



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2016-0100386
(43) 공개일자 2016년08월23일

(51) 국제특허분류(Int. Cl.) H04L 29/08 (2006.01) G06F 17/30 (2006.01) H04L 12/26 (2006.01) H04L 12/721 (2013.01) (52) CPC특허분류 H04L 67/2842 (2013.01) G06F 17/30902 (2013.01) (21) 출원번호 10-2016-7019627 (22) 출원일자(국제) 2015년01월12일 심사청구일자 2016년07월19일 (85) 번역문제출일자 2016년07월19일 (86) 국제출원번호 PCT/US2015/011000 (87) 국제공개번호 WO 2015/112362 국제공개일자 2015년07월30일 (30) 우선권주장 61/930,072 2014년01월22일 미국(US) 14/514,905 2014년10월15일 미국(US)	(71) 출원인 알까멜 루슨트 프랑스 92100 불롱뉴-비영꾸르 루뜨 들 라 렌느 148/152 (72) 발명자 툴리노, 안토니아 미국 뉴저지주 07974-0636 머레이 힐 마운틴 애비 뉴 600-700 요르카, 제이미 미국 뉴저지주 07974-0636 머레이 힐 마운틴 애비 뉴 600-700 산무감, 카디케안 미국 뉴저지주 07974-0636 머레이 힐 마운틴 애비 뉴 600-700 (74) 대리인 양영준, 전경석, 백만기
--	---

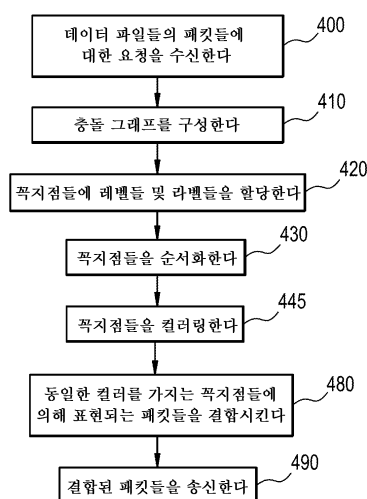
전체 청구항 수 : 총 10 항

(54) 발명의 명칭 **네트워크-코딩되고 캐싱-지원되는 콘텐츠 배포를 위한 디바이스들 및 방법들**

(57) 요약

네트워크에서 데이터 파일들을 전송하기 위한 방법은 데이터 파일들의 패킷들에 대한 요청들을 목적지 디바이스들로부터 수신하는 단계를 포함한다. 방법은 각각의 목적지 디바이스에 의해 요청되는 각각의 패킷이 충돌 그래프의 복수의 꼭지점들 내의 별개의 꼭지점에 의해 표현되도록 충돌 그래프를 구성하는 단계를 포함하고, 복수의 꼭지점들은 목적지 디바이스들과 연관된다. 방법은 라벨들을 복수의 꼭지점들에 할당하는 단계를 포함한다. 방법은 복수의 꼭지점들에 레벨들을 할당하는 단계를 포함한다. 방법은 가장 높은 레벨을 가지는 꼭지점들로부터 가장 낮은 레벨을 가지는 꼭지점들로 복수의 꼭지점들을 순서화하는 단계를 포함한다. 방법은 순서화에 기초하여 복수의 꼭지점들을 컬러링하는 단계를 포함한다. 방법은 동일한 컬러를 가지는 복수의 꼭지점들 내의 꼭지점들에 의해 표현되는 패킷들을 결합시키는 단계를 포함한다. 방법은 결합된 패킷들을 송신하는 단계를 포함한다.

대표도 - 도4



(52) CPC특허분류

H04L 43/045 (2013.01)

H04L 43/062 (2013.01)

H04L 45/14 (2013.01)

H04L 67/06 (2013.01)

H04L 67/1097 (2013.01)

H04L 67/2833 (2013.01)

H04L 67/2847 (2013.01)

H04L 67/327 (2013.01)

H04L 67/36 (2013.01)

명세서

청구범위

청구항 1

네트워크 엘리먼트(network element)로서,

데이터 파일들의 패킷들에 대한 요청들을 목적지 디바이스들로부터 수신하고;

각각의 목적지 디바이스에 의해 요청되는 각각의 패킷이 충돌 그래프(conflict graph)의 복수의 꼭지점들(vertices) 내의 별개의 꼭지점에 의해 표현되도록 상기 충돌 그래프를 구성하고 - 상기 복수의 꼭지점들은 상기 목적지 디바이스들과 연관됨 - ;

