각의 노드에 대한 실질적인 정보인 이름 프리픽스와 출력 페이지스를 외부 메모리(130)에 저장함으로써 외부 메모리(130)에서의 프로세싱을 최소화할 수 있다. 이때, 우선순위-이름 프리픽스 트라이를 결정하는 방법과 비트맵을 생성하는 방법은 추후 도면을 통해 자세히 설명하도록 한다.
- [0038] 도 2는 본 발명의 일실시예에 따른 이름 프리픽스의 예시 세트 및 상기 예시 세트를 이용하여 구축된 이름 프리픽스 트라이의 예를 도시한 도면이다.
- [0039] 이름 프리픽스 트라이(Name Prefix Trie, NPT)는 IP주소 검색에서 이용되었던 이진 트라이(Binary trie)를 확장한 개념이다. 프리픽스가 저장되는 노드(node)까지의 경로가 프리픽스를 의미한다는 점에서는 이진트라이와 같다 그러나 경로 값으로 0과 1만을 사용하였던 IP주소와는 달리, NDN(Named Data Network)에서 경로 값으로 사용하는 콘텐츠의 이름은 길이와 형태에 제한이 없기 때문에 하나의 노드가 가질 수 있는 자식 노드의 개수는 무한히 많아질 수 있다. 이때, 이름 프리픽스 트라이는 IP주소 검색에서 사용되었던 이진 트라이를 확장한 개념이기 때문에, 이진 트라이와 마찬가지로 불필요한 빈 노드(internal node)로 인한 메모리 낭비가 많다. 또한, 입력 패킷에 대한 이름 프리픽스와 가장 많은 컴포넌트가 일치하는 이름 프리픽스를 찾기 위해 검색을 진행할 경로가 더 이상 존재하지 않을 때까지 계속 진행해야 한다는 점에서 검색능력이 좋지 않은 문제가 있다.
- [0040] 이러한 문제에 대한 방안으로 본 발명에서는 이름 프리픽스 트라이에서 도출된 우선순위 노드 및 비우선순위 노

드를 포함하는 우선순위-이름 프리픽스 트라이(Priority-Name Prefix Trie, P-NPT)를 결정하는 방법을 제공한다.

- [0041] 도 2의 (a)는 이름 프리픽스 및 출력 페이스의 예시 세트이며, 도 2의 (b)는 도 2의 (a)에 대한 이름 프리픽스 트라이를 구축 예를 보여준다. 이하에서는 이름 프리픽스 트라이에서 이름 프리픽스가 저장된 노드를 삽입 노드, 이름 프리픽스가 저장된 노드로 가는 경로에 부득이하게 존재하는 비어 있는 노드를 빈 노드로 부르기로 한다. 이때, 각각의 이름 프리픽스는 출력페이스 정보를 가지며, 이름 프리픽스 트라이의 삽입 노드 역시 서로 다른 노드 번호가 설정될 수 있다.
- [0042] 예를 들어, 이름 프리픽스 kr/or/visitkorea를 이름 프리픽스 트라이로 구축하면 다음과 같다. 먼저 이름 프리픽스 트라이의 최상위 레벨에 빈 노드(210)이 위치한다. 이후 빈 노드(210)로부터 kr에 해당하는 패스를 따라 이동하면 빈 노드(220)이 위치한다. 다시 빈 노드(220)으로부터 or에 해당하는 패스를 따라 이동하면 빈 노드(230)이 위치하고, 다시 빈 노드(230)으로부터 visitkorea에 해당하는 패스를 따라 이동하면 이름 프리픽스 kr/or/visitkorea가 저장된 삽입 노드(240)가 위치할 수 있다.
- [0043] 도 3은 본 발명의 일실시예에 따른 우선순위 이름 프리픽스 트라이의 예를 도시한 도면이다.
- [0044] 삽입 노드와 빈 노드만을 가졌던 이름 프리픽스 트라이와는 다르게, 이름 프리픽스 트라이로부터 결정된 우선순위-이름 프리픽스 트라이는 우선순위 노드(priority node)와 우선순위 노드를 제외한 삽입 노드를 나타내는 비 우선순위 노드(non-priority node)만을 갖는다.
- [0045] 우선순위-이름 프리픽스 트라이를 결정하는 과정은 다음과 같다. 먼저 이름 프리픽스 트라이에서 이름 프리픽스를 가지지 않는 빈 노드가 루트 노드로 설정된 서브 트라이를 결정할 수 있다. 이때 선택되는 빈 노드는 이름 프리픽스 트라이의 최상위 레벨에 존재하는 노드일 수 있다. 이후, 결정된 서브 트라이에서 이름 프리픽스가 저장된 적어도 하나의 삽입 노드들 중 가장 많은 컴포넌트를 갖는 이름 프리픽스가 저장된 삽입 노드를 우선순위 노드로 결정하여 서브 트라이의 루트 노드로 대체할 수 있다. 이와 같은 우선순위-이름 프리픽스 트라이를 결정하는 과정은 이름 프리픽스 검색 장치(100)에 포함된 프로세서(110)에 의해 수행될 수 있다.
- [0046] 예를 들어, 도 2의 (b)와 같은 이름 프리픽스 트라이를 우선순위-이름 프리픽스 트라이로 결정하는 과정은 다음과 같다. 먼저 프로세서(110)는 도 3의 (a)와 같은 이름 프리픽스 트라이의 최상위 레벨에 존재하는 빈 노드(310)를 루트 노드로 설정하는 서브 트라이(311)를 결정할 수 있다. 이후, 프로세서(110)는 서브 트라이(311)에 존재하는 삽입 노드들 중 가장 많은 컴포넌트를 갖는 이름 프리픽스가 저장된 삽입 노드(312)를 우선순위 노드로 결정할 수 있다. 이때, 우선순위 노드로 결정된 삽입 노드(312)에 저장된 이름 프리픽스는 'com/youtube/user/skyDoesMinecraft'이고 출력 페이스(output face)는 7이다. 프로세서(110)는 이와 같은 이름 프리픽스와 출력 페이스를 서브 트라이(310)의 루트 노드(310)에 대체하여 저장할 수 있다. 이때, 동일한 서브 트라이(311) 내에 가장 많은 컴포넌트를 갖는 이름 프리픽스가 저장된 삽입 노드(312,313)가 둘 이상일 경우 프로세서(110)는 둘 중 하나를 선택하여 루트 노드로 대체할 수 있다. 본 발명에서는 가장 많은 컴포넌트를 갖는 이름 프리픽스가 저장된 삽입 노드(312,313)가 둘 이상일 경우 왼쪽에 위치하는 삽입 노드(312)를 우선순위 노드로 결정하여 루트 노드로 대체하였다.
- [0047] 이제 이름 프리픽스 트라이의 최상위 레벨에는 빈 노드가 더 이상 존재하지 않으므로 다음 하위 레벨로 넘어갈 수 있다. 따라서, 프로세서(110)는 이름 프리픽스 트라이의 두 번째 레벨에 존재하는 빈 노드(320)가 루트 노드로 설정된 서브 트라이(321)를 결정할 수 있다. 이후, 프로세서(110)는 서브 트라이(321)에 존재하는 삽입 노드들 중 가장 많은 컴포넌트를 갖는 이름 프리픽스가 저장된 삽입 노드(322)를 우선순위 노드로 결정할 수 있다. 이때, 우선순위 노드로 결정된 삽입 노드(322)에 저장된 이름 프리픽스는 'kr/or/visitkorea'이고, 출력 페이스는 1이다. 프로세서(110)는 이와 같은 이름 프리픽스와 출력 페이스를 서브 트라이(321)의 루트 노드(320)에 대체하여 저장할 수 있다. 마찬가지로 동일한 서브 트라이(321) 내에 가장 많은 컴포넌트를 갖는 이름 프리픽스가 저장된 삽입 노드(322,323)가 둘 이상일 경우 프로세서(110)는 둘 중 하나를 선택하여 루트 노드로 대체할 수 있다.
- [0048] 프로세서(110)는 이름 프리픽스 트라이 내에 더 이상 빈 노드가 존재하지 않을 때까지 이와 같은 과정을 반복하여 수행함으로써 도 3의 (b)와 같은 우선순위-이름 프리픽스 트라이를 구축할 수 있다. 이와 같이 우선순위-이름 프리픽스 트라이는 이름 프리픽스를 검색하기 위하여 우선순위 노드에 접근하였을 때 가장 많은 컴포넌트가 일치할 가능성이 있는 삽입 노드의 정보를 저장한다. 따라서, 입력 패킷에 대한 이름 프리픽스와 우선순위 노드에 저장된 이름 프리픽스 정보가 일치할 경우, 프로세서(110)는 검색을 진행할 경로가 남아 있더라도 검색을 종

