



(19) 대한민국특허청(KR)

(12) 등록특허공보(B1)

(45) 공고일자 2015년12월31일

(11) 등록번호 10-1582050

(24) 등록일자 2015년12월24일

(51) 국제특허분류(Int. Cl.)

G06F 17/30 (2006.01)

(21) 출원번호 10-2014-0145334

(22) 출원일자 2014년10월24일

심사청구일자 2014년10월24일

(56) 선행기술조사문헌

JP07064801 A*

KR1020100010987 A*

KR100996136 B1

*는 심사관에 의하여 인용된 문헌

(73) 특허권자

이화여자대학교 산학협력단

서울특별시 서대문구 이화여대길 52 (대현동, 이화여자대학교)

(72) 발명자

임혜숙

서울특별시 강남구 남부순환로 2803 삼성래미안아파트 108동 1102호

심미란

서울특별시 서대문구 봉원사길 41-3, 1층

(74) 대리인

특허법인 무한

전체 청구항 수 : 총 12 항

심사관 : 김병균

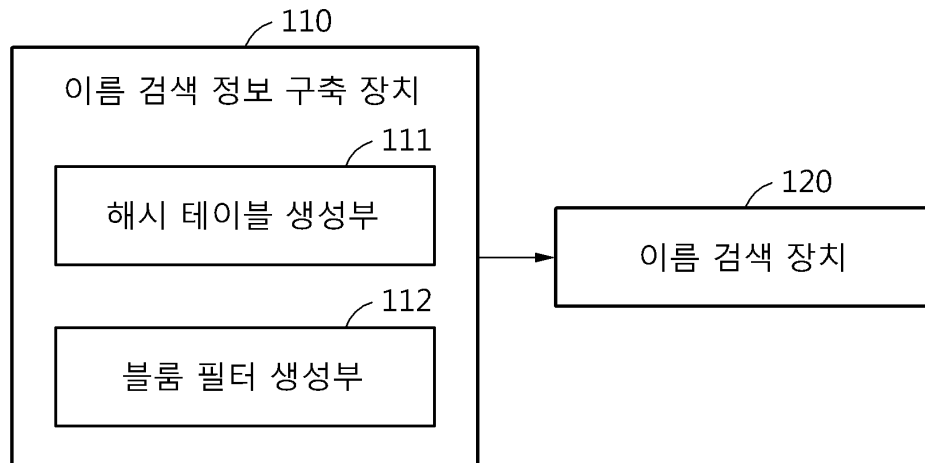
(54) 발명의 명칭 **블룸 필터 선-검색을 이용한 이름 검색 장치 및 방법**

(57) 요약

블룸 필터 선-검색을 이용한 이름 검색 장치 및 방법이 개시된다.

복수의 레벨로 구성된 이름 프리픽스 트리(Name Prefix Trie: NPT)를 이용한 이름 검색 방법은 NPT의 레벨들 별로 생성된 블룸 필터로 콘텐츠 이름을 검색하는 단계; 및 상기 블룸 필터의 검색 결과에 기초하여 상기 블룸 필터에 대응하는 해시 테이블에서 상기 콘텐츠 이름을 검색하는 단계를 포함할 수 있다.

대표도 - 도1



이 발명을 지원한 국가연구개발사업

과제고유번호 1711004610

부처명 미래창조과학부

연구관리전문기관 한국연구재단

연구사업명 기초연구사업(학술진흥)-중견연구자지원사업-핵심연구(개인)

연구과제명 차세대인터넷을위한고속패킷전달기술에관한연구

기 여 율 1/1

주관기관 이화여자대학교 산학협력단

연구기간 2014.05.01 ~ 2015.04.30

명세서

청구범위

청구항 1

복수의 레벨로 구성된 이름 프리픽스 트리(Name Prefix Trie: NPT)를 이용한 이름 검색 방법에 있어서,
 해시 함수를 사용하여 NPT의 레벨들 각각에 대응하는 콘텐츠 이름의 컴포넌트를 변환하는 단계; 및
 NPT의 레벨들 별로 생성된 해시 테이블에서 상기 해시 함수를 사용하여 변환된 컴포넌트를 검색하는 단계
 를 포함하고,
 상기 컴포넌트를 변환하는 단계는,
 NPT의 레벨이 두 번째 레벨 이상인 경우, 해시 함수를 사용하여 이전 레벨에 대응하는 콘텐츠 이름의 컴포넌트
 와 현재 레벨에 대응하는 콘텐츠 이름의 컴포넌트가 연속된 문자열을 변환하는 이름 검색 방법.

청구항 2

제1항에 있어서,
 상기 컴포넌트를 변환하는 단계는,
 NPT의 레벨들 별로 생성된 해시 테이블에서 해시 인덱스의 생성에 이용된 해시 함수를 이용하여 NPT의 레벨들
 각각에 대응하는 콘텐츠 이름의 컴포넌트를 변환하는 이름 검색 방법.

청구항 3

삭제

청구항 4

제1항에 있어서,
 상기 컴포넌트를 검색하는 단계는,
 현재 레벨에 대응하는 해시 테이블에서 이전 레벨에 대응하는 콘텐츠 이름의 컴포넌트와 현재 레벨에 대응하는
 콘텐츠 이름의 컴포넌트가 연속된 문자열이 해시 함수로 변환된 해시 인덱스를 검색하는 이름 검색 방법.

청구항 5

복수의 레벨로 구성된 이름 프리픽스 트리(Name Prefix Trie: NPT)를 이용한 이름 검색 방법에 있어서,
 해시 함수를 사용하여 NPT의 레벨들 각각에 대응하는 콘텐츠 이름의 컴포넌트를 변환하는 단계;
 NPT의 레벨들 별로 생성된 해시 테이블에서 상기 해시 함수를 사용하여 변환된 컴포넌트를 검색하는 단계; 및
 현재 레벨에 대응하는 해시 테이블에서 상기 해시 함수를 사용하여 변환된 컴포넌트가 검색되지 않는 경우, 이
 전 레벨의 해시 테이블에 포함된 마지막으로 검색된 출력 페이스로 상기 콘텐츠 이름에 대응하는 콘텐츠 요청
 패킷을 전송하는 단계
 를 포함하는 이름 검색 방법.

청구항 6

복수의 레벨로 구성된 이름 프리픽스 트리(Name Prefix Trie: NPT)를 이용한 이름 검색 방법에 있어서,
 NPT의 레벨들 별로 생성된 블룸 필터로 콘텐츠 이름을 검색하는 단계; 및
 상기 블룸 필터의 검색 결과에 기초하여 상기 블룸 필터에 대응하는 해시 테이블에서 상기 콘텐츠 이름을 검색
 하는 단계

를 포함하고,

상기 블록 필터로 콘텐츠 이름을 검색하는 단계는,

NPT의 현재 레벨의 블록 필터에 대응하는 해시 테이블에서 상기 콘텐츠 이름이 검색되는 경우, NPT의 다음 레벨의 블록 필터로 콘텐츠 이름을 검색하는 이름 검색 방법.

청구항 7

삭제

청구항 8

복수의 레벨로 구성된 이름 프리픽스 트리(Name Prefix Trie: NPT)를 이용한 이름 검색 방법에 있어서,

NPT의 레벨들 별로 생성된 블록 필터로 콘텐츠 이름을 검색하는 단계; 및

상기 블록 필터의 검색 결과에 기초하여 상기 블록 필터에 대응하는 해시 테이블에서 상기 콘텐츠 이름을 검색하는 단계

를 포함하고,

상기 해시 테이블에서 상기 콘텐츠 이름을 검색하는 단계는,

상기 블록 필터의 검색 결과가 양성인 경우, 상기 블록 필터에 대응하는 해시 테이블에서 상기 콘텐츠 이름을 검색하는 단계; 및

상기 블록 필터의 검색 결과가 음성이거나, 상기 해시 테이블에서 상기 콘텐츠 이름이 검색되지 않는 경우, 검색을 종료하는 단계

를 포함하는 이름 검색 방법.

청구항 9

제8항에 있어서,

상기 검색을 종료하는 단계는,

콘텐츠 이름이 마지막으로 검색된 해시 테이블에 포함된 출력 페이스로 상기 콘텐츠 이름에 대응하는 콘텐츠 요청 패킷을 전송하는 이름 검색 방법.

청구항 10

복수의 레벨로 구성된 이름 프리픽스 트리(Name Prefix Trie: NPT)를 이용한 이름 검색 방법에 있어서,

NPT의 레벨들 별로 생성된 블록 필터로 콘텐츠 이름을 검색하는 단계; 및

상기 블록 필터의 검색 결과에 기초하여 상기 블록 필터에 대응하는 해시 테이블에서 상기 콘텐츠 이름을 검색하는 단계

를 포함하고,

상기 블록 필터는,

이전 레벨에 대응하는 콘텐츠 이름의 컴포넌트와 현재 레벨에 대응하는 컴포넌트가 연속된 문자열을 포함하는 이름 검색 방법.

청구항 11

복수의 레벨로 구성된 이름 프리픽스 트리(Name Prefix Trie: NPT)를 이용한 이름 검색 방법에 있어서,

NPT의 레벨들 별로 생성된 블록 필터들에서 콘텐츠 이름을 연쇄적으로 검색하는 단계; 및

상기 블록 필터들의 검색 결과에 따른 해시 테이블에서 상기 콘텐츠 이름을 검색하는 단계

를 포함하고,

상기 콘텐츠 이름을 연쇄적으로 검색하는 단계는,

NPT의 현재 레벨에 대응하는 블록 필터로 콘텐츠 이름에 포함된 컴포넌트를 검색하는 단계;

상기 블록 필터의 검색 결과가 양성인 경우, NPT의 다음 레벨의 블록 필터로 콘텐츠 이름에 포함된 컴포넌트를 검색하는 단계; 및

상기 블록 필터의 검색 결과가 음성이거나, 현재 레벨이 NPT의 마지막 레벨인 경우, 콘텐츠 이름의 연쇄 검색을 종료하는 단계

를 포함하는 이름 검색 방법.

청구항 12

삭제

청구항 13

제11항에 있어서,

상기 해시 테이블에서 상기 콘텐츠 이름을 검색하는 단계는,

상기 블록 필터의 검색 결과가 음성인 경우, 이전 레벨의 블록 필터에 대응하는 해시 테이블에서 상기 콘텐츠 이름을 검색하는 이름 검색 방법.

청구항 14

제11항에 있어서,

상기 해시 테이블에서 상기 콘텐츠 이름을 검색하는 단계는,

블록 필터의 검색 결과가 양성이며, 현재 레벨이 NPT의 마지막 레벨인 경우, 현재 레벨의 블록 필터에 대응하는 해시 테이블에서 상기 콘텐츠 이름을 검색하는 이름 검색 방법.

청구항 15

복수의 레벨로 구성된 이름 프리픽스 트리(Name Prefix Trie: NPT)를 이용한 이름 검색 방법에 있어서,

NPT의 레벨들 별로 생성된 블록 필터들에서 콘텐츠 이름을 연쇄적으로 검색하는 단계; 및

상기 블록 필터들의 검색 결과에 따른 해시 테이블에서 상기 콘텐츠 이름을 검색하는 단계

를 포함하고,

상기 해시 테이블에서 상기 콘텐츠 이름을 검색하는 단계는,

블록 필터들의 검색 결과에 따른 해시 테이블에서 콘텐츠 이름이 검색되지 않는 경우, 이전 레벨의 블록 필터에 대응하는 해시 테이블에서 상기 콘텐츠 이름을 검색하는 이름 검색 방법.

발명의 설명

기술 분야

[0001] 본 발명은 이름 검색 장치 및 방법에 관한 것으로, 보다 상세하게는 콘텐츠 중심 네트워킹 기술에서 콘텐츠 이름을 검색하는 장치 및 방법에 관한 것이다.

배경 기술

[0002] 미래 인터넷으로 최근 각광 받고 있는 콘텐츠 중심 네트워킹(Content Centric Networking, CCN) 기술은 콘텐츠 중심의 전송 방식을 제공함으로써 대용량 정보를 빠르고 효율적으로 전송하고자 하는 기술이다.

[0003] CCN은 송수신자라는 개념 대신 정보를 생성한 콘텐츠 생성자와 정보를 수신하여 소비하는 콘텐츠 소비자의 개념을 도입하였고, 라우터는 IP 주소 대신 콘텐츠 이름을 사용하여 라우팅을 수행할 수 있다. 이때, CCN 라우터는

네트워크상에서 정보를 일시 저장하는 캐시 역할을 수행하며, 콘텐츠를 저장하고 있다가 콘텐츠를 요구하는 인근의 새로운 소비자들에게 콘텐츠를 배포할 수 있다.

[0004] 즉, CCN 라우터가 캐시 역할까지 수행함으로써, CCN은 소비자들에게 빠른 서비스를 제공할 수 있으며 동일한 콘텐츠가 네트워크 상에서 반복적으로 전송되는 횟수를 줄일 수 있다.