상기 복수의 꼭지점들에 라벨들(labels)을 할당하고 - 각각의 라벨은 패킷을 요청하는 목적지 디바이스 및 상기 패킷을 저장하는 목적지 디바이스 캐시들을 표기하는 인덱스들의 세트임 - ;

상기 복수의 꼭지점들에 레벨들(levels)을 할당하고 - 각각의 레벨은 상기 패킷을 요청하는 목적지 디바이스들의 수 및 상기 패킷을 저장하는 목적지 디바이스 캐시들의 수를 표시함 - ;

가장 높은 레벨을 가지는 꼭지점들로부터 가장 낮은 레벨을 가지는 꼭지점들로 상기 복수의 꼭지점들을 순서화(order)하고;

상기 순서화에 기초하여 상기 복수의 꼭지점들을 컬러링하고;

동일한 컬러를 가지는 상기 복수의 꼭지점들 내의 꼭지점들에 의해 표현되는 패킷들을 결합시키고;

상기 결합된 패킷들을 송신하도록

구성되는 프로세서

를 포함하는 네트워크 엘리먼트.

청구항 2

제1항에 있어서, 각각의 레벨은 패킷을 요청하는 목적지 디바이스들의 수와 상기 요청된 패킷을 저장하는 목적지 디바이스 캐시들의 수의 합산인 네트워크 엘리먼트.

청구항 3

제1항에 있어서, 상기 프로세서는 동일한 컬러를 가지는 꼭지점들에 의해 표현되는 패킷들에 대해 유한 필드(finite field) 상에서 선형 조합 연산들을 수행함으로써 패킷들을 결합시키도록 구성되는 네트워크 엘리먼트.

청구항 4

제1항에 있어서, 상기 프로세서는 (i) 상기 복수의 꼭지점들 중 제1 꼭지점과 제2 꼭지점이 동일한 패킷을 표현하지 않고, (ii) 상기 제1 꼭지점에 의해 표현되는 패킷이 상기 제2 꼭지점과 연관된 목적지 디바이스의 캐시에 저장되지 않거나, 또는 상기 제2 꼭지점에 의해 표현되는 패킷이 상기 제1 꼭지점과 연관된 목적지 디바이스의 캐시에 저장되지 않는 경우, 상기 제1 꼭지점과 상기 제2 꼭지점 사이에 링크를 생성함으로써 상기 충돌 그래프를 구축하도록 구성되는 네트워크 엘리먼트.

청구항 5

제4항에 있어서, 상기 프로세서는,

가장 높은 레벨과 동일한 라벨 길이를 가지는 컬러링되지 않은 꼭지점(uncolored vertex)을 선택하고, 그리고

(i) 상기 복수의 꼭지점들 내의 컬러링되지 않은 꼭지점들의 서브세트 내의 꼭지점들이 원하는 라벨 길이를 가지고 - 상기 원하는 라벨 길이는 입력 파라미터들에 그리고 선택된 꼭지점의 레벨에 기초함 - (ii) 상기 서브세트 내의 꼭지점들이 선택된 꼭지점에 대한 링크를 가지지 않고, (iii) 상기 서브세트 내의 꼭지점들이 서로간

에 링크를 가지지 않고, 그리고 (iv) 상기 서브세트의 카디널리티(cardinality)가 상기 레벨 마이너스 1의 값과 같은 경우, 상기 선택된 꼭지점에, 그리고 상기 서브세트에 동일한 컬러를 할당함으로써,

상기 복수의 꼭지점들을 컬러링하도록 구성되는 네트워크 엘리먼트.

청구항 6

제5항에 있어서, 상기 프로세서는,

가장 높은 레벨의 모든 꼭지점들이 선택될 때까지 동일한 컬러 동작들을 선택하고 할당하는 것을 반복적으로 수행하고, 그리고

여전히 컬러링되지 않은 가장 높은 레벨의 꼭지점들의 레벨들을 가장 높은 레벨 바로 아래의 레벨로 업데이트하도록

구성되는 네트워크 엘리먼트.

청구항 7

제6항에 있어서, 상기 프로세서는, 상기 복수의 꼭지점들이 컬러링됨을 표시하기 위해 가장 낮은 레벨에 도달될 때까지 상기 레벨들을 업데이트하도록 구성되는 네트워크 엘리먼트.