료할 수 있다. 즉, 이와 같은 우선순위-이름 프리픽스 트라이를 통해 우선순위 노드를 정의함으로써, 불필요한 빈 노드를 줄이고, 검색 성능 또한 향상시킬 수 있다.

[0049] 도 4는 본 발명의 일실시예에 따른 우선순위 이름 프리픽스 트라이 기반의 비트맵 예를 도시한 도면이다.

[0050] 이름 프리픽스 트라이의 각각의 노드에는 플래그(flag)(1bit), 자식 노드로의 포인터(pointer), 자식 노드에 대한 에지(edge) 정보(에지 1개당 32bit), 이름 프리픽스(128bit) 및 출력 페이스 정보를 모두 포함하여야 하기 때문에 많은 메모리를 필요로 한다. 본 발명에서는 비트맵(Bitmap)을 적용하여 프로세서(110)의 내부 메모리(120)에 우선순위-이름 프리픽스 트라이의 윤곽 정보를 저장하여, 프로세서(110)의 외부에 배치된 외부 메모리(130)에서의 프로세싱을 최소화할 수 있다. 이와 같은 비트맵은 우선순위-이름 프리픽스 트라이에 기초하여 이름 프리픽스 검색 장치(100)의 프로세서(110)에 의해 생성될 수 있다. 도 4의 비트맵 예는 도 3의 우선순위-이름 프리픽스 트라이에 기초하여 생성되었다.

[0051] 비트맵은 비트맵 테이블(Bitmap table), 인코딩 테이블(Encoding table) 및 노드 테이블(Node table)로 구성될 수 있다. 이때, 비트맵 테이블 및 인코딩 테이블은 이름 프리픽스를 검색하는 프로세서(110)의 내부 메모리(120)에 저장되고, 노드 테이블은 이름 프리픽스를 검색하는 프로세서(110)의 외부 메모리(130)에 저장될 수 있다.

[0052] 우선순위-이름 프리픽스 트라이의 노드 번호는 0부터 시작하며, 상위레벨에서 하위레벨 순으로 부여될 수 있다. 예를 들어, 도 3의 (b)와 같은 우선순위-이름 프리픽스 트라이에서 루트 노드의 노드 번호는 0이고, 'kr/or/warning'이 저장된 우선순위 노드의 노드 번호는 5가 된다.

[0053] 이때, 우선순위-이름 프리픽스 트라이를 이용하여 비트맵을 생성함에 있어서, 우선순위-이름 프리픽스 트라이에 포함된 각각의 노드 번호와 동일한 비트맵 테이블의 인덱스 번호 및 노드 테이블의 인덱스 번호를 매칭하여 노드 정보를 저장할 수 있다. 따라서, 비트맵 테이블과 노드 테이블을 연결하기 위한 별도의 포인터가 필요 없다.

[0054] A. 구체적으로 비트맵을 구성하는 비트맵 테이블은 우선순위-이름 프리픽스 트라이에 포함된 각각의 노드에 대한 5개의 필드 정보로 구성된다. 먼저 도 4의 비트맵 테이블의 a열과 같이 우선순위-이름 프리픽스 트라이에 포함된 각각의 노드들이 우선순위 노드인지 또는 비우선순위 노드인지를 나타내는 노드 타입에 대한 정보를 포함할 수 있다. 즉, 입력 패킷의 이름 프리픽스를 검색하기 위하여 접근한 우선순위-이름 프리픽스 트라이의 노드가 삽입 노드일 경우에는 0을, 우선순위 노드일 경우에는 1을 저장한다.

[0055] 다음으로 비트맵 테이블은 b열과 같이 우선순위-이름 프리픽스 트라이에 포함된 노드들 각각에 저장된 이름 프리픽스의 컴포넌트 개수에 대한 정보를 포함할 수 있다. 예를 들어, 비트맵 테이블의 인덱스 0은 도 3에서 우선순위-이름 프리픽스 트라이의 최상위 레벨에 존재하는 우선순위 노드를 나타낸다. 우선순위-이름 프리픽스 트라이의 0번 노드는 'com/youtube/user/skyDoesMinecraft'의 이름 프리픽스를 저장하고 있으며, 각각의 com, youtube, user 및 skyDoesMinecraft는 이름 프리픽스를 구성하는 컴포넌트에 해당한다. 따라서, 비트맵 테이블의 인덱스 0에 대응하는 컴포넌트의 개수는 4이다.