[0005] 그리고, CCN 라우터는 콘텐츠 전송을 위하여 CS(Contents Store), PIT(Pending Interest Table), FIB(Forwarding Information Base)의 3종류의 테이블을 사용할 수 있다. 이때, CS는 콘텐츠를 저장하는 캐시이고, PIT는 요청 받은 콘텐츠 중에서 소비자에게 제공하지 못한 콘텐츠들의 리스트이며, FIB는 소비자의 콘텐츠 요청 패킷을 전달해야 하는 출력 인터페이스(output interface) 리스트일 수 있다. 그리고, PIT와 FIB를 검색할 때에는 콘텐츠 이름과 가장 길게 일치하는 엔트리를 찾아야 하며, 신속도로 수행되어야 하므로 콘텐츠 이름을 빠른 속도로 검색할 수 있는 방법이 요청되고 있다.

발명의 내용

해결하려는 과제

[0006] 본 발명은 해시 테이블을 이용하여 NPT에 기초한 콘텐츠 이름을 검색할 수 있는 장치 및 방법을 제공할 수 있다.

[0007] 또한, 본 발명은 블록 필터의 검색 결과에 따라 해시 테이블의 검색 여부를 결정함으로써, 해시 테이블의 접근 회수를 감소시켜 검색 속도를 향상시킬 수 있는 장치 및 방법을 제공할 수 있다.

[0008] 그리고, 본 발명은 NPT의 레벨 별로 생성된 블록 필터를 연속으로 검색하여 최종 레벨을 검색하고, 최종 레벨의 해시 테이블을 검색함으로써, 해시 테이블의 접근 회수를 최소화시킬 수도 있는 장치 및 방법을 제공할 수 있다.

과제의 해결 수단

[0009] 본 발명의 일실시예에 따른 이름 검색 방법은 해시 함수를 사용하여 NPT의 레벨들 각각에 대응하는 콘텐츠 이름의 컴포넌트를 변환하는 단계; 및 NPT의 레벨들 별로 생성된 해시 테이블에서 상기 해시 함수를 사용하여 변환된 컴포넌트를 검색하는 단계를 포함할 수 있다.

[0010] 본 발명의 일실시예에 따른 이름 검색 방법의 컴포넌트를 변환하는 단계는 NPT의 레벨들 별로 생성된 해시 테이블에서 해시 인덱스의 생성에 이용된 해시 함수를 이용하여 NPT의 레벨들 각각에 대응하는 콘텐츠 이름의 컴포넌트를 변환할 수 있다.

[0011] 본 발명의 일실시예에 따른 이름 검색 방법의 컴포넌트를 변환하는 단계는 NPT의 레벨이 두 번째 레벨 이상인 경우, 해시 함수를 사용하여 이전 레벨에 대응하는 콘텐츠 이름의 컴포넌트와 현재 레벨에 대응하는 콘텐츠 이름의 컴포넌트가 연속된 문자열을 변환할 수 있다.

[0012] 본 발명의 일실시예에 따른 이름 검색 방법의 컴포넌트를 검색하는 단계는 현재 레벨에 대응하는 해시 테이블에서 이전 레벨에 대응하는 콘텐츠 이름의 컴포넌트와 현재 레벨에 대응하는 콘텐츠 이름의 컴포넌트가 연속된 문자열이 해시 함수로 변환된 해시 인덱스를 검색할 수 있다.

[0013] 본 발명의 일실시예에 따른 이름 검색 방법은 현재 레벨에 대응하는 해시 테이블에서 상기 해시 함수를 사용하여 변환된 컴포넌트가 검색되지 않는 경우, 이전 레벨의 해시 테이블에 포함된 마지막으로 검색된 출력 페이지로 상기 콘텐츠 이름에 대응하는 콘텐츠 요청 패킷을 전송하는 단계를 더 포함할 수 있다.

[0014] 본 발명의 일실시예에 따른 이름 검색 방법은 NPT의 레벨들 별로 생성된 블록 필터로 콘텐츠 이름을 검색하는 단계; 및 상기 블록 필터의 검색 결과에 기초하여 상기 블록 필터에 대응하는 해시 테이블에서 상기 콘텐츠 이름을 검색하는 단계를 포함할 수 있다.

[0015] 본 발명의 일실시예에 따른 이름 검색 방법의 블록 필터로 콘텐츠 이름을 검색하는 단계는, NPT의 현재 레벨의 블록 필터에 대응하는 해시 테이블에서 상기 콘텐츠 이름이 검색되는 경우, NPT의 다음 레벨의 블록 필터로 콘텐츠 이름을 검색할 수 있다.

[0016] 본 발명의 일실시예에 따른 이름 검색 방법의 해시 테이블에서 상기 콘텐츠 이름을 검색하는 단계는 상기 블록 필터의 검색 결과가 양성인 경우, 상기 블록 필터에 대응하는 해시 테이블에서 상기 콘텐츠 이름을 검색하는 단

계; 및 상기 블록 필터의 검색 결과가 음성이거나, 상기 해시 테이블에서 상기 콘텐츠 이름이 검색되지 않는 경우, 검색을 종료하는 단계를 포함할 수 있다.

[0017] 본 발명의 일실시예에 따른 이름 검색 방법의 검색을 종료하는 단계는 콘텐츠 이름이 마지막으로 검색된 해시 테이블에 포함된 출력 페이스로 상기 콘텐츠 이름에 대응하는 콘텐츠 요청 패킷을 전송할 수 있다.

[0018] 본 발명의 일실시예에 따른 이름 검색 방법의 블록 필터는 상기 이전 레벨에 대응하는 콘텐츠 이름의 컴포넌트와 현재 레벨에 대응하는 컴포넌트가 연속된 문자열을 포함할 수 있다.

[0019] 본 발명의 일실시예에 따른 이름 검색 방법은 NPT의 레벨들 별로 생성된 블록 필터들에서 콘텐츠 이름을 연쇄적으로 검색하는 단계; 및 상기 블록 필터들의 검색 결과에 따른 해시 테이블에서 상기 콘텐츠 이름을 검색하는 단계를 포함할 수 있다.

[0020] 본 발명의 일실시예에 따른 이름 검색 방법의 콘텐츠 이름을 연쇄적으로 검색하는 단계는, NPT의 현재 레벨에 대응하는 블록 필터로 콘텐츠 이름에 포함된 컴포넌트를 검색하는 단계; 상기 블록 필터의 검색 결과가 양성인 경우, NPT의 다음 레벨의 블록 필터로 콘텐츠 이름에 포함된 컴포넌트를 검색하는 단계; 및 상기 블록 필터의 검색 결과가 음성이거나, 현재 레벨이 NPT의 마지막 레벨인 경우, 콘텐츠 이름의 연쇄 검색을 종료하는 단계를 포함할 수 있다.

[0021] 본 발명의 일실시예에 따른 이름 검색 방법의 해시 테이블에서 상기 콘텐츠 이름을 검색하는 단계는, 상기 블록 필터의 검색 결과가 양성인 경우, 이전 레벨의 블록 필터에 대응하는 해시 테이블에서 상기 콘텐츠 이름을 검색할 수 있다.

[0022] 본 발명의 일실시예에 따른 이름 검색 방법의 해시 테이블에서 상기 콘텐츠 이름을 검색하는 단계는, 블록 필터의 검색 결과가 양성이며, 현재 레벨이 NPT의 마지막 레벨인 경우, 현재 레벨의 블록 필터에 대응하는 해시 테이블에서 상기 콘텐츠 이름을 검색할 수 있다.

[0023] 본 발명의 일실시예에 따른 이름 검색 방법의 해시 테이블에서 상기 콘텐츠 이름을 검색하는 단계는, 블록 필터들의 검색 결과에 따른 해시 테이블에서 콘텐츠 이름이 검색되지 않는 경우, 이전 레벨의 블록 필터에 대응하는 해시 테이블에서 상기 콘텐츠 이름을 검색할 수 있다.

발명의 효과

[0024] 본 발명의 일실시예에 의하면, 해시 테이블을 이용하여 NPT에 기초한 콘텐츠 이름을 검색할 수 있다.

[0025] 또한, 본 발명의 일실시예에 의하면, 블록 필터의 검색 결과에 따라 해시 테이블의 검색 여부를 결정함으로써, 해시 테이블의 접근 회수를 감소시켜 검색 속도를 향상시킬 수 있다.

[0026] 그리고, 본 발명의 일실시예에 의하면, NPT의 레벨 별로 생성된 블록 필터를 연속으로 검색하여 최종 레벨을 검색하고, 최종 레벨의 해시 테이블을 검색함으로써, 해시 테이블의 접근 회수를 최소화시킬 수도 있다.

도면의 간단한 설명

[0027] 도 1은 본 발명의 일실시예에 따른 이름 검색 시스템을 나타내는 도면이다.

도 2는 본 발명의 일실시예에 따른 NPT의 일례이다.

도 3은 본 발명의 일실시예에 따른 이름 검색 정보 구축 과정에서 사용하는 코드의 일례이다.

도 4는 본 발명의 일실시예에 따른 블록 필터 정보 구축 과정에서 사용하는 의사 코드의 일례이다.

도 5는 본 발명의 제1 실시예에 따른 이름 검색 장치를 나타내는 도면이다.

도 6은 본 발명의 제1 실시예에 따라 구축한 이름 검색 정보의 일례이다.

도 7은 본 발명의 제1 실시예에 따른 이름 검색 과정에서 사용하는 의사 코드의 일례이다.

도 8은 본 발명의 제2 실시예에 따른 이름 검색 장치를 나타내는 도면이다.

도 9는 본 발명의 제2 실시예에 따라 구축한 이름 검색 정보의 일례이다.

도 10은 본 발명의 일실시예에 따른 블록 필터가 콘텐츠 이름 검색 과정에서 사용하는 코드의 일례이다.

- 도 11은 본 발명의 제2 실시예에 따른 이름 검색 과정에서 사용하는 의사 코드의 일례이다.
- 도 12는 본 발명의 제3 실시예에 따른 이름 검색 장치를 나타내는 도면이다.
- 도 13은 본 발명의 제3 실시예에 따른 이름 검색 과정에서 사용하는 의사 코드의 일례이다.
- 도 14는 본 발명의 일실시예에 따른 이름 검색 과정에서 해시 테이블에 접근한 회수의 일례이다.
- 도 15는 본 발명의 일실시예에 따른 이름 검색 과정에서 해시 테이블의 메모리 요구량의 일례이다.
- 도 16은 본 발명의 제1 실시예에 따른 이름 검색 방법을 도시한 플로우차트이다.
- 도 17은 본 발명의 제2 실시예에 따른 이름 검색 방법을 도시한 플로우차트이다.
- 도 18은 본 발명의 제3 실시예에 따른 이름 검색 방법을 도시한 플로우차트이다.

발명을 실시하기 위한 구체적인 내용

- [0028] 이하, 본 발명의 실시예를 첨부된 도면을 참조하여 상세하게 설명한다. 본 발명의 일실시예에 따른 이름 검색 방법은 이름 검색 장치에 의해 수행될 수 있다.
- [0029] 도 1은 본 발명의 일실시예에 따른 이름 검색 시스템을 나타내는 도면이다.
- [0030] 도 1을 참고하면, 본 발명의 일실시예에 따른 이름 검색 시스템은 이름 검색 정보 구축 장치(110) 및 이름 검색 장치(120)를 포함할 수 있다. 이때, 이름 검색 시스템은 콘텐츠 중심 네트워킹 기술에서 콘텐츠가 저장된 노드로 요청 패킷을 전달하기 위한 출력 페이지를 검색하기 위하여 사용될 수 있다.
- [0031] 이름 검색 정보 구축 장치(110)는 복수의 레벨로 구성된 이름 프리픽스 트리(Name Prefix Trie: NPT)에 따라 콘텐츠 이름을 해시 테이블, 또는 블룸 필터에 저장하여 이름 검색 정보를 구축할 수 있다.
- [0032] 도 1을 참고하면, 본 발명의 일실시예에 따른 이름 검색 정보 구축 장치(110)는 해시 테이블 생성부(111) 및 블룸 필터 생성부(112)를 포함할 수 있다.
- [0033] 해시 테이블 생성부(111)는 NPT의 레벨 별로 해시 테이블을 생성할 수 있다. 그리고, 해시 테이블 생성부(111)는 NPT의 레벨들 각각에 대응하는 콘텐츠 이름의 컴포넌트를 해시 함수로 변환하여 해시 테이블에 저장할 수 있다.
- [0034] 이때, 해시 테이블 생성부(111)는 NPT의 레벨이 두 번째 레벨 이상인 경우, 이전 레벨에 대응하는 콘텐츠 이름의 컴포넌트와 현재 레벨에 대응하는 콘텐츠 이름의 컴포넌트가 연속된 문자열을 해시 함수로 변환할 수 있다.
- [0035] 예를 들어, 해시 테이블 생성부(111)는 NPT의 첫 번째 레벨에 대응하는 콘텐츠 이름의 첫 번째 컴포넌트를 해시 함수로 변환하여 제1 해시 테이블에 저장할 수 있다. 그리고, 해시 테이블 생성부(111)는 콘텐츠 이름의 첫 번째 컴포넌트와 NPT의 두 번째 레벨에 대응하는 콘텐츠 이름의 두 번째 컴포넌트가 연속된 문자열을 해시 함수로 변환하여 제2 해시 테이블에 저장할 수 있다.
- [0036] 해시 테이블 생성부(111)의 동작은 이하 도 6을 참조하여 상세히 설명한다.
- [0037] 블룸 필터 생성부(112)는 콘텐츠 이름의 컴포넌트가 해시 테이블 생성부(111)가 생성한 해시 테이블에 포함되어 있는지 여부를 나타내는 블룸 필터를 생성할 수 있다.
- [0038] 블룸 필터는 어떤 원소들의 집합이 주어졌을 때 주어진 집합에 속하는 원소들의 존재를 비트-벡터 형식으로 나타내는 정보 구조일 수 있다. 그리고, 블룸 필터는 입력과 일치하는 원소가 검색하는 대상인 집합 내에 존재하는지의 여부를 판단할 수 있다. 이때, 블룸 필터는 해시 함수를 이용하여 획득한 해시 인덱스에 검색한 원소를 저장하지 않고, 해시 인덱스에서 해당하는 비트 값을 1로 하여 원소의 존재를 표시할 수 있다.
- [0039] 이때, 블룸 필터 생성부(112)는 NPT의 레벨들 각각에 대응하는 콘텐츠 이름의 컴포넌트들을 미리 정해진 규칙에 따라 프로그래밍하여 블룸 필터를 생성할 수 있다. 그리고, 블룸 필터는 블룸 필터에 입력되는 정보들을 미리 정해진 규칙에 따라 인덱스로 변환할 수 있다. 또한, 블룸 필터는 인덱스로 변환된 입력 정보가 프로그래밍된 컴포넌트와 일치하는 경우, 해시 테이블에 NPT의 레벨들 각각에 대응하는 콘텐츠 이름의 컴포넌트가 포함된 것으로 판단할 수 있다.
- [0040] 블룸 필터 생성부(112)가 블룸 필터를 생성하는 과정은 이하 도 4를 참조하여 상세히 설명한다.