청구항 8

제7항에 있어서, 상기 프로세서는 상기 복수의 컬러링된 꼭지점들에 대해 로컬 탐색(local search)을 수행하여 상기 충돌 그래프에 사용되는 컬러들의 개수를 감소시키도록 구성되는 네트워크 엘리먼트.

청구항 9

제8항에 있어서, 상기 프로세서는,

상기 충돌 그래프의 기존의 컬러들로부터 컬러를 선택하고,

상기 선택된 컬러를 가지는 꼭지점들을 식별하고, 그리고

상기 식별되는 꼭지점들에 링크되는 꼭지점들이 상기 기존의 컬러들로부터 선택되는 상이한 컬러로 컬러링되지 않은 경우, 상기 선택된 컬러를 상기 상이한 컬러로 대체함으로써,

상기 로컬 탐색을 수행하도록 구성되는 네트워크 엘리먼트.

청구항 10

네트워크에서 데이터 파일들을 전송하기 위한 방법으로서,

상기 데이터 파일들의 패킷들에 대한 요청들을 목적지 디바이스들로부터 수신하는 단계;

각각의 목적지 디바이스에 의해 요청되는 각각의 패킷이 충돌 그래프의 복수의 꼭지점들 내의 별개의 꼭지점에 의해 표현되도록 상기 충돌 그래프를 구성하는 단계 - 상기 복수의 꼭지점들은 상기 목적지 디바이스들과 연관됨 - ;

상기 복수의 꼭지점들에 라벨들을 할당하는 단계 - 각각의 라벨은 패킷을 요청하는 목적지 디바이스들 및 상기 패킷을 저장하는 목적지 디바이스 캐시들을 표기하는 인덱스들의 세트임 - ;

상기 복수의 꼭지점들에 레벨들을 할당하는 단계 - 각각의 레벨은 상기 패킷을 요청하는 목적지 디바이스들의 수 및 상기 패킷을 저장하는 목적지 디바이스 캐시들의 수를 표시함 - ;

가장 높은 레벨을 가지는 꼭지점들로부터 가장 낮은 레벨을 가지는 꼭지점들로 상기 복수의 꼭지점들을 순서화하는 단계;

상기 순서화에 기초하여 상기 복수의 꼭지점들을 컬러링하는 단계;

동일한 컬러를 가지는 상기 복수의 꼭지점들 내의 꼭지점들에 의해 표현되는 패킷들을 결합시키는 단계; 및

상기 결합된 패킷들을 송신하는 단계

를 포함하는 방법.

발명의 설명

기술 분야

[0001] 관련 출원들에 대한 교차-참조

[0002] 이 출원은 2014년 1월 22일에 출원되었으며, 그 전체 내용이 참조로 본원에 포함되는 미국 가출원 제61/930,072호를, 35 U.S.C. § 119(e) 하에서, 우선권 주장한다.

배경 기술

[0003] 현재, 콘텐츠 배포 네트워크(content distribution network)(CDN)들은 주문형 오디오/비디오 스트리밍의 인기의 증가와 연관된 용량 및 효율성 이슈들에 당면한다. 이러한 이슈들을 다루는 한 가지 방식은 네트워크 캐싱 및 네트워크 코딩을 통하는 것이다. 예를 들어, 종래의 콘텐츠 배포 네트워크(CDN) 솔루션은 네트워크 내의 캐싱 위치들 중 콘텐츠 복제본들의 배치를 위해 중앙화된 알고리즘을 사용한다. 종래의 솔루션들은 또한 캐시 적중률들(cache hit ratios)을 개선하기 위해 배포된 캐시들을 국부적으로 관리하도록 LRU(least recently used) 또는 LFU(least frequently used)와 같은 캐시 대체 정책들을 포함한다. 다른 종래의 솔루션들은 랜덤 선형 네트워크 코딩을 사용하여 그룹들 내의 패킷들을 전달하는데, 이는 용량-제한된 네트워크들에서의 스루풋을 개선할 수 있다.

[0004] 그러나, 종래의 네트워크 캐싱 및 네트워크 코딩 솔루션들은 캐싱 및 전송 자원들의 상대적인 효율성을 고려하지 않는다. 이는 전달되는 객체 또는 파일 당 차선의 비용을 초래한다. 더욱이, 종래의 콘텐츠 전달 솔루션들은 네트워크 캐싱 및 네트워크 코딩의 가능한 조합된 이점들을 사용하지 않는다.