[0056] 다음으로 비트맵 테이블은 c열과 같이 우선순위-이름 프리픽스 트라이에 포함된 노드들 각각에 저장된 이름 프리픽스에 해시 함수를 적용하여 도출된 해시브리프를 포함할 수 있다. 이때, 도출된 해시브리프는 이름 프리픽스에 대한 핑거프린트(fingerprint)를 나타낼 수 있다.

[0057] 이와 같은 해시브리프는 이름 프리픽스 검색 장치(100)가 이름 프리픽스를 검색할 때 이용될 수 있다. 이름 프리픽스 트라이에서는 입력 패킷에 대한 이름 프리픽스가 경로상에 존재하는 모든 이름 프리픽스와 일치하는 반면 우선순위 트라이의 우선순위 노드에는 가장 많은 컴포넌트를 갖는 이름 프리픽스가 저장되기 때문에, 입력 패킷에 대한 이름 프리픽스가 우선순위 노드에 저장된 이름 프리픽스와 일치할 가능성이 적다. 이러한 우선순위-이름 프리픽스 트라이의 특성으로 인해 불필요한 외부 메모리(130)로의 접근이 빈번하게 발생할 수 있다. 본 발명에서는 해시브리프의 개념을 제안 및 적용하여, 우선순위 노드에 저장된 컴포넌트 개수에 대응하는 입력 패킷에 대한 이름 프리픽스의 컴포넌트에 대한 해시 값을 결정하고, 이를 이용하여 외부 메모리(130)로의 접근을 결정할 수 있다. 즉, 결정된 해시 값을 비트맵 테이블에 저장된 해시브리프와 비교하여 일치할 경우에만 외부 메모리(130)에 접근하기 때문에, 우선순위-이름 프리픽스 트라이의 특성으로 인해 불필요하게 발생하는 외부 메모리(130)로의 접근을 최소화할 수 있다.

[0058] 다음으로 비트맵 테이블은 d열과 같이 우선순위-이름 프리픽스 트라이에 포함된 노드들 각각에 대한 자식 노드의 개수를 나타내는 차일드 개수 정보를 포함할 수 있다. 예를 들어, 우선순위-이름 프리픽스 트라이의 0번 노드는 하위에 1번, 2번, 3번 및 4번의 4개의 자식 노드를 가지고 있는 것을 확인할 수 있다. 따라서, 비트맵 테

이블의 인덱스 0에 대응하는 차일드 개수 정보는 4이다.

- [0059] 다음으로 비트맵 테이블은 e열과 같이 우선순위-이름 프리픽스 트라이에 포함된 노드들 각각에 대한 차일드 포인터 정보를 포함할 수 있다. 차일드 포인터 정보는 우선순위-이름 프리픽스 트라이에 포함된 노드들 각각이 가진 첫번째 자식 노드로의 경로에 해당하는 인코딩 테이블의 주소값을 나타낼 수 있다.
- [0060] B. 비트맵을 구성하는 인코딩 테이블은 우선순위-이름 프리픽스 트라이에 포함된 노드들(우선순위 노드와 비우선순위 노드) 각각의 자식 노드에 대한 에지(edge) 정보 및 상기 우선순위-이름 프리픽스 트라이에 포함된 노드들과 상기 자식 노드 간의 인덱스 번호 차이를 나타내는 인코딩 번호를 포함할 수 있다.
- [0061] IP주소를 이용한 프리픽스 검색에서의 자식 노드는 최대 2개로 제한되어 있던 반면 NDN에서의 자식 노드의 개수에는 제한이 없다. 그러므로, 본 발명은 우선순위-이름 프리픽스 트라이에 포함된 노드들 각각에 대한 불특정한 자식 노드의 개수를 고려하여 비트맵을 적용할 때의 어려움을 해결하기 위해 인코딩 테이블을 도입하였다. 비트맵 테이블이 우선순위-이름 프리픽스 트라이에 포함된 노드들의 윤곽 정보를 저장했다면, 인코딩 테이블은 우선순위-이름 프리픽스 트라이에 포함된 노드가 가진 자식 노드로의 에지(edge) 정보를 포함할 수 있다. 이와 더불어 인코딩 테이블은 우선순위-이름 프리픽스 트라이에 포함된 노드와 자식 노드 간의 노드 번호 차이를 나타내는 인코딩 번호를 이용하여 우선순위-이름 프리픽스 트라이의 노드와 자식 노드를 연결할 수 있다.
- [0062] 예를 들어, 우선순위-이름 프리픽스 트라이의 0번 노드는 하위에 1번, 2번, 3번 및 4번의 4개의 자식 노드를 가지고 있다. 이때, 0번 노드로부터 각각의 자식 노드로의 경로에 해당하는 에지(edge) 정보는 kr, de, hk 및 com이며, 해당 에지 정보는 인코딩 테이블에 순차적으로 저장된다. 도 3의 (b)를 보면 알 수 있듯이 0번 노드에서 에지 정보 kr을 따라 이동하면 우선순위-이름 프리픽스 트라이의 1번 노드가 자식노드로 검색된다. 따라서 인코딩 테이블에는 0번 노드와 1번 노드 간의 노드 번호 차이인 1이 인코딩 번호로 저장될 수 있다. 마찬가지로 0번 노드에서 에지 정보 de를 따라 이동하면 우선순위-이름 프리픽스 트라이의 2번 노드가 자식노드로 검색된다. 이때, 인코딩 테이블에는 0번 노드와 2번 노드 간의 번호 차이인 2가 인코딩 번호로 저장될 수 있다.
- [0063] C. 비트맵을 구성하는 노드 테이블은 우선순위-이름 프리픽스 트라이에 포함된 노드들(우선순위 노드와 비우선순위 노드) 각각에 포함된 이름 프리픽스와 상기 이름 프리픽스에 대응하는 출력 페이지가 저장될 수 있다.
- [0064] 본 발명은 우선순위-이름 프리픽스 트라이를 이용하여 상기 A, B, C에 개시된 비트맵을 생성할 수 있다. 구체적으로 도 3의 우선순위-이름 프리픽스 트라이에 포함된 4번 노드를 비트맵으로 생성하는 예는 다음과 같다. 먼저 4번 노드에 대한 노드 정보는 비트맵 테이블 및 노드 테이블의 인덱스 4번의 위치에 저장된다. 4번 노드는 우선순위 노드이므로, 노드 타입에 대한 정보는 1이 되고, 저장된 이름 프리픽스는 'com/youtube/user/PewDiePie'이므로 컴포넌트 정보는 4(com, youtube, user, PewDiePie)가 된다. 그리고, 4번 노드는 하위에 4개의 자식 노드를 가지고 있기 때문에 차일드 개수 정보는 4가 된다. 4번 노드에서 자식 노드로의 경로에 해당하는 에지 정보는 순서대로 empowernetwork, youtube, goole, facebook이며 이와 같은 에지 정보를 인코딩 테이블에 순차적으로 저장된다. 그리고 비트맵 테이블의 차일드 포인터에는 4번 노드의 첫번째 자식 노드(empowernetwork)로의 경로에 해당하는 인코딩 테이블의 주소값이 저장되며, 4번 노드와 첫번째 자식 노드인 6번 노드 간의 노드 번호 차이인 2가 인코딩 번호로 저장된다.
- [0065] 도 5는 본 발명의 일실시예에 따른 비우선순위로 판단된 우선순위 이름 프리픽스 트라이의 노드에 대한 이름 프리픽스 검색 방법을 순서대로 도시한 도면이다.
- [0066] 이름 프리픽스 검색 장치(100)는 이름 프리픽스 트라이에서 도출된 우선순위 노드 및 비우선순위 노드를 포함하는 우선순위-이름 프리픽스 트라이에 기초하여 생성된 비트맵을 식별할 수 있다. 이때, 식별된 비트맵은 비트맵 테이블, 인코딩 테이블 및 노드 테이블로 구성될 수 있다.
- [0067] 먼저, 단계(510)에서 이름 프리픽스 검색 장치(100)는 입력 패킷의 이름 프리픽스를 검색하기 위해 비트맵 테이블의 인덱스 번호에 기초하여 우선순위-이름 프리픽스 트라이 내의 노드에 접근할 수 있다. 이때, 비트맵 테이블의 인덱스 번호는 우선순위-이름 프리픽스 트라이 내의 노드 번호에 대응할 수 있다. 즉, 비트맵 테이블의 0번 인덱스는 우선순위-이름 프리픽스 트라이 내의 0번 노드와 대응할 수 있다. 그리고 이름 프리픽스 검색 장치(100)는 우선순위-이름 프리픽스 트라이 내의 최상위 레벨에 존재하는 노드부터 입력 패킷의 이름 프리픽스를 검색할 수 있다.
- [0068] 따라서, 이름 프리픽스 검색 장치(100)는 비트맵 테이블의 0번 인덱스에 포함된 노드 타입 정보에 기초하여 우선순위-이름 프리픽스 트라이 내의 노드가 우선순위 노드인지 또는 비우선순위 노드인지를 확인할 수 있다.