- [0041] 이름 검색 장치(120)는 이름 검색 정보 구축 장치(110)가 구축한 이름 검색 정보의 해시 테이블, 또는 블록 필터로 콘텐츠 이름을 검색하여 콘텐츠가 저장된 노드로 요청 패킷을 전달하기 위한 출력 페이지를 검색할 수 있다.
- [0042] 이때, 이름 검색 장치(120)는 해시 테이블만 이용하여 콘텐츠 이름을 검색할 수 있다. 이름 검색 장치(120)가 해시 테이블만 이용하여 콘텐츠 이름을 검색하는 구체적인 구성 및 동작은 이하 도 5를 참조하여 상세히 설명한다.
- [0043] 또한, 이름 검색 장치(120)는 블록 필터를 선 필터로 이용하여 해시 테이블이 콘텐츠 이름이 포함되어있는지 여부를 확인할 수 있다. 그리고, 이름 검색 장치(120)는 블록 필터의 확인 결과에 따라 콘텐츠 이름을 검색할 수 있다.
- [0044] 예를 들어, 이름 검색 장치(120)는 블록 필터로 콘텐츠 이름이 포함된 것으로 확인된 해시 테이블에서 콘텐츠 이름을 검색하는 과정을 NPT의 레벨들 별로 콘텐츠 이름의 마지막 컴포넌트가 검색될 때까지 반복할 수 있다. 이름 검색 장치(120)가 블록 필터로 콘텐츠 이름이 포함된 것으로 확인된 해시 테이블에서 콘텐츠 이름을 검색하는 구체적인 구성 및 동작은 이하 도 8을 참조하여 상세히 설명한다.
- [0045] 또한, 이름 검색 장치(120)는 블록 필터로 해시 테이블에 콘텐츠 이름이 포함된 것으로 확인된 경우, 해시 테이블에서 콘텐츠 이름을 검색하는 과정을 생략하고, 하위 레벨에서 블록 필터를 적용함으로써, 블록 필터를 연속으로 적용할 수 있다. 이때, 이름 검색 장치(120)는 블록 필터를 연속으로 적용하여 콘텐츠 이름의 마지막 컴포넌트가 포함된 NPT의 레벨을 먼저 검색하고, 마지막 컴포넌트가 포함된 레벨의 해시 테이블에서 콘텐츠 이름을 검색할 수 있다. 이름 검색 장치(120)가 블록 필터를 연속으로 적용하여 콘텐츠 이름을 검색하는 구체적인 구성 및 동작은 이하 도 12를 참조하여 상세히 설명한다.
- [0046] 도 2는 본 발명의 일실시예에 따른 NPT의 일례이다.
- [0047] 콘텐츠가 저장된 위치를 나타내는 콘텐츠 이름은 복수의 컴포넌트로 구성될 수 있다. 예를 들어, 콘텐츠 이름은 URL에 기초하여 결정될 수 있다. 이때, URL은 URL을 구성하는 각각의 컴포넌트를 슬래시(/) 또는 점(.)으로 구분할 수 있다. 예를 들어, "youtube.com/user/PewDiePie"와 같은 URL은 youtube, com, user, PewDiePie, 4개의 컴포넌트로 구성될 수 있다.
- [0048] 이때, 이름 검색 정보 구축 장치(110)는 URL에서 온점(.)으로 구분되는 컴포넌트는 역방향으로 뒤집고, 슬래시(/) 이후의 컴포넌트들의 순서는 유지함으로써, 이름 검색 정보의 콘텐츠 이름으로 사용할 수 있다. 또한, 콘텐츠 이름을 구성하는 컴포넌트 간 구분은 슬래시(/)를 이용할 수 있다.
- [0049] 예를 들어, 이름 검색 정보 구축 장치(110)는 "youtube.com/user/PewDiePie"와 같은 URL을 기초로 "com/youtube/user/PewDiePie"라는 콘텐츠 이름을 결정할 수 있다.
- [0050] NPT는 트라이 기반의 이름 검색 알고리즘이며, 도 2는 이름 검색 정보 구축 장치(110)가 결정한 10개의 콘텐츠 이름을 이용하여 구성된 NPT의 일례이다.
- [0051] 이때, NPT는 콘텐츠 이름에 포함된 컴포넌트의 수에 따라 최대 레벨이 결정될 수 있다.
- [0052] 예를 들어, 콘텐츠 이름(210)은 구성하는 컴포넌트가 "kr", "or", "visitkorea"의 3개이므로, 콘텐츠 이름(210)의 최대 레벨은 3일 수 있다. 또한, 콘텐츠 이름(220)은 구성하는 컴포넌트가 "com", "youtube", "user", "PewDiePie"의 4개이므로, 콘텐츠 이름(220)의 최대 레벨은 4일 수 있다.
- [0053] 즉, NPT는 콘텐츠 이름에 따라 최대 레벨이 서로 다를 수 있다.
- [0054] 도 3은 본 발명의 일실시예에 따른 이름 검색 정보 구축 과정에서 사용하는 코드의 일례이다.
- [0055] 먼저 해시 테이블 생성부(111)는 NPT의 레벨 별로 해시 테이블을 생성할 수 있다.
- [0056] 다음으로, 해시 테이블 생성부(111)는 도 3의 코드에 도시된 바와 같이 NPT의 레벨 별로 생성된 해시 테이블에 FIB의 엔트리에 해당하는 콘텐츠 이름을 저장할 수 있다. 이때, 해시 테이블 생성부(111)는 첫 번째 레벨 해시 테이블부터 마지막 레벨 해시 테이블까지 순차적으로 콘텐츠 이름을 저장할 수 있다.
- [0057] 구체적으로, 해시 테이블 생성부(111)는 콘텐츠 이름의 첫 번째 컴포넌트로 구성된 문자열을 해시 함수로 변환하여 제1 해시 인덱스를 얻을 수 있다. 그리고, 해시 테이블 생성부(111)는 첫 번째 해시 테이블에서 제1 해시 인덱스에 대응하는 인덱스에 첫 번째 컴포넌트로 구성된 문자열을 저장할 수 있다.

- [0058] 다음으로, 해시 테이블 생성부(111)는 콘텐츠 이름의 첫 번째 컴포넌트와 두 번째 컴포넌트가 연속된(concatenation) 문자열을 해시 함수로 변환하여 제2 해시 인덱스를 얻을 수 있다. 그리고, 해시 테이블 생성부(111)는 두 번째 해시 테이블에서 제2 해시 인덱스에 대응하는 인덱스에 콘텐츠 이름의 첫 번째 컴포넌트와 두 번째 컴포넌트가 연속된 문자열을 저장할 수 있다.
- [0059] 그리고, 해시 테이블 생성부(111)는 상기 과정을 콘텐츠 이름의 첫 번째 컴포넌트에서 마지막 컴포넌트까지 연속된 문자열이 해시 테이블에 저장될 때까지 수행할 수 있다. 또한, 해시 테이블 생성부(111)는 콘텐츠 이름의 마지막 컴포넌트가 해시 테이블에 저장될 때, 출력 페이스(face)의 정보를 함께 저장할 수 있다.
- [0060] 그리고, 해시 테이블 생성부(111)는 콘텐츠 이름들 각각을 도 3의 코드에 입력하여 콘텐츠 이름에 대응하는 해시 테이블들을 생성할 수 있다.
- [0061] 도 4는 본 발명의 일실시예에 따른 블록 필터 정보 구축 과정에서 사용하는 의사 코드의 일례이다.
- [0062] 먼저, 블록 필터 생성부(112)는 블록 필터의 모든 색인의 비트-벡터 값을 0으로 초기화할 수 있다.
- [0063] 다음으로, 블록 필터 생성부(112)는 도 4의 코드에 도시된 바와 같이 문자열에 포함된 컴포넌트 x의 정보를 해시 함수에 입력하여 블록 필터를 프로그래밍할 위치를 나타내는 색인(for $i = 1, \dots, k$)를 출력할 수 있다. 이때, 미리 프로그래밍된 컴포넌트와 동일한 색인이 출력되는 경우, 블록 필터 생성부(112)는 해당 색인의 비트-벡터 값을 1로 유지할 수 있다.
- [0064] 그리고, 블록 필터 생성부(112)는 문자열에 포함된 모든 컴포넌트에 대해서, 해당 색인에 대응되는 비트-벡터 값을 1로 지정하여 블록 필터를 생성할 수 있다.
- [0065] 도 5는 본 발명의 제1 실시예에 따른 이름 검색 장치를 나타내는 도면이다.
- [0066] 본 발명의 제1 실시예에 따른 이름 검색 장치(120)는 해시 테이블만 이용하여 콘텐츠 이름을 검색하는 해싱 기반-NPT 알고리즘을 수행할 수 있다.
- [0067] 도 5를 참고하면, 본 발명의 제1 실시예에 따른 이름 검색 장치(120)는 콘텐츠 이름 변환부(510), 및 해시 테이블 검색부(520)를 포함할 수 있다.
- [0068] 콘텐츠 이름 변환부(510)는 NPT의 레벨들 각각에 대응하는 콘텐츠 이름의 컴포넌트를 해시 함수로 변환할 수 있다.
- [0069] 이때, 콘텐츠 이름 변환부(510)는 해시 테이블 생성부(111)가 NPT의 레벨들 별로 해시 테이블을 생성하는 과정에서 해시 인덱스의 생성에 이용된 해시 함수를 이용하여 NPT의 레벨들 각각에 대응하는 콘텐츠 이름의 컴포넌트를 변환할 수 있다.
- [0070] 또한, NPT의 레벨이 두 번째 레벨 이상인 경우, 콘텐츠 이름 변환부(510)는 이전 레벨에 대응하는 콘텐츠 이름의 컴포넌트와 현재 레벨에 대응하는 콘텐츠 이름의 컴포넌트가 연속된 문자열을 해시 함수로 변환할 수 있다.
- [0071] 해시 테이블 검색부(520)는 NPT의 레벨들 별로 생성된 해시 테이블에서 콘텐츠 이름 변환부(510)가 해시 함수로 변환한 컴포넌트를 검색할 수 있다. 이때, 해시 테이블 검색부(520)는 현재 레벨에 대응하는 해시 테이블에서 이전 레벨에 대응하는 콘텐츠 이름의 컴포넌트와 현재 레벨에 대응하는 콘텐츠 이름의 컴포넌트가 연속된 문자열이 해시 함수로 변환된 해시 인덱스를 검색할 수 있다.
- [0072] 또한, 해시 테이블 검색부(520)는 현재 레벨의 해시 테이블에서 컴포넌트가 검색되는 경우, 다음 레벨의 해시 테이블에서 컴포넌트를 검색할 수 있다.
- [0073] 이때, 해시 테이블 검색부(520)는 현재 레벨의 해시 테이블에서 컴포넌트가 검색되지 않거나, 현재 레벨이 콘텐츠 이름에 대응하는 NPT의 마지막 레벨이 될 때까지 상기 과정을 반복할 수 있다.
- [0074] 그리고, 현재 레벨에 대응하는 해시 테이블에서 상기 해시 함수로 변환된 컴포넌트가 검색되지 않는 경우, 해시 테이블 검색부(520)는 이전 레벨의 해시 테이블에 포함된 출력 페이스 중에서 마지막으로 검색된 출력 페이스로 상기 콘텐츠 이름에 대응하는 콘텐츠 요청 패킷을 전송할 수 있다.
- [0075] 도 6은 본 발명의 제1 실시예에 따라 구축한 이름 검색 정보의 일례이다.
- [0076] 이름 검색 정보 구축 장치(110)의 해시 테이블 생성부(111)는 NPT(610)의 레벨 별로 해시 테이블(620)을 생성할 수 있다.

- [0077] 다음으로, 해시 테이블 생성부(111)는 NPT의 레벨 별로 생성된 해시 테이블(620)에 콘텐츠 이름을 저장할 수 있다.
- [0078] 구체적으로, 해시 테이블 생성부(111)는 콘텐츠 이름의 첫 번째 컴포넌트인 "com"으로 구성된 문자열을 해시 함수로 변환하여 제1 해시 인덱스를 얻을 수 있다. 그리고, 해시 테이블 생성부(111)는 첫 번째 해시 테이블(621)에서 제1 해시 인덱스에 대응하는 인덱스에 "com"으로 구성된 문자열을 저장할 수 있다.
- [0079] 다음으로, 해시 테이블 생성부(111)는 콘텐츠 이름의 첫 번째 컴포넌트인 "com"과 두 번째 컴포넌트인 "facebook"가 연속된 문자열인 "com/facebook"을 해시 함수로 변환하여 제2 해시 인덱스를 얻을 수 있다. 그리고, 해시 테이블 생성부(111)는 두 번째 해시 테이블(622)에서 제2 해시 인덱스에 대응하는 인덱스에 문자열 "com/facebook"을 저장할 수 있다. 이때, 문자열 "com/facebook"와 관련된 콘텐츠 이름(611)는 도 6에 도시된 바와 같이 2레벨이 NPT의 마지막 레벨일 수 있다. 따라서, 해시 테이블 생성부(111)는 문자열 "com/facebook"과 관련된 이름 검색 정보의 구축을 종료할 수 있다.
- [0080] 또한, 콘텐츠 이름(612)이 "com/youtube/user/PewDiePie"인 경우, 콘텐츠 이름(612)은 4개의 컴포넌트로 구성될 수 있다. 따라서, 해당 콘텐츠는 도 6에 도시된 바와 같이 4레벨이 NPT의 마지막 레벨일 수 있다. 그리고, 해시 테이블 생성부(111)는 도 6에 도시된 바와 같이 네 번째 해시 테이블(623)에서 제4 해시 인덱스에 대응하는 인덱스에 문자열 "com/youtube/user/PewDiePie"를 저장할 수 있다.
- [0081] 도 7은 본 발명의 제1 실시예에 따른 이름 검색 과정에서 사용하는 의사 코드의 일례이다.
- [0082] 해시 테이블 검색부(520)는 도 7의 코드에 도시된 바와 같이 수신한 정보 요청 패킷에 따라 패킷의 콘텐츠 이름의 첫 번째 컴포넌트를 사용하여 첫 번째 레벨부터 해시 테이블 검색을 시작할 수 있다.
- [0083] 첫 번째 레벨의 해시 테이블에 정보 요청 패킷과 일치하는 콘텐츠 이름의 컴포넌트가 존재하는 경우, 해시 테이블 검색부(520)는 두 번째 레벨의 해시 테이블에서 콘텐츠 이름을 검색할 수 있다. 이때, 해시 테이블 검색부(520)가 두 번째 레벨의 해시 테이블에서 검색하는 이름은 콘텐츠의 이름에 포함된 처음 두 개 컴포넌트가 연속된 문자열일 수 있다.
- [0084] 그리고, 해시 테이블 검색부(520)는 상기 과정을 해시 테이블에서 일치하는 문자열을 발견하지 못했거나 마지막 레벨에 도달할 때까지 반복할 수 있다.
- [0085] 또한, FIB의 콘텐츠 이름 검색 방법은 IP 주소 검색에서와 마찬가지로 최장 길이 일치(Longest Prefix Matching) 룰이 적용될 수 있다. 따라서, 해시 테이블에서 검색된 인덱스에 인터레스트(Interest) 패킷을 내보내야 할 페이지가 저장된 경우, 해시 테이블 검색부(520)는 해시 테이블에 저장된 페이지를 저장할 수 있다. 그리고, NPT는 이전 레벨의 문자열이 다음 레벨의 문자열을 포함하는 특성을 가지고 있으므로, 현재 레벨의 해시 테이블에서 콘텐츠 이름과 일치하는 문자열을 검색하지 못한 경우, NPT의 다음 레벨에서도 일치하는 문자열이 검색되지 않는다. 따라서, 현재 레벨의 해시 테이블에서 콘텐츠 이름과 일치하는 문자열을 검색하지 못한 경우, 해시 테이블 검색부(520)는 검색을 종료하고 가장 최근에 저장된 페이지로 정보 요청 패킷을 전송할 수 있다.
- [0086] 도 8은 본 발명의 제2 실시예에 따른 이름 검색 장치를 나타내는 도면이다.
- [0087] 본 발명의 제2 실시예에 따른 이름 검색 장치(120)는 bloom 필터로 콘텐츠 이름이 포함된 것으로 확인된 해시 테이블에서 콘텐츠 이름을 검색하는 NPT-BF(Bloom Filter) 알고리즘을 수행할 수 있다.
- [0088] 도 8을 참고하면, 본 발명의 제2 실시예에 따른 이름 검색 장치(120)는 콘텐츠 이름 변환부(810), bloom 필터 검색부(820) 및 해시 테이블 검색부(830)를 포함할 수 있다.
- [0089] 콘텐츠 이름 변환부(810)는 해시 함수를 사용하여 NPT의 레벨들 각각에 대응하는 콘텐츠 이름의 컴포넌트를 변환할 수 있다.
- [0090] 이때, 콘텐츠 이름 변환부(810)는 해시 테이블 생성부(111)가 NPT의 레벨들 별로 해시 테이블을 생성하는 과정에서 해시 인덱스의 생성에 이용된 해시 함수 및 bloom 필터 생성부(112)가 NPT의 레벨들 별로 해시 테이블에 대응하는 bloom 필터를 생성하는 과정에서 bloom 필터의 생성에 이용된 해시 함수를 이용하여 NPT의 레벨들 각각에 대응하는 콘텐츠 이름의 컴포넌트를 변환할 수 있다.
- [0091] bloom 필터 검색부(820)는 NPT의 레벨들 별로 생성된 bloom 필터로 콘텐츠 이름을 검색할 수 있다. 이때, bloom 필터는 이전 레벨에 대응하는 콘텐츠 이름의 컴포넌트와 현재 레벨에 대응하는 컴포넌트가 연속된 문자열을 포

함할 수 있다.

- [0092] 해시 테이블 검색부(830)는 블록 필터 검색부(820)의 검색 결과에 기초하여 블록 필터에 대응하는 해시 테이블에서 콘텐츠 이름을 검색할 수 있다.
- [0093] 블록 필터의 검색 결과가 양성인 경우, 해시 테이블 검색부(830)는 블록 필터에 대응하는 해시 테이블에서 콘텐츠 이름을 검색할 수 있다.
- [0094] 또한, 블록 필터의 검색 결과가 음성이거나, 해시 테이블에서 콘텐츠 이름이 검색되지 않는 경우, 해시 테이블 검색부(830)는 콘텐츠 이름의 검색을 종료할 수 있다. 이때, 해시 테이블 검색부(830)는 콘텐츠 이름이 마지막으로 검색된 해시 테이블에 포함된 출력 페이스로 콘텐츠 이름에 대응하는 콘텐츠 요청 패킷을 전송할 수 있다.
- [0095] 그리고, 해시 테이블 검색부(830)가 해시 테이블에서 콘텐츠 이름을 검색 성공한 경우, 블록 필터 검색부(820)는 NPT의 다음 레벨의 블록 필터로 콘텐츠 이름을 검색할 수 있다.
- [0096] 본 발명의 제2 실시예에 따른 이름 검색 장치(120)는 블록 필터를 선 필터로 사용하여 해시 테이블에 콘텐츠 이름이 포함될 가능성을 식별할 수 있다. 그리고, 이름 검색 장치(120)는 해시 테이블에 콘텐츠 이름이 포함될 가능성이 높은 경우에만 해시 테이블에서 콘텐츠 이름을 검색함으로써, 콘텐츠 이름이 저장되지 않은 해시 테이블을 검색하기 위한 시간, 및 처리 능력 낭비를 최소화할 수 있다.
- [0097] 도 9는 본 발명의 제2 실시예에 따라 구축한 이름 검색 정보의 일례이다.
- [0098] 이름 검색 정보 구축 장치(110)의 해시 테이블 생성부(111)는 NPT(920)의 레벨 별로 해시 테이블(930)을 생성할 수 있다.
- [0099] 다음으로, 해시 테이블 생성부(111)는 NPT의 레벨 별로 생성된 해시 테이블(930)에 콘텐츠 이름을 저장할 수 있다. 이때, 해시 테이블 생성부(111)는 NPT의 레벨들 각각에 대응하는 콘텐츠 이름의 컴포넌트를 해시 함수로 변환하여 해시 테이블에 저장할 수 있다. 또한, 해시 테이블 생성부(111)는 NPT의 레벨이 두 번째 레벨 이상인 경우, 이전 레벨에 대응하는 콘텐츠 이름의 컴포넌트와 현재 레벨에 대응하는 콘텐츠 이름의 컴포넌트가 연속된 문자열을 해시 함수로 변환할 수 있다.
- [0100] 또한, 블록 필터 생성부(111)는 NPT의 레벨 및 해시 테이블(930)에 대응하는 블록 필터(910)를 생성할 수 있다. 이때, 블록 필터 생성부(111)는 해시 테이블(930)에 저장된 문자열을 미리 정해진 해시 함수, 또는 해시 테이블 생성부(111)가 사용한 해시 함수로 프로그래밍하여 블록 필터를 생성할 수 있다.
- [0101] 도 10은 본 발명의 일 실시예에 따른 블록 필터가 콘텐츠 이름 검색 과정에서 사용하는 코드의 일례이다.
- [0102] 블록 필터 검색부(820)는 NPT의 레벨들 별로 생성된 블록 필터로 콘텐츠 이름을 검색할 수 있다.
- [0103] 이때, 블록 필터 검색부(820)는 도 10의 코드에 도시된 바와 같이 입력과 블록 필터에 저장된 값을 비교하는 쿼리 과정을 진행할 수 있다. 이때, 블록 필터의 입력은 콘텐츠 이름 변환부(810)가 콘텐츠 이름을 변환하여 생성한 블록 필터 색인일 수 있다.
- [0104] 콘텐츠 이름 변환부(810)는 블록 필터 생성부(112)가 블록 필터 프로그래밍 과정에서 사용했던 해시 함수와 동일한 해시 함수로 콘텐츠 이름을 변환하여 블록 필터 색인을 출력할 수 있다. 또한, 콘텐츠 이름 변환부(810)가 출력하는 블록 필터 색인의 개수는 블록 필터 생성부(112)가 블록 필터를 프로그래밍 될 때의 색인 개수와 동일하다.
- [0105] 그리고, 블록 필터 검색부(820)는 블록 필터에서 콘텐츠 이름 변환부(810)가 출력하는 블록 필터 색인의 위치에 해당하는 비트-벡터 값을 확인할 수 있다.
- [0106] 이때, 블록 필터의 입력에 따라 블록 필터 검색부(820)가 확인한 모든 색인의 비트-벡터 값이 1인 경우, 블록 필터의 입력은 블록 필터를 통과할 수 있다. 그리고, 블록 필터 검색부(820)는 블록 필터의 검색 결과를 양성이라고 한다.
- [0107] 또한, 블록 필터의 입력에 따라 블록 필터 검색부(820)가 확인한 색인의 비트-벡터 값 중 적어도 하나의 0이 포함된 경우, 블록 필터의 입력은 블록 필터를 통과할 수 없다. 그리고, 블록 필터 검색부(820)는 블록 필터의 검색 결과를 음성이라고 한다.
- [0108] 도 11은 본 발명의 제2 실시예에 따른 이름 검색 과정에서 사용하는 의사 코드의 일례이다.