- [0069] 단계(520)에서 이름 프리픽스 검색 장치(100)는 확인 결과 우선순위-이름 프리픽스 트라이에 포함된 노드가 비 우선순위 노드로 확인되면, 상기 확인된 비우선순위 노드의 인덱스 번호와 동일한 노드 테이블의 인덱스 번호에 접근할 수 있다. 이후 이름 프리픽스 검색 장치(100)는 접근한 노드 테이블의 인덱스 번호에 저장된 출력 페이지를 추출하여 기억할 수 있다.
- [0070] 이후 단계(530)에서 이름 프리픽스 검색 장치(100)는 비트맵 테이블의 차일드 개수 정보를 이용하여 상기 확인된 비우선순위 노드가 자식 노드를 가지는지의 여부를 확인할 수 있다.
- [0071] 상기 확인된 비우선순위 노드가 자식 노드를 가지지 않는 경우, 이름 프리픽스 검색 장치(100)는 단계(560)과 같이 단계(520)에서 기억하고 있는 출력 페이지를 검색 결과로 반환하고 검색을 종료할 수 있다.
- [0072] 이와는 달리 상기 확인된 비우선순위 노드가 자식 노드를 가지는 경우, 이름 프리픽스 검색 장치(100)는 단계(540)에서 상기 확인된 비우선순위 노드에 대응하는 비트맵 테이블의 차일드 포인터 정보를 이용하여 인코딩 테이블에 접근할 수 있다. 이후 이름 프리픽스 검색 장치(100)는 인코딩 테이블을 이용하여 다음에 검색을 진행할 경로와 일치하는 자식 노드에 대한 에지 정보를 확인할 수 있다.
- [0073] 이때, 일치하는 자식 노드에 대한 에지 정보가 없는 경우, 이름 프리픽스 검색 장치(100)는 단계(560)과 같이 단계(520)에서 기억하고 있는 출력 페이지를 검색 결과로 반환하고 검색을 종료할 수 있다.
- [0074] 그러나 일치하는 자식 노드에 대한 에지 정보가 있는 경우, 이름 프리픽스 검색 장치(100)는 단계(550)에서 상기 확인된 비우선순위 노드의 인덱스 번호와 상기 확인된 에지 정보에 대응하는 인코딩 번호를 이용하여 다음에 검색을 진행할 비트맵 테이블의 인덱스를 결정할 수 있다.
- [0075] 이후 이름 프리픽스 검색 장치(100)는 결정된 비트맵 테이블의 인덱스에 기초하여 단계(510)~단계(560)을 반복하여 수행함으로써 입력 패킷에 대한 이름 프리픽스를 검색할 수 있다.
- [0076] 도 6은 본 발명의 일실시예에 따른 우선순위로 판단된 우선순위 이름 프리픽스 트라이의 노드에 대한 이름 프리픽스 검색 방법을 순서대로 도시한 도면이다.
- [0077] 먼저 단계(610)에서 이름 프리픽스 검색 장치(100)는 입력 패킷의 이름 프리픽스를 검색하기 위해 비트맵 테이블의 인덱스 번호에 기초하여 우선순위-이름 프리픽스 트라이 내의 노드에 접근할 수 있다. 이때, 비트맵 테이블의 인덱스 번호는 우선순위-이름 프리픽스 트라이 내의 노드 번호에 대응할 수 있다. 즉, 비트맵 테이블의 0번 인덱스는 우선순위-이름 프리픽스 트라이 내의 0번 노드와 대응할 수 있다. 그리고 이름 프리픽스 검색 장치(100)는 우선순위-이름 프리픽스 트라이 내의 최상위 레벨에 존재하는 노드부터 입력 패킷의 이름 프리픽스를 검색할 수 있다.
- [0078] 따라서, 이름 프리픽스 검색 장치(100)는 비트맵 테이블의 0번 인덱스에 포함된 노드 타입 정보에 기초하여 우선순위-이름 프리픽스 트라이 내의 노드가 우선순위 노드인지 또는 비우선순위 노드인지를 확인할 수 있다.
- [0079] 단계(620)에서 이름 프리픽스 검색 장치(100)는 단계(610)에서 확인된 우선순위 노드에 대한 해시브리프와 입력 패킷의 이름 프리픽스에 대한 해시 값을 비교할 수 있다. 이때, 입력 패킷의 이름 프리픽스에 대한 해시 값은 비트맵 테이블의 해시브리프를 생성할 때 사용한 해시 함수와 동일한 해시 함수를 이용하여 추출할 수 있다. 그리고 이름 프리픽스 검색 장치(100)는 상기 확인된 우선순위 노드에 저장된 이름 프리픽스의 컴포넌트의 개수에 대응하는 입력 패킷에 대한 이름 프리픽스를 이루는 컴포넌트에 기초하여 해시 값을 결정할 수 있다.
- [0080] 이때, 비교된 우선순위 노드에 대한 해시브리프와 입력 패킷의 이름 프리픽스에 대한 해시 값이 일치하는 경우, 이름 프리픽스 검색 장치(100)는 단계(630)에서 상기 확인된 우선순위 노드에 저장된 이름 프리픽스와 상기 확인된 우선순위 노드의 인덱스 번호와 동일한 노드 테이블의 인덱스 번호에 저장된 이름 프리픽스가 일치하는지 비교할 수 있다.
- [0081] 비교 결과 일치하는 경우, 이름 프리픽스 검색 장치(100)는 단계(640)과 같이 상기 확인된 우선순위 노드의 인덱스 번호와 동일한 노드 테이블의 인덱스 번호에 저장된 출력 페이지를 추출하여 기억할 수 있다.
- [0082] 이후 단계(680)에서 이름 프리픽스 검색 장치(100)는 단계(640)에서 기억하고 있는 출력 페이지를 검색 결과로 반환하고 검색을 종료할 수 있다.
- [0083] 이와는 달리 비교된 우선순위 노드에 대한 해시브리프와 입력 패킷의 이름 프리픽스에 대한 해시 값이 일치하지 않는 경우, 이름 프리픽스 검색 장치(100)는 단계(650)에서 비트맵 테이블의 차일드 개수 정보를 이용하여 상기