- [0109] 블록 필터 검색부(820)는 도 11의 코드에 도시된 바와 같이 블록 필터를 먼저 검색할 수 있다. 그리고, 해시 테이블 검색부(830)는 블록 필터의 검색 결과가 양성인 경우에만 해시 테이블에 접근하여 콘텐츠 이름이 해시 테이블에 존재하는지 확인할 수 있다.
- [0110] 콘텐츠 이름이 해시 테이블에 존재하는 경우, 블록 필터 쿼리 결과가 참-양성이므로 블록 필터 검색부(820)는 다음 레벨의 블록 필터를 쿼리할 수 있다.
- [0111] 그러나, 콘텐츠 이름이 해시 테이블에 존재하지 않는 경우, 블록 필터 쿼리 결과가 음성이거나 거짓-양성이므로 블록 필터 검색부(820)는 검색을 종료할 수 있다. 이전 레벨에서 검색하여 저장, 또는 기억한 출력 페이지가 존재하는 경우, 해시 테이블 검색부(830)는 저장한 출력 페이지를 출력할 수 있다.
- [0112] 즉, 이름 검색 장치(120)는 해시 테이블에 콘텐츠 이름이 포함될 가능성이 없는 경우, 해당 해시 테이블에 접근하지 않고 검색을 종료할 수 있다. 따라서, 이름 검색 장치(120)는 해시 테이블 접근회수를 감소시킴으로써 검색 성능을 향상시킬 수 있다.
- [0113] 도 12는 본 발명의 제3 실시예에 따른 이름 검색 장치를 나타내는 도면이다.
- [0114] 본 발명의 제3 실시예에 따른 이름 검색 장치(120)는 블록 필터를 연속으로 적용하여 콘텐츠 이름을 검색하는 NPT-BF-Chaining 알고리즘을 수행할 수 있다. 이때, NPT-BF-Chaining 알고리즘은 블록 필터의 거짓양성의 확률이 매우 적다는 점을 이용한 알고리즘이다.
- [0115] 구체적으로 블록 필터가 양성을 출력한 경우, 이름 검색 장치(120)는 블록 필터의 결과를 잠정적으로 참-양성이라 판단하고, 다음 레벨로 블록 필터 쿼리를 계속 진행할 수 있다.
- [0116] 즉, 이름 검색 장치(120)는 블록 필터 검색을 통하여 최장 길이 일치 콘텐츠 이름이 있는 레벨을 먼저 식별할 수 있다. 그리고, 이름 검색 장치(120)는 식별한 레벨에 해당하는 해시 테이블에서 콘텐츠 이름을 검색함으로써, 빠르게 콘텐츠 이름을 검색할 수 있다. 또한, 식별한 레벨에 해당하는 해시 테이블에 콘텐츠 이름이 존재하지 않는 경우, 백-트래킹(back-tracking)을 통하여 이전 레벨로 검색을 진행할 수 있다.
- [0117] 도 12를 참고하면, 본 발명의 제3 실시예에 따른 이름 검색 장치(120)는 콘텐츠 이름 변환부(1210), 블록 필터 검색부(1220) 및 해시 테이블 검색부(1230)를 포함할 수 있다.
- [0118] 콘텐츠 이름 변환부(1210)는 NPT의 레벨들 각각에 대응하는 콘텐츠 이름의 컴포넌트를 해시 함수로 변환할 수 있다.
- [0119] 이때, 콘텐츠 이름 변환부(1210)는 해시 테이블 생성부(111)가 NPT의 레벨들 별로 해시 테이블을 생성하는 과정에서 해시 인덱스의 생성에 이용된 해시 함수 및 블록 필터 생성부(112)가 NPT의 레벨들 별로 해시 테이블에 대응하는 블록 필터를 생성하는 과정에서 블록 필터의 생성에 이용된 해시 함수를 이용하여 NPT의 레벨들 각각에 대응하는 콘텐츠 이름의 컴포넌트를 변환할 수 있다.
- [0120] 블록 필터 검색부(1220)는 NPT의 레벨들 별로 생성된 블록 필터들에서 콘텐츠 이름을 연쇄적으로 검색할 수 있다. 이때, 블록 필터는 이전 레벨에 대응하는 콘텐츠 이름의 컴포넌트와 현재 레벨에 대응하는 컴포넌트가 연속된 문자열을 포함할 수 있다.
- [0121] 또한, 블록 필터 검색부(1220)는 NPT의 현재 레벨에 대응하는 블록 필터로 콘텐츠 이름에 포함된 컴포넌트를 검색할 수 있다. 이때, 블록 필터의 검색 결과가 양성인 경우, 블록 필터 검색부(1220)는 NPT의 다음 레벨의 블록 필터로 콘텐츠 이름에 포함된 컴포넌트를 검색할 수 있다.
- [0122] 그리고, 블록 필터의 검색 결과가 음성이거나, 현재 레벨이 NPT의 마지막 레벨인 경우, 블록 필터 검색부(1220)는 콘텐츠 이름의 연쇄 검색을 종료할 수 있다.
- [0123] 해시 테이블 검색부(1230)는 블록 필터 검색부(1220)의 검색 결과에 따른 해시 테이블에서 콘텐츠 이름을 검색할 수 있다.
- [0124] 블록 필터 검색부(1220)의 블록 필터의 검색 결과가 음성인 경우, 해시 테이블 검색부(1230)는 이전 레벨의 블록 필터에 대응하는 해시 테이블에서 콘텐츠 이름을 검색할 수 있다.
- [0125] 또한, 현재 레벨이 NPT의 마지막 레벨인 경우, 해시 테이블 검색부(1230)는 현재 레벨의 블록 필터에 대응하는 해시 테이블에서 콘텐츠 이름을 검색할 수 있다. 이때, 현재 레벨의 블록 필터에 대응하는 해시 테이블이 블록 필터들의 검색 결과에 따른 해시 테이블일 수 있다. 그리고, 현재 레벨의 블록 필터에 대응하는 해시 테이블에

서 콘텐츠 이름이 검색되지 않는 경우, 해시 테이블 검색부(1230)는 백-트래킹(back-tracking)을 통하여 이전 레벨의 블록 필터에 대응하는 해시 테이블에서 콘텐츠 이름을 검색할 수 있다.

[0126] 즉, 본 발명의 제3 실시예에 따른 이름 검색 장치(120)는 NPT의 레벨들 별로 생성된 블록 필터들에서 콘텐츠 이름을 연쇄적으로 검색하여 음성의 결과가 나온 경우 혹은 NPT의 마지막 레벨인 경우와 같이 최장 길이 일치 콘텐츠 이름이 있는 레벨을 먼저 식별할 수 있다. 그리고, 이름 검색 장치(120)는 식별한 레벨에 해당하는 해시 테이블에서 콘텐츠 이름을 검색함으로써, 빠르게 콘텐츠 이름을 검색할 수 있다.

[0127] 도 13은 본 발명의 제3 실시예에 따른 이름 검색 과정에서 사용하는 의사 코드의 일례이다.

[0128] 블록 필터 검색부(1220)는 도 13의 코드에 도시된 바와 같이 블록 필터를 먼저 검색할 수 있다. 그리고, 블록 필터의 검색 결과가 양성인 경우, 블록 필터 검색부(1220)는 다음 레벨의 블록 필터를 연쇄적으로 검색할 수 있다.

[0129] 이때, 블록 필터 검색부(1220)는 블록 필터의 검색 결과로 음성이 출력되거나, NPT의 마지막 레벨의 블록 필터를 검색할 때까지 연쇄적으로 다음 레벨 블록 필터를 검색할 수 있다.

[0130] 블록 필터의 검색 결과로 음성이 출력되었다는 것은 콘텐츠 이름과 일치하는 컴포넌트들의 연속된 문자열이 현재 레벨 및 다음 레벨들의 해시 테이블에 존재하지 않는다는 의미이다. 따라서, 블록 필터의 검색 결과로 음성이 출력되는 경우, 해시 테이블 검색부(1230)는 이전 레벨의 해시 테이블에서 콘텐츠 이름을 검색할 수 있다.

[0131] 이전 레벨의 해시 테이블에 콘텐츠 이름이 존재하는 경우, 이전 레벨의 블록 필터 검색 결과가 참-양성이라는 의미이다. 따라서, 해시 테이블 검색부(1230)는 출력 페이스를 반환하고 검색을 종료할 수 있다.

[0132] 그러나, 이전 레벨의 해시 테이블에 콘텐츠 이름이 존재하지 않는 경우, 이전 레벨의 블록 필터 검색 결과가 거짓-양성이라는 의미이다. 따라서, 해시 테이블 검색부(1230)는 콘텐츠 이름이 검색될 때까지 이전 레벨의 해시 테이블을 연쇄적으로 재 검색하는 백-트래킹을 수행할 수 있다.

[0133] 본 발명의 제3 실시예에 따른 이름 검색 장치(120)는 레벨 별로 생성된 블록 필터를 연쇄적으로 검색하여 콘텐츠 이름이 존재할 가능성이 있는 최종 레벨을 알아낼 수 있다. 따라서, 본 발명의 제3 실시예에 따른 이름 검색 장치(120)는 백-트래킹이 발생하지 않는 경우, 하나의 입력 당 한 번의 해시 테이블에 접근하여 검색을 완료할 수 있다.

[0134] 도 14는 본 발명의 일 실시예에 따른 이름 검색 과정에서 해시 테이블에 접근한 회수의 일례이다.

[0135] 도 14는 CCN 콘텐츠 이름과 같이 계층적 특성을 가지는 URL을 이용하여 본 발명의 제1 실시예, 제2 실시예 및 제3 실시예의 성능 평가를 수행한 결과의 일례이다.

[0136] 이때, 제1 실시예에 따른 이름 검색 장치(120)가 수행하는 해싱 기반-NPT, 제2 실시예에 따른 이름 검색 장치(120)가 수행하는 NPT-BF 알고리즘 및 제3 실시예에 따른 이름 검색 장치(120)가 수행하는 NPT-BF-Chaining 알고리즘은 10,000 개, 50,000 개, 100,000 개, 또는 300,000 개의 콘텐츠 이름을 이용해 시뮬레이션될 수 있다. 이때, 콘텐츠 이름의 개수를 각각 10k, 50k, 100k, 300k로 간단하게 표현될 수 있다.

[0137] 또한, 콘텐츠 이름을 검색할 경우, 각각 30,000 개, 150,000 개, 300,000 개, 또는 900,000 개의 콘텐츠 이름을 입력으로 이용할 수 있다.

[0138] 표 1은 해시 테이블에 저장된 엔트리와 해시 인덱스 충돌로 인해 연결 리스트에 저장된 엔트리들의 정보 구조를 나타내고 있다. 표 2는 각각의 실험에서 level-1부터 level-8까지에 저장된 콘텐츠 이름들의 개수를 나타내고 있다. 표 3은 구축된 트라이의 특성을 보인다.

표 1

Entry	Type	Field
Hash table entry	string	Hash key
	int	Output face
	pointer	Linked list pointer
Linked list entry	string	Hash key
	int	Output face
	pointer	Next linked list pointer

[0139]

표 2

Level	10k	50k	100k	300k
1	0	0	0	0
2	9007	44831	88857	258702
3	944	4805	10185	37189
4	48	341	909	3880
5	0	19	42	207
6	1	4	6	19
7	-	-	1	1
8	-	-	-	2

[0140]

표 3

	10k	50k	100k	300k
No. of names	10000	50000	100000	300000
Maximum depth	6	6	7	8
No. of linked list	185	2975	2478	16420
No. of nodes	10320	50798	101344	303638

[0141]

[0142]

이름 검색 장치(120)는 해시 함수로 CRC-64 생성기를 사용할 수 있다. 이때, CRC 내의 모든 플립-플롭(flip-flop)은 0으로 초기화 되어 있으며 입력 값이 한 비트씩 들어오면서 플립-플롭에 저장된 값과 차례로 XOR할 수 있다. 그리고, 입력 비트열의 마지막 비트까지 들어오면 CRC 생성기의 연산 수행을 종료하고 플립-플롭에 저장되어 있는 값을 CRC 코드로 출력할 수 있다.

[0143]

또한, 블록 필터의 프로그래밍에 사용된 해시 함수의 개수 k는 블록 필터의 크기에 따라서 정해질 수 있다. 만약 컴포넌트 하나당 k개의 해시 함수를 사용했다면 프로그래밍 해야 할 위치 색인을 k개 얻을 수 있다.

[0144]

그리고, 블록 필터에 저장되어야 하는 전체 노드의 개수를 N이라고 할 때 N보다 큰 2의 배수를 블록 필터의 기본크기로 정하면, 블록 필터의 기본크기인 N'는 수학적 1로 결정될 수 있다.

수학적 1

$$N' = 2^{\lceil \log_2 N \rceil}$$

[0145]

[0146]

또한, 기본 블록 필터의 크기에 확장배수를 곱하여 크기가 M인 블록 필터를 사용할 경우, 프로그래밍 되는 색인의 개수도 늘어나야 올바른 성능을 얻을 수 있다. 이때, 프로그래밍 해야 할 색인은 블록 필터의 크기에 비례하여 증가할 수 있다. 예를 들어, 블록 필터의 해시 함수의 개수 k는 수학적 2로 결정될 수 있다.

수학적 2

$$k = \frac{M}{N'} \ln 2$$

[0147]

[0148]

블록 필터의 크기를 기본 크기 N'에 확장배수 4, 8, 16을 곱하여 4N', 8N', 16N'으로 변화시키는 경우, 블록 필터의 크기가 FIB 검색 속도에 어떤 영향을 주는지 확인할 수 있다.