확인된 우선순위 노드가 자식 노드를 가지는지의 여부를 확인할 수 있다.

- [0084] 이후 상기 확인된 우선순위 노드가 자식 노드를 가지지 않는다면, 이름 프리픽스 검색 장치(100)는 단계(680)과 같이 단계(640)에서 기억하고 있는 출력 페이지를 검색 결과로 반환하고 검색을 종료할 수 있다.
- [0085] 그러나 상기 확인된 우선순위 노드가 자식 노드를 가지는 경우, 이름 프리픽스 검색 장치(100)는 단계(660)에서 상기 확인된 우선순위 노드에 대응하는 비트맵 테이블의 차일드 포인터 정보를 이용하여 인코딩 테이블에 접근할 수 있다. 이후 이름 프리픽스 검색 장치(100)는 인코딩 테이블을 이용하여 다음에 검색을 진행할 경로와 일치하는 자식 노드에 대한 예지 정보를 확인할 수 있다.
- [0086] 이때, 일치하는 자식 노드에 대한 예지 정보가 없는 경우, 이름 프리픽스 검색 장치(100)는 단계(680)과 같이 단계(640)에서 기억하고 있는 출력 페이지를 검색 결과로 반환하고 검색을 종료할 수 있다.
- [0087] 그러나 일치하는 자식 노드에 대한 예지 정보가 있는 경우, 이름 프리픽스 검색 장치(100)는 단계(670)에서 상기 확인된 우선순위 노드의 인덱스 번호와 상기 확인된 예지 정보에 대응하는 인코딩 번호를 이용하여 다음에 검색을 진행할 비트맵 테이블의 인덱스를 결정할 수 있다.
- [0088] 이후 이름 프리픽스 검색 장치(100)는 결정된 비트맵 테이블의 인덱스에 기초하여 단계(610)~단계(680)을 반복하여 수행함으로써 입력 패킷에 대한 이름 프리픽스를 검색할 수 있다.
- [0089] 예를 들어, 이름 프리픽스 검색 방법에 따라 입력 패킷의 이름 프리픽스인 'com/goole/mail/mailbox/user1'을 검색하면 다음과 같을 수 있다. 먼저 비트맵 테이블의 0번 인덱스에 대응하는 우선순위-이름 프리픽스 트라이의 노드는 노드 타입이 1이므로 우선순위 노드에 해당한다. 그리고 이름 프리픽스 검색 장치(100)는 0번 인덱스에 대응하는 컴포넌트의 개수가 4이므로 'com/goole/mail/mailbox/user1' 중 'comgoolemailmailbox'만을 이용하여 해시 값을 결정할 수 있다. 이때, 결정된 해시 값이 010이라고 가정하자.
- [0090] 이때 결정된 해시 값이 0번 인덱스에 대응하는 해시브리프와 일치하지 않으므로 이름 프리픽스 검색 장치(100)는 차일드 포인터 정보를 이용하여 인코딩 테이블에 접근할 수 있다. 이후 이름 프리픽스 검색 장치(100)는 인코딩 테이블을 이용하여 다음에 검색을 진행할 경로와 일치하는 자식 노드에 대한 예지 정보를 확인할 수 있다. 확인 결과 입력 패킷의 이름 프리픽스 중 'com'이 일치하는 것을 확인할 수 있다.
- [0091] 이름 프리픽스 검색 장치(100)는 인코딩 테이블에서 'com'에 대응하는 인코딩 번호 4와 현재 검색을 진행중인 우선순위-이름 프리픽스 트라이의 노드에 대한 인덱스 번호 0을 합하여 다음에 검색을 진행할 비트맵 테이블의 인덱스 번호 4를 결정할 수 있다.
- [0092] 이후 이름 프리픽스 검색 장치(100)는 새롭게 결정된 비트맵 테이블의 인덱스 번호 4에 기초하여 이름 프리픽스의 검색을 재수행할 수 있다. 비트맵 테이블의 인덱스 번호 4에 대응하는 우선순위-이름 프리픽스 트라이의 노드는 우선순위 노드이므로 입력 패킷의 이름 프리픽스에 대한 해시 값과 4번 인덱스에 대응하는 해시브리프를 비교할 수 있다.
- [0093] 만약 비교 결과 일치하지 않는다고 가정하면 이름 프리픽스 검색 장치(100)는 차일드 포인터 정보를 이용하여 인코딩 테이블에 접근할 수 있다. 이후 이름 프리픽스 검색 장치(100)는 인코딩 테이블을 이용하여 다음에 검색을 진행할 경로와 일치하는 자식 노드에 대한 예지 정보를 확인할 수 있다. 확인 결과 입력 패킷의 이름 프리픽스 중 'google'이 일치하는 것을 확인할 수 있다.
- [0094] 이후 이름 프리픽스 검색 장치(100)는 인코딩 테이블에서 'google'에 대응하는 인코딩 번호 4와 현재 검색을 진행중인 우선순위-이름 프리픽스 트라이의 노드에 대한 인덱스 번호 4를 합하여 다음에 검색을 진행할 비트맵 테이블의 인덱스 번호 8을 결정할 수 있다.
- [0095] 이후 이름 프리픽스 검색 장치(100)는 결정된 비트맵 테이블의 인덱스 번호 8에 기초하여 이름 프리픽스의 검색을 재수행할 수 있다. 비트맵 테이블의 인덱스 번호 8에 대응하는 우선순위-이름 프리픽스 트라이의 노드는 비우선순위 노드이다. 따라서, 이름 프리픽스 검색 장치(100)는 비트맵 테이블의 인덱스 번호 8과 동일한 노드 테이블의 인덱스 번호에 접근할 수 있다. 이후 이름 프리픽스 검색 장치(100)는 접근한 노드 테이블의 인덱스 번호에 저장된 출력 페이지인 9를 추출하여 기억할 수 있다.
- [0096] 이때, 비트맵 테이블의 인덱스 번호 8에 대응하는 우선순위-이름 프리픽스 트라이는 자식 노드가 없으므로 이름 프리픽스 검색 장치(100)는 상기 기억된 출력 페이지 9를 반환하고 검색을 종료할 수 있다.
- [0097] 이상에서 설명된 장치는 하드웨어 구성요소, 소프트웨어 구성요소, 및/또는 하드웨어 구성요소 및 소프트웨어