표 4

	해싱기반-NPT	NPT-BF			NPT-BF-Chaining		
BF size	-	4N	8N	16N	4N	8N	16N
BF memory[KB]	-	8.66	17.31	34.63	8.66	17.31	34.63
Total HT access	63233	43771	42908	42860	30789	29950	29903
Avg. HT access	2.11	1.46	1.43	1.43	1.03	0.998	0.997
Linked list access	419	225	199	198	211	185	184
False positive	-	884	47	0	886	47	0
Worst back-tracking	-	-	-	-	2	1	0

[0149]

[0150]

표 4는 10,000개의 콘텐츠 이름을 사용하여 구축하고 구축에 사용된 10,000개의 콘텐츠 이름과 구축에 사용되지 않은 20,000개의 콘텐츠 이름을 추가해 총 30,000개의 콘텐츠 이름으로 검색한 결과일 수 있다.

[0151]

표 4에서 해싱기반-NPT 알고리즘은 평균 2.11번 해시 테이블에 접근하였는데 이는 10,000개의 콘텐츠 이름 중 약 90%가 level-2에 저장되어 있어 level-3에서 일치하는 노드가 없는 경우 검색을 종료할 수 있기 때문이다.

[0152]

표 4에서 NPT-BF 알고리즘은 블록 필터의 크기가 4N', 8N', 16N' 일 때 해시 테이블에 평균적으로 1.46번, 1.43번, 1.43번 접근하여 출력 페이지를 찾았다. 블록 필터의 크기가 클수록 전체 해시 테이블 접근 횟수가 43771번에서 42908번으로, 그리고 42860번으로 감소하였다. 그리고 거짓-양성의 수는 4N' 일 때 884번이었으나 8N' 일 때는 47번으로 감소하고 16N' 일 때는 0번이 된다. 거짓 양성이 줄어든 숫자만큼 해시 테이블 접근 횟수가 함께 감소하였다.

[0153]

반면, 음성의 결과가 나올 때까지 블록 필터 쿼리를 연쇄적으로 진행한 후에 해시 테이블에 접근하는 NPT-BF-Chaining 알고리즘은 평균 0.997 접근하였다. 즉, 대부분 한 번만 해시 테이블에 접근하거나 아예 접근하지 않았다.

[0154]

즉, NPT-BF-Chaining 알고리즘은 선-필터 검색과정을 통해 콘텐츠 이름이 존재할 가능성이 있는 최장 레벨을 알아낼 수 있기 때문에 최장 레벨의 해시 테이블에 접근함으로써 단 한 번의 오프-칩 접근으로 원하는 검색 결과를 찾아낼 수 있었다. 또한 첫 번째 레벨의 블록 필터 검색 결과가 음성인 경우, 해시 테이블에 콘텐츠 이름이 존재하지 않는다는 의미이므로, 해시 테이블에 접근할 필요가 없게 되어 단 한 번도 오프-칩 메모리에 접근하지 않을 수 있다.

[0155]

거짓-양성이 일어나지 않으면 해시 테이블에 한 번만 검색해서 결과를 찾을 수 있지만 거짓-양성이 일어날 때에는 거짓-양성이 일어난 만큼 백-트래킹을 해야 하며 백-트래킹을 많이 할수록 해시 테이블 검색 횟수가 증가해 검색 성능을 저하시킨다.

표 5

	해싱기반-NPT	NPT-BF			NPT-BF-Chaining		
BF size	-	4N	8N	16N	4N	8N	16N
BF memory[KB]	-	36.52	73.04	146.07	36.52	73.04	146.07
Total HT access	320877	225716	219822	219407	167410	161514	161099
Avg. HT access	2.14	1.50	1.47	1.46	1.12	1.08	1.07
Linked list access	2975	1470	1342	1330	1432	1304	1292
False positive	-	6175	409	6	6177	409	6
Worst back-tracking	-	-	-	-	2	1	1

[0156]

표 6

	해싱기반- NPT	NPT-BF			NPT-BF-Chaining		
BF size	-	4N	8N	16N	4N	8N	16N
BF memory[KB]	-	72.79	145.57	291.14	72.79	145.57	291.14
Total HT access	645661	455610	443503	442636	336098	300517	299670
Avg. HT access	2.15	1.52	1.48	1.48	1.12	1.00	0.999
Linked list access	5827	2957	2624	2604	2900	2566	2546
False positive	-	12627	853	6	12639	853	6
Worst back-tracking	-	-	-	-	3	1	1

[0157]

표 7

	해싱기반- NPT	NPT-BF			NPT-BF-Chaining		
BF size	-	4N	8N	16N	4N	8N	16N
BF memory[KB]	-	162.39	324.79	649.57	162.39	324.79	649.57
Total HT access	1962731	1430879	1363877	1353298	1046750	979705	969126
Avg. HT access	2.18	1.59	1.52	1.50	1.16	1.09	1.08
Linked list access	46553	22510	17674	16970	22197	17360	16656
False positive	-	72363	10197	322	72405	10197	322
Worst back-tracking	-	-	-	-	3	1	1

[0158]

[0159]

표 5 내지 표 7은 콘텐츠 이름의 개수가 각각 50k, 100k, 300k인 경우의 실험 결과일 수 있다. 표 5 내지 표 7에 따르면, 거짓-양성이 가장 많이 일어나는 4N' 블록 필터에서도 최대 3번의 백-트래킹이 일어나는 것으로 확인될 수 있다.

[0160]

또한, 블록 필터의 메모리 사용량은 빌드에 사용되는 입력의 개수가 같을 때 NPT-BF 알고리즘과 NPT-BF-Chaining 알고리즘에서 동일하다. 왜냐하면 블록 필터와 해시 테이블은 동일하고 블록 필터와 해시 테이블의 검색 방식만 다르기 때문이다.

[0161]

도 14는 블록 필터 사이즈를 16N으로 고정하고 프로그래밍에 사용된 컴포넌트의 개수를 증가시킬 경우, 해시 테이블에 접근한 총 횟수를 각 알고리즘 별로 나타낸 그래프이다. 도 14에 도시된 바에 따르면, 입력 트레이스가 적을수록 해시 테이블에 접근하는 횟수도 적을 수 있다. 또한 NPT, NPT-BF, NPT-BF-Chaining 알고리즘 순으로 해시 테이블의 접근 횟수가 감소함을 볼 수 있다.

[0162]

도 15는 본 발명의 일실시예에 따른 이름 검색 과정에서 해시 테이블의 메모리 요구량의 일례이다.

[0163]

표 8은 NPT의 레벨 별 해시 테이블의 데이터 구조의 일례이다. 또한, 표 9는 콘텐츠 이름의 개수에 따른 해시 테이블의 메모리 요구량일 수 있다. 그리고, 도 15는 표 9를 그래프로 나타낸 도면일 수 있다. 이때, 해시 테이블의 크기는 빌드에 사용되는 입력의 개수가 같다면 NPT, NPT-BF, NPT-BF-Chaining 알고리즘에서 동일할 수 있다. 그리고, 콘텐츠 이름의 개수에 비례하여 해시 테이블의 크기가 증가할 수 있다.

표 8

Entry width	Field	No. of bits
160 bit	BMN (Best Matching Name)	32
	Linked list pointer	32
	Key string	64
	Output face	32
48 bit	Number of keys	32
	Bit width	16

[0164]

표 9

	최대 해시 테이블 레벨 (L_m)	해시 테이블 엔트리 개수 (N_e)	메모리 요구량 (KB) ($L_m \times N_e \times 160bit + L_m \times 48bit$)
10k	6	35360	692.60
50k	6	149576	2921.44
100k	7	298130	5822.89
300k	8	665164	12991.53

[0165]

[0166]

도 16은 본 발명의 제1 실시예에 따른 이름 검색 방법을 도시한 플로우차트이다.

[0167]

단계(1610)에서 콘텐츠 이름 변환부(510)는 해시 함수를 사용하여 NPT의 레벨들 각각에 대응하는 콘텐츠 이름의 컴포넌트를 변환할 수 있다. 이때, 콘텐츠 이름 변환부(510)는 해시 테이블 생성부(111)가 NPT의 레벨들 별로 해시 테이블을 생성하는 과정에서 해시 인덱스의 생성에 이용된 해시 함수를 이용하여 NPT의 레벨들 각각에 대응하는 콘텐츠 이름의 컴포넌트를 변환할 수 있다. 또한, NPT의 레벨이 두 번째 레벨 이상인 경우, 콘텐츠 이름 변환부(510)는 이전 레벨에 대응하는 콘텐츠 이름의 컴포넌트와 현재 레벨에 대응하는 콘텐츠 이름의 컴포넌트가 연속된 문자열을 해시 함수로 변환할 수 있다.

[0168]

단계(1620)에서 해시 테이블 검색부(520)는 단계(1610)에서 해시 함수로 변환한 컴포넌트를 NPT의 레벨들 별로 생성된 해시 테이블에서 검색할 수 있다.

[0169]

단계(1630)에서 해시 테이블 검색부(520)는 단계(1620)에서 콘텐츠 이름의 컴포넌트가 검색되었는지 여부를 확인할 수 있다. 단계(1620)에서 콘텐츠 이름의 컴포넌트가 검색된 경우, 해시 테이블 검색부(520)는 단계(1640)을 수행할 수 있다. 또한, 단계(1620)에서 콘텐츠 이름의 컴포넌트가 검색되지 않은 경우, 해시 테이블 검색부(520)는 단계(1660)을 수행할 수 있다.

[0170]

단계(1640)에서 해시 테이블 검색부(520)는 현재 레벨이 NPT의 마지막 레벨인지 여부를 확인할 수 있다. 현재 레벨이 NPT의 마지막 레벨인 경우, 해시 테이블 검색부(520)는 단계(1660)을 수행할 수 있다. 또한, 현재 레벨이 NPT의 마지막 레벨이 아닌 경우, 해시 테이블 검색부(520)는 단계(1650)을 수행할 수 있다.

[0171]

단계(1650)에서 해시 테이블 검색부(520)는 NPT의 레벨을 증가시킬 수 있다. 즉, 해시 테이블 검색부(520)는 다음 레벨의 해시 테이블에서 단계(1610) 내지 단계(1640)을 수행할 수 있다.

[0172]

단계(1660)에서 해시 테이블 검색부(520)는 마지막으로 콘텐츠 이름이 검색된 레벨에 따라 콘텐츠 이름을 식별하고, 식별된 콘텐츠 이름에 해당하는 출력 페이스로 요청 패킷을 전달할 수 있다. 이때, 단계(1630)에서 콘텐츠 이름이 검색되지 않은 경우, 마지막으로 콘텐츠 이름이 검색된 이전 레벨의 해시 테이블에 포함된 출력 페이스로 상기 콘텐츠 이름에 대응하는 콘텐츠 요청 패킷을 전송할 수 있다.

[0173]

그리고, 현재 레벨이 NPT의 마지막 레벨인 경우, 마지막으로 콘텐츠 이름이 검색된 레벨은 현재 레벨이고, 해시 테이블 검색부(520)는 현재 레벨의 해시 테이블에서 콘텐츠 이름을 검색하여 식별할 수 있다.

[0174]

도 17은 본 발명의 제2 실시예에 따른 이름 검색 방법을 도시한 플로우차트이다.