구성요소의 조합으로 구현될 수 있다. 예를 들어, 실시예들에서 설명된 장치 및 구성요소는, 예를 들어, 프로세서, 콘트롤러, ALU(arithmetic logic unit), 디지털 신호 프로세서(digital signal processor), 마이크로컴퓨터, FPGA(field programmable gate array), PLU(programmable logic unit), 마이크로프로세서, 또는 명령(instruction)을 실행하고 응답할 수 있는 다른 어떠한 장치와 같이, 하나 이상의 범용 컴퓨터 또는 특수 목적 컴퓨터를 이용하여 구현될 수 있다. 처리 장치는 운영 체제(OS) 및 상기 운영 체제 상에서 수행되는 하나 이상의 소프트웨어 애플리케이션을 수행할 수 있다. 또한, 처리 장치는 소프트웨어의 실행에 응답하여, 데이터를 접근, 저장, 조작, 처리 및 생성할 수도 있다. 이해의 편의를 위하여, 처리 장치는 하나가 사용되는 것으로 설명된 경우도 있지만, 해당 기술분야에서 통상의 지식을 가진 자는, 처리 장치가 복수 개의 처리 요소(processing element) 및/또는 복수 유형의 처리 요소를 포함할 수 있음을 알 수 있다. 예를 들어, 처리 장치는 복수 개의 프로세서 또는 하나의 프로세서 및 하나의 콘트롤러를 포함할 수 있다. 또한, 병렬 프로세서(parallel processor)와 같은, 다른 처리 구성(processing configuration)도 가능하다.

[0098] 소프트웨어는 컴퓨터 프로그램(computer program), 코드(code), 명령(instruction), 또는 이들 중 하나 이상의 조합을 포함할 수 있으며, 원하는 대로 동작하도록 처리 장치를 구성하거나 독립적으로 또는 결합적으로(collectively) 처리 장치를 명령할 수 있다. 소프트웨어 및/또는 데이터는, 처리 장치에 의하여 해석되거나 처리 장치에 명령 또는 데이터를 제공하기 위하여, 어떤 유형의 기계, 구성요소(component), 물리적 장치, 가상장치(virtual equipment), 컴퓨터 저장 매체 또는 장치, 또는 전송되는 신호 파(signal wave)에 영구적으로, 또는 일시적으로 구체화(embody)될 수 있다. 소프트웨어는 네트워크로 연결된 컴퓨터 시스템 상에 분산되어서, 분산된 방법으로 저장되거나 실행될 수도 있다. 소프트웨어 및 데이터는 하나 이상의 컴퓨터 판독 가능 기록 매체에 저장될 수 있다.

[0099] 실시예에 따른 방법은 다양한 컴퓨터 수단을 통하여 수행될 수 있는 프로그램 명령 형태로 구현되어 컴퓨터 판독 가능 매체에 기록될 수 있다. 상기 컴퓨터 판독 가능 매체는 프로그램 명령, 데이터 파일, 데이터 구조 등을 단독으로 또는 조합하여 포함할 수 있다. 상기 매체에 기록되는 프로그램 명령은 실시예를 위하여 특별히 설계되고 구성된 것들이거나 컴퓨터 소프트웨어 당업자에게 공지되어 사용 가능한 것일 수도 있다. 컴퓨터 판독 가능 기록 매체의 예에는 하드 디스크, 플로피 디스크 및 자기 테이프와 같은 자기 매체(magnetic media), CD-ROM, DVD와 같은 광기록 매체(optical media), 플롭티컬 디스크(floptical disk)와 같은 자기-광 매체(magneto-optical media), 및 롬(ROM), 램(RAM), 플래시 메모리 등과 같은 프로그램 명령을 저장하고 수행하도록 특별히 구성된 하드웨어 장치가 포함된다. 프로그램 명령의 예에는 컴파일러에 의해 만들어지는 것과 같은 기계어 코드뿐만 아니라 인터프리터 등을 사용해서 컴퓨터에 의해서 실행될 수 있는 고급 언어 코드를 포함한다. 상기된 하드웨어 장치는 실시예의 동작을 수행하기 위해 하나 이상의 소프트웨어 모듈로서 작동하도록 구성될 수 있으며, 그 역도 마찬가지이다.