- [0175] 단계(1710)에서 콘텐츠 이름 변환부(810)는 해시 함수를 사용하여 NPT의 레벨들 각각에 대응하는 콘텐츠 이름의 컴포넌트를 변환 할 수 있다. 이때, 콘텐츠 이름 변환부(810)는 해시 테이블 생성부(111)가 NPT의 레벨들 별로 해시 테이블을 생성하는 과정에서 해시 인덱스의 생성에 이용된 해시 함수 및 블록 필터 생성부(112)가 NPT의 레벨들 별로 해시 테이블에 대응하는 블록 필터를 생성하는 과정에서 블록 필터의 생성에 이용된 해시 함수를 이용하여 NPT의 레벨들 각각에 대응하는 콘텐츠 이름의 컴포넌트를 변환할 수 있다.
- [0176] 단계(1720)에서 블록 필터 검색부(820)는 NPT의 레벨들 별로 생성된 블록 필터로 단계(1710)에서 변환된 콘텐츠 이름을 검색할 수 있다. 이때, 블록 필터의 출력이 양성인 경우, 블록 필터 검색부(820)는 콘텐츠 이름이 검색되었다고 판단할 수 있다. 또한, 블록 필터의 출력이 음성인 경우, 블록 필터 검색부(820)는 콘텐츠 이름이 검색되지 않았다고 판단할 수 있다.
- [0177] 단계(1730)에서 해시 테이블 검색부(830)는 단계(1720)에서 콘텐츠 이름이 검색되었는지 여부를 확인할 수 있다. 단계(1720)에서 콘텐츠 이름의 컴포넌트가 검색된 경우, 해시 테이블 검색부(830)는 단계(1740)을 수행할 수 있다. 또한, 단계(1720)에서 콘텐츠 이름이 검색되지 않은 경우, 해시 테이블 검색부(830)는 단계(1780)을 수행할 수 있다.
- [0178] 단계(1740)에서 해시 테이블 검색부(830)는 블록 필터에 대응하는 해시 테이블에서 콘텐츠 이름을 검색할 수 있다.
- [0179] 단계(1750)에서 해시 테이블 검색부(830)는 단계(1740)에서 콘텐츠 이름의 컴포넌트가 검색되었는지 여부를 확인할 수 있다. 단계(1740)에서 콘텐츠 이름의 컴포넌트가 검색된 경우, 해시 테이블 검색부(830)는 단계(1760)을 수행할 수 있다. 또한, 단계(1740)에서 콘텐츠 이름의 컴포넌트가 검색되지 않은 경우, 해시 테이블 검색부(830)는 단계(1780)을 수행할 수 있다.
- [0180] 단계(1760)에서 해시 테이블 검색부(830)는 현재 레벨이 NPT의 마지막 레벨인지 여부를 확인할 수 있다. 현재 레벨이 NPT의 마지막 레벨인 경우, 해시 테이블 검색부(830)는 단계(1780)을 수행할 수 있다. 또한, 현재 레벨이 NPT의 마지막 레벨이 아닌 경우, 해시 테이블 검색부(830)는 단계(1770)을 수행할 수 있다.
- [0181] 단계(1770)에서 해시 테이블 검색부(830)는 NPT의 레벨을 증가시킬 수 있다. 즉, 해시 테이블 검색부(830)는 다음 레벨의 해시 테이블에서 단계(1710) 내지 단계(1760)을 수행할 수 있다.
- [0182] 단계(1780)에서 해시 테이블 검색부(830)는 마지막으로 콘텐츠 이름이 검색된 레벨에 따라 콘텐츠 이름을 식별하고, 식별된 콘텐츠 이름에 해당하는 출력 페이스로 요청 패킷을 전달할 수 있다. 이때, 단계(1750)에서 콘텐츠 이름이 검색되지 않은 경우, 마지막으로 콘텐츠 이름이 검색된 이전 레벨의 해시 테이블에 포함된 출력 페이스로 상기 콘텐츠 이름에 대응하는 콘텐츠 요청 패킷을 전송할 수 있다.
- [0183] 그리고, 현재 레벨이 NPT의 마지막 레벨인 경우, 마지막으로 콘텐츠 이름이 검색된 레벨은 현재 레벨이고, 해시 테이블 검색부(830)는 현재 레벨의 해시 테이블에서 콘텐츠 이름을 검색하여 식별할 수 있다.
- [0184] 도 18은 본 발명의 제3 실시예에 따른 이름 검색 방법을 도시한 플로우차트이다.
- [0185] 단계(1810)에서 콘텐츠 이름 변환부(1210)는 해시 함수를 사용하여 NPT의 레벨들 각각에 대응하는 콘텐츠 이름의 컴포넌트를 변환 할 수 있다. 이때, 콘텐츠 이름 변환부(1210)는 해시 테이블 생성부(111)가 NPT의 레벨들 별로 해시 테이블을 생성하는 과정에서 해시 인덱스의 생성에 이용된 해시 함수 및 블록 필터 생성부(112)가 NPT의 레벨들 별로 해시 테이블에 대응하는 블록 필터를 생성하는 과정에서 블록 필터의 생성에 이용된 해시 함수를 이용하여 NPT의 레벨들 각각에 대응하는 콘텐츠 이름의 컴포넌트를 변환할 수 있다.
- [0186] 단계(1820)에서 블록 필터 검색부(1220)는 NPT의 레벨들 별로 생성된 블록 필터로 단계(1810)에서 변환된 콘텐츠 이름을 검색할 수 있다. 이때, 블록 필터의 출력이 양성인 경우, 블록 필터 검색부(1220)는 콘텐츠 이름이 검색되었다고 판단할 수 있다. 또한, 블록 필터의 출력이 음성인 경우, 블록 필터 검색부(1220)는 콘텐츠 이름이 검색되지 않았다고 판단할 수 있다.
- [0187] 단계(1830)에서 블록 필터 검색부(1220)는 단계(1820)에서 콘텐츠 이름이 검색되었는지 여부를 확인할 수 있다. 단계(1820)에서 콘텐츠 이름의 컴포넌트가 검색된 경우, 블록 필터 검색부(1220)는 단계(1840)을 수행할 수 있다. 또한, 단계(1820)에서 콘텐츠 이름이 검색되지 않은 경우, 블록 필터 검색부(1220)는 단계(1835)을 수행할 수 있다.
- [0188] 단계(1835)에서 해시 테이블 검색부(1230)는 NPT의 레벨을 감소시킬 수 있다. 즉, 해시 테이블 검색부(1230)는

이전 레벨의 해시 테이블에서 단계(1860) 내지 단계(1870)을 수행하여 이전 레벨의 해시 테이블 중에서 콘텐츠 이름을 연쇄적으로 검색할 수 있다.

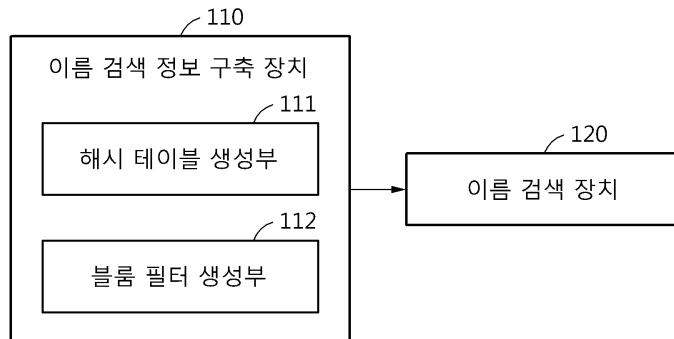
- [0189] 단계(1840)에서 블록 필터 검색부(1220)는 현재 레벨이 NPT의 마지막 레벨인지 여부를 확인할 수 있다. 현재 레벨이 NPT의 마지막 레벨인 경우, 블록 필터 검색부(1220)는 단계(1860)을 수행할 수 있다. 또한, 현재 레벨이 NPT의 마지막 레벨이 아닌 경우, 블록 필터 검색부(1220)는 단계(1850)을 수행할 수 있다.
- [0190] 단계(1850)에서 블록 필터 검색부(1220)는 NPT의 레벨을 증가시킬 수 있다. 즉, 블록 필터 검색부(1220)는 다음 레벨의 블록 필터에서 단계(1810) 내지 단계(1820)을 수행하여 콘텐츠 이름을 연쇄적으로 검색할 수 있다.
- [0191] 단계(1860)에서 해시 테이블 검색부(1230)는 블록 필터 검색부(1220)의 검색 결과에 따른 해시 테이블에서 콘텐츠 이름을 검색할 수 있다. 구체적으로, 단계(1830)에서 블록 필터 검색부(1220)의 블록 필터의 검색 결과가 음성인 경우, 해시 테이블 검색부(1230)는 이전 레벨의 블록 필터에 대응하는 해시 테이블에서 콘텐츠 이름을 검색할 수 있다. 또한, 현재 레벨이 NPT의 마지막 레벨인 경우, 해시 테이블 검색부(1230)는 현재 레벨의 블록 필터에 대응하는 해시 테이블에서 콘텐츠 이름을 검색할 수 있다.
- [0192] 단계(1870)에서 해시 테이블 검색부(1230)는 단계(1860)에서 콘텐츠 이름이 검색되었는지 여부를 확인할 수 있다. 단계(1860)에서 콘텐츠 이름이 검색된 경우, 해시 테이블 검색부(1230)는 단계(1890)을 수행할 수 있다. 또한, 단계(1860)에서 콘텐츠 이름이 검색되지 않은 경우, 해시 테이블 검색부(1230)는 단계(1880)을 수행할 수 있다.
- [0193] 단계(1880)에서 해시 테이블 검색부(1230)는 백 트래킹을 수행할 수 있다. 구체적으로, 단계(1830)에서 콘텐츠 이름이 검색되고, 단계(1870)에서 콘텐츠 이름이 검색되지 않는 경우, 해시 테이블 검색부(1230)는 단계(1830)의 검색 결과를 거트 양성으로 판단할 수 있다. 이때, 해시 테이블 검색부(1230)는 NPT의 레벨을 감소시켜 백 트래킹을 수행할 수 있다.
- [0194] 단계(1890)에서 해시 테이블 검색부(1230)는 단계(1860)에서 검색된 콘텐츠 이름을 식별하고, 식별된 콘텐츠 이름에 해당하는 출력 페이지로 요청 패킷을 전달할 수 있다.
- [0195] 본 발명은 해시 테이블을 이용하여 NPT에 기초한 콘텐츠 이름을 검색할 수 있다. 또한, 본 발명은 블록 필터의 검색 결과에 따라 해시 테이블의 검색 여부를 결정함으로써, 해시 테이블의 접근 회수를 감소시켜 검색 속도를 향상시킬 수 있다.
- [0196] 그리고, 본 발명은 NPT의 레벨 별로 생성된 블록 필터를 연속으로 검색하여 최종 레벨을 검색하고, 최종 레벨의 해시 테이블을 검색함으로써, 해시 테이블의 접근 회수를 최소화시킬 수도 있다.
- [0197] 이상과 같이 본 발명은 비록 한정된 실시예와 도면에 의해 설명되었으나, 본 발명은 상기의 실시예에 한정되는 것은 아니며, 본 발명이 속하는 분야에서 통상의 지식을 가진 자라면 이러한 기재로부터 다양한 수정 및 변형이 가능하다.
- [0198] 그러므로, 본 발명의 범위는 설명된 실시예에 국한되어 정해져서는 아니 되며, 후술하는 특허청구범위뿐 아니라 이 특허청구범위와 균등한 것들에 의해 정해져야 한다.

부호의 설명

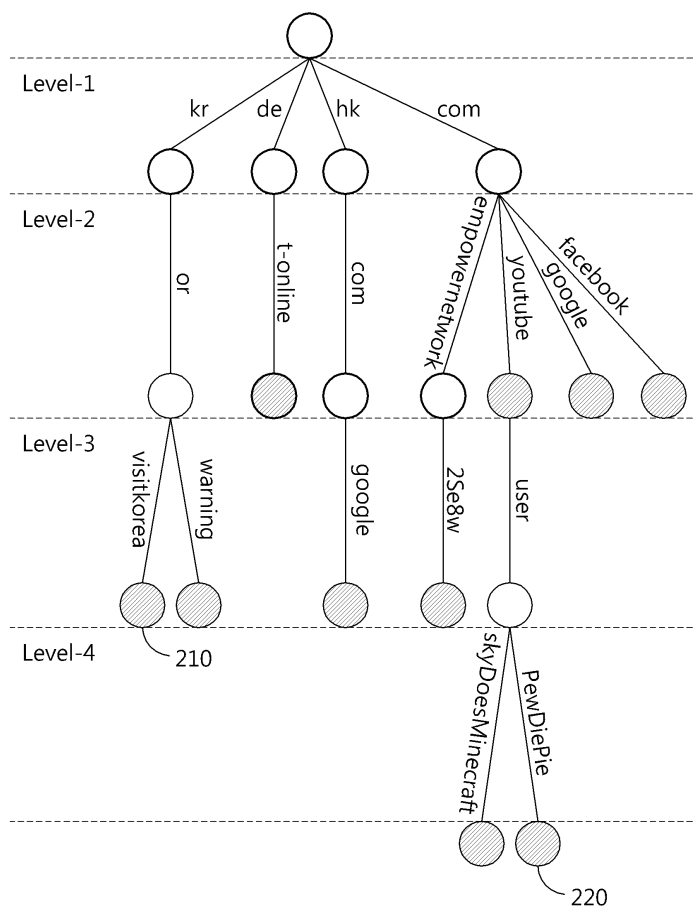
- [0199] 110: 이름 검색 정보 구축 장치
- 120: 이름 검색 장치
- 820: 블록 필터 검색부
- 830: 해시 테이블 검색부

도면

도면1



도면2



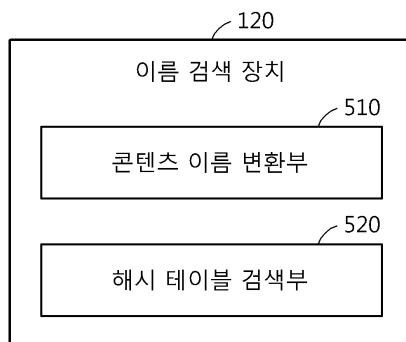
도면3

```
void build_NPT(contentName, outFace) {
    (c1, c2, ..., cN) = Decompose(contentName);
    for (i = 1 to C) { //C is the length of the contentName
        HT_ind = HashFunc(c1 || c2 || ... || ci);
        if (i == C) StoreHT (Hash_ind, (c1, c2, ..., ci), outFace);
        else StoreHT (Hash_ind, (c1, c2, ..., ci));
    }
}
```

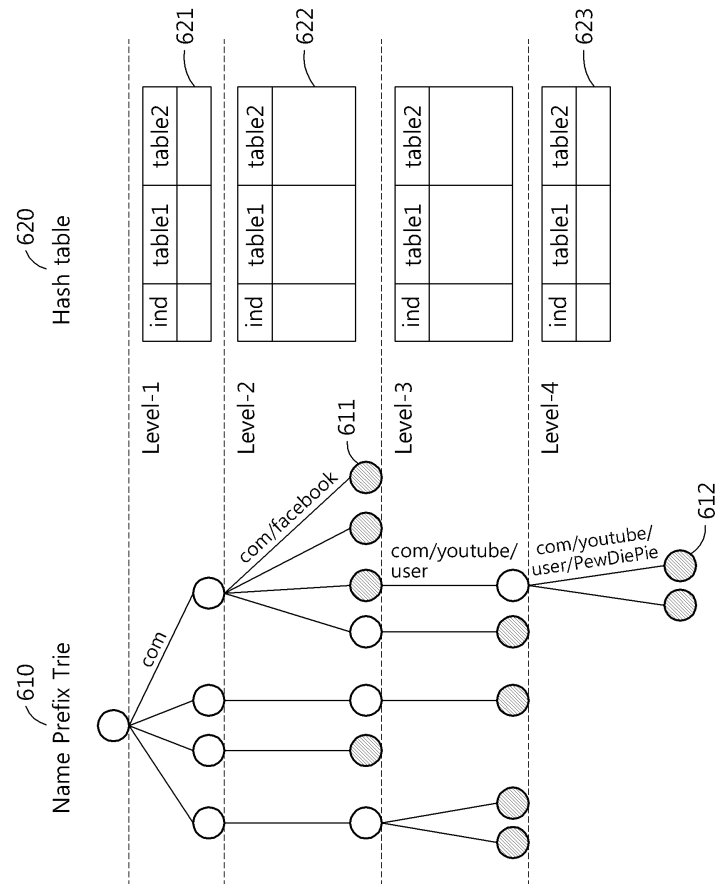
도면4

```
void BFAdd(x) {
    for (i = 1 to k)
        Vector[hi(x)]=1;
}
```

도면5



도면6



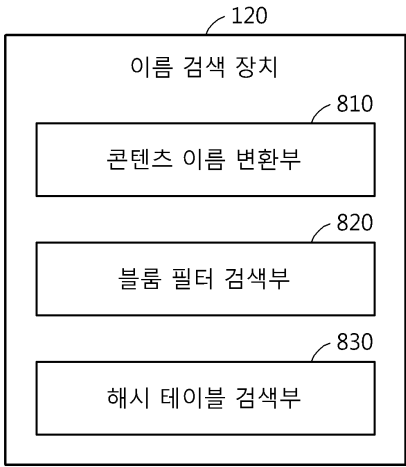
도면7

```

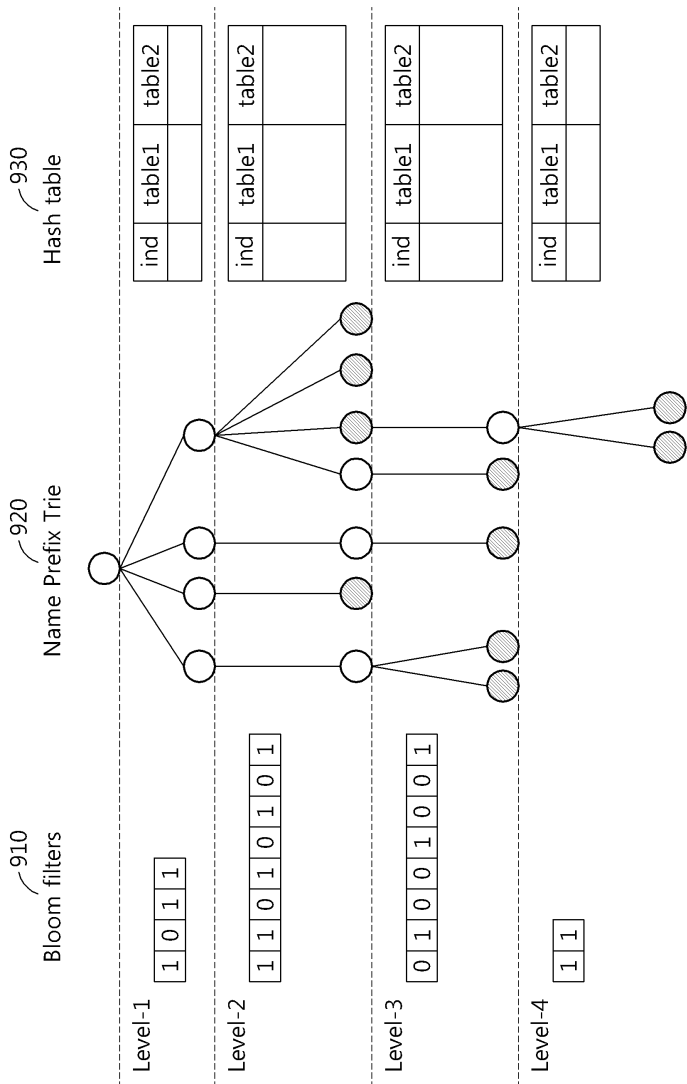
int search_NPT(inName) {
    (c1, c2, ..., cL) = Decompose(inName);
    for (i = 1 to L) { //L is the length of the inName
        Hash_ind = HashFunc(c1 || c2 || ... || ci);
        out_result = SearchHT(Hash_ind, i);
        if (out_result != NULL) BMN = out_result;
        break; //not found
    }
    return BMN;
}

```

도면8



도면9



도면10

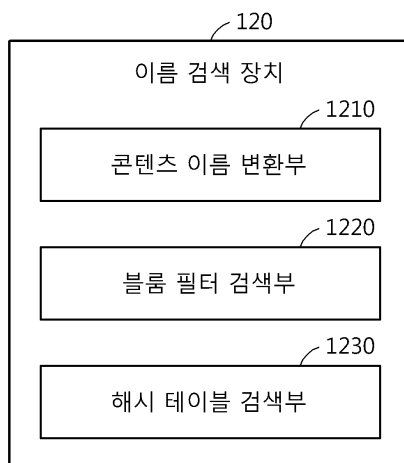
```
bool BFQuery(x) {
    for (i = 1 to k) {
        if (Vector[hi(x)] = 0) return negative;
        else return positive;
    }
}
```

도면11

```
int search_NPT_BF(inName) {
    (c1, c2, ..., cN) = Decompose(inName);
    for (i = 1 to L) { //L is the length of the inName

        BF_ind = MakeBFInd(c1 || c2 || ... || ci);
        if (BFQuery(BF_ind) == positive) {
            Hash_ind = HashFunc(c1 || c2 || ... || ci);
            out_result = SearchHT(Hash_ind, i);
            if (out_result != NULL) BMN = out_result;
            break; //false positive
        }
        else break; //Search is immediately finished when BF negative occurs
    }
    return BMN;
}
```

도면12



도면13

```

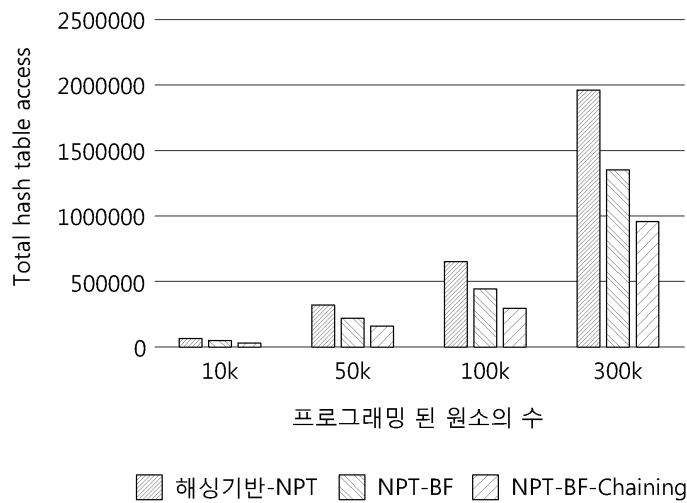
int search_NPT_BF_Chaining(inName) {
    (c1, c2, ..., cL) = Decompose(inName);
    for (i = 1 to L) {
        BF_ind = MakeBFInd (c1 || c2 || ... || ci);
        if (BFQuery(BF_ind) == negative) {
            break;
        }
    }

    i = i - 1;

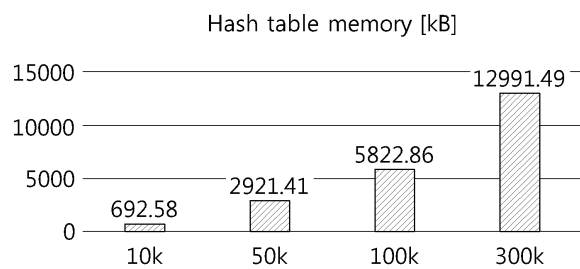
    do {
        Hash_ind = HashFunc(c1 || c2 || ... || ci);
        out_result = SearchHT(Hash_ind, i);
        i = i - 1; //back tracking
    } while (out_result == NULL);
    return BMN = out_result;
}

```

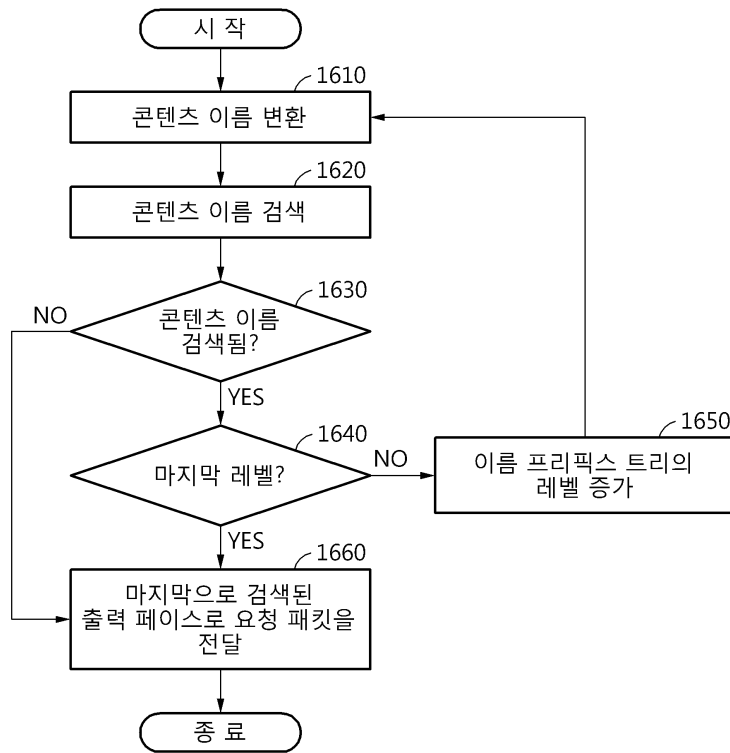
도면14



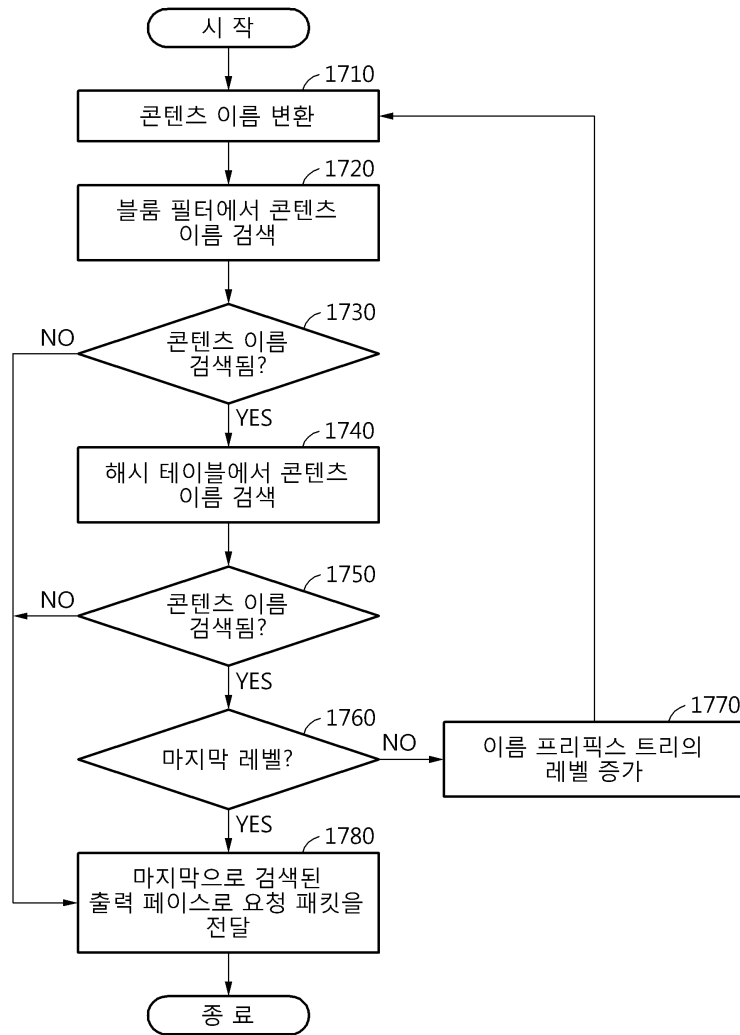
도면15



도면16



도면17



도면18