[0100] 이상과 같이 실시예들이 비록 한정된 실시예와 도면에 의해 설명되었으나, 해당 기술분야에서 통상의 지식을 가진 자라면 상기의 기재로부터 다양한 수정 및 변형이 가능하다. 예를 들어, 설명된 기술들이 설명된 방법과 다른 순서로 수행되거나, 및/또는 설명된 시스템, 구조, 장치, 회로 등의 구성요소들이 설명된 방법과 다른 형태로 결합 또는 조합되거나, 다른 구성요소 또는 균등물에 의하여 대치되거나 치환되더라도 적절한 결과가 달성될 수 있다.

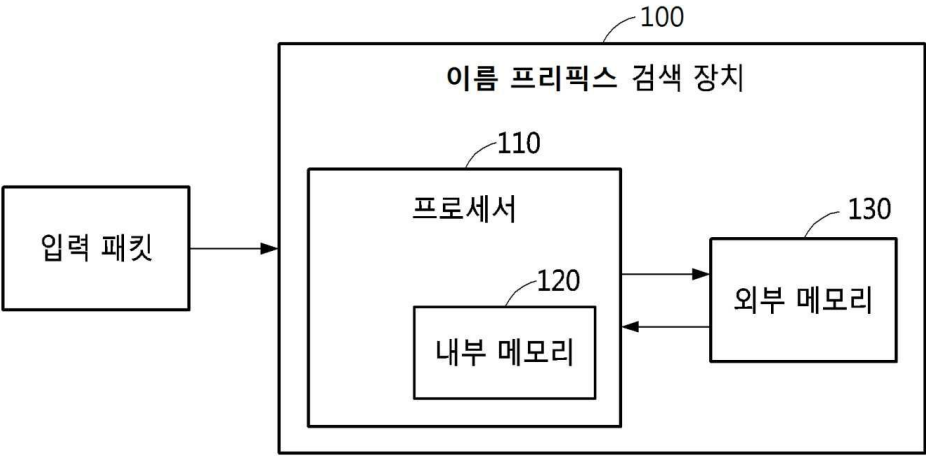
[0101] 그러므로, 다른 구현들, 다른 실시예들 및 특허청구범위와 균등한 것들도 후술하는 특허청구범위의 범위에 속한다.

부호의 설명

[0102] 100 : 이름 프리픽스 검색 장치
110 : 프로세서
120 : 내부 메모리
130 : 외부 메모리

도면

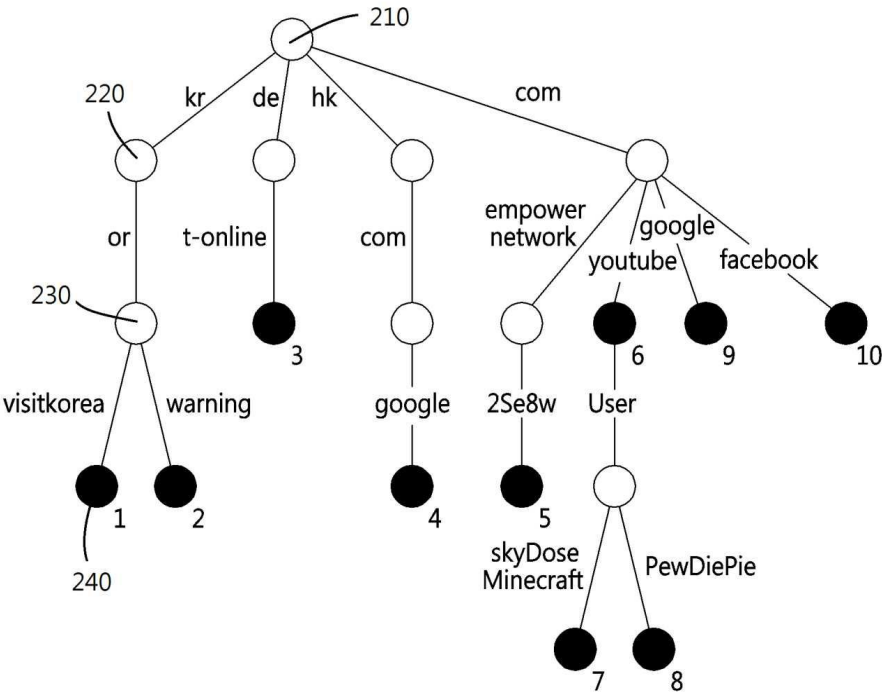
도면1



도면2

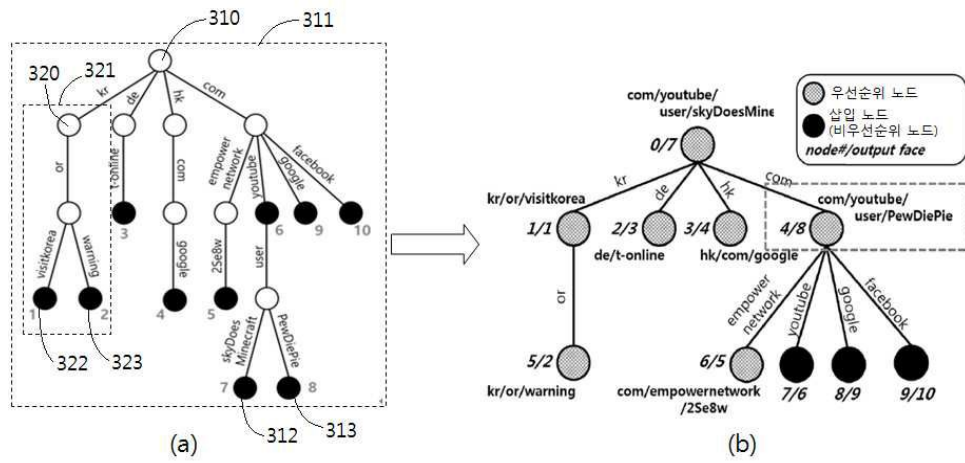
Name prefix	Output face
kr/or/visitkorea	1
kr/or/warning	2
de/t-online	3
hk/com/google	4
com/empowernetwork/2Se8w	5
com/youtube	6
com/youtube/user/skyDoesMinecraft	7
com/youtube/user/pewDiePie	8
com/google	9
com/facebook	10

(a)

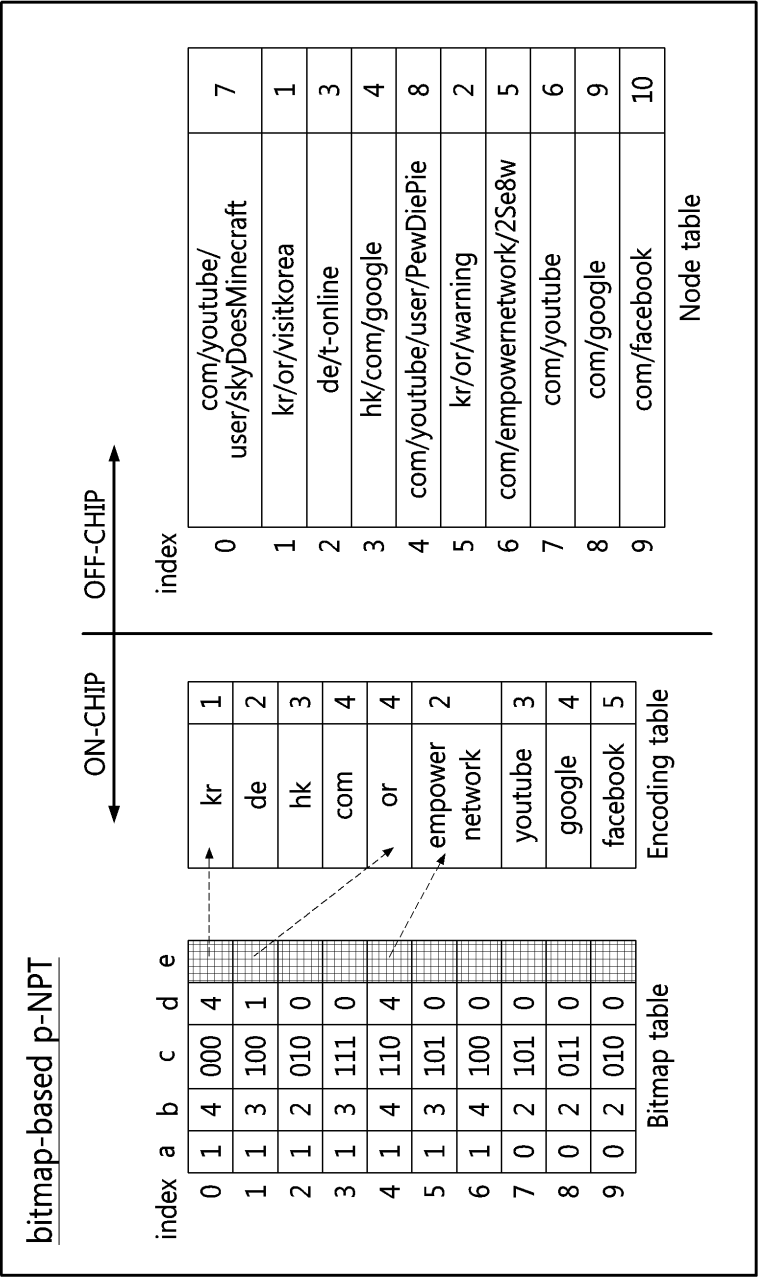


(b)

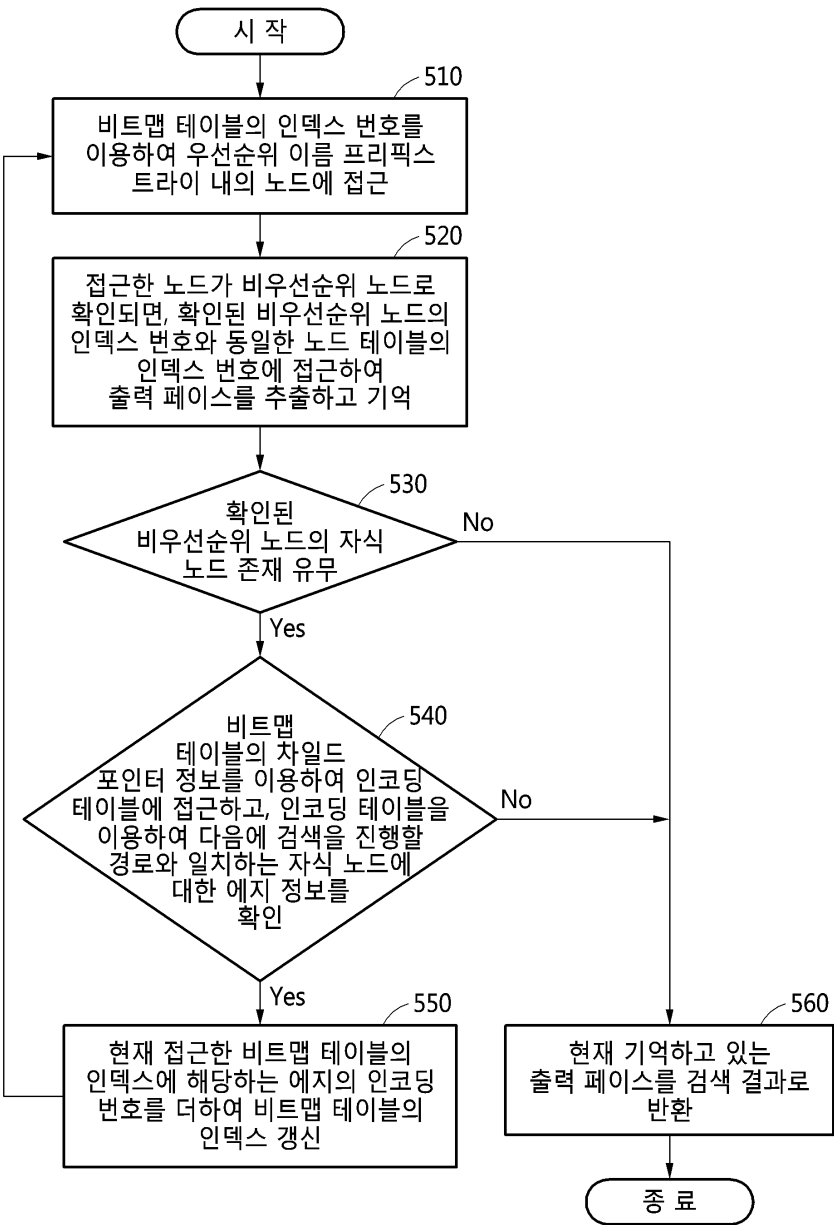
도면3



도면4



도면5



도면6