발명의 내용

[0005] 적어도 일 예시적인 실시예는 캐싱 페이지 및 전달 페이지를 포함하는 콘텐츠 배포를 위한 방법들 및/또는 디바이스들에 관한 것이다.

[0006] 적어도 일 예시적인 실시예에 따르면, 네트워크에서 데이터 파일들을 전송하기 위한 방법은 데이터 파일들의 패킷들에 대한 요청들을 목적지 디바이스들로부터 수신하는 것을 포함한다. 방법은 각각의 목적지 디바이스에 의해 요청되는 각각의 패킷이 충돌 그래프의 복수의 꼭지점들 내의 별개의 꼭지점에 의해 표현되도록 충돌 그래프를 구성하는 것을 포함하고, 복수의 꼭지점들은 목적지 디바이스들과 연관된다. 방법은 복수의 꼭지점들에 라벨들을 할당하는 것을 포함하고, 각각의 라벨은 패킷을 요청하는 목적지 디바이스들 및 패킷을 저장하는 목적지 디바이스 캐시들을 표기하는 인덱스들의 세트이다. 방법은 복수의 꼭지점들에 레벨들을 할당하는 것을 포함하고, 각각의 레벨은 패킷을 요청하는 목적지 디바이스들의 수 및 패킷을 저장하는 목적지 디바이스 캐시들의 수를 나타낸다. 방법은 가장 높은 레벨을 가지는 꼭지점들로부터 가장 낮은 레벨을 가지는 꼭지점들로 복수의 꼭지점들을 순서화하는 것을 포함한다. 방법은 순서화에 기초하여 복수의 꼭지점들을 컬러링하는 것을 포함한다. 방법은 동일한 컬러를 가지는 복수의 꼭지점들 내의 꼭지점들에 의해 표현되는 패킷들을 결합시키는 것을 포함한다. 방법은 결합된 패킷들을 송신하는 것을 포함한다.

[0007] 적어도 일 예시적인 실시예에 따르면, 각각의 레벨은 패킷을 요청하는 목적지 디바이스들의 수와 요청된 패킷을 저장하는 목적지 디바이스 캐시들의 수의 합산이다.

[0008] 적어도 일 예시적인 실시예에 따르면, 결합은 동일한 컬러를 가지는 꼭지점들에 의해 표현되는 패킷들에 대해 유한 필드 상에서 선형 조합 연산들을 수행한다.

[0009] 적어도 일 예시적인 실시예에 따르면, 구성은 (i) 제1 꼭지점과 제2 꼭지점이 동일한 패킷을 나타내지 않고, (ii) 제1 꼭지점에 의해 표현된 패킷이 제2 꼭지점과 연관되는 목적지 디바이스의 캐시에 저장되지 않거나, 또는 제2 꼭지점에 의해 표현되는 패킷이 제1 꼭지점과 연관되는 목적지 디바이스의 캐시에 저장되지 않는 경우, 복수의 꼭지점들 중 제1 꼭지점과 제2 꼭지점 사이에 링크를 생성하는 것을 포함한다.

[0010] 적어도 일 예시적인 실시예에 따르면, 컬러링은 가장 높은 레벨을 가지는 컬러링되지 않은 꼭지점을 선택하는 것, 및 (i) 서브세트 내의 꼭지점들이 원하는 레벨을 가지고 - 입력 파라미터들 및 선택된 꼭지점의 라벨에 기초함 -, (ii) 서브세트 내의 꼭지점들이 선택된 꼭지점에 대한 링크를 가지지 않고, (iii) 서브세트 내의 꼭

지점들이 서로 사이에 링크를 가지지 않고, (iv) 서브세트의 카디널리티가 레벨의 값 마이너스 일과 같은 경우, 선택된 꼭지점에 그리고 복수의 꼭지점들 내의 컬러링되지 않은 꼭지점들의 서브세트에 동일한 컬러를 할당하는 것을 포함한다.

- [0011] 적어도 일 예시적인 실시예에 따르면, 컬러링은 가장 높은 레벨의 모든 꼭지점들이 선택될 때까지, 동일한 컬러 동작들을 선택하고 할당하는 것을 반복적으로 수행하는 것을 포함한다.
- [0012] 적어도 일 예시적인 실시예에 따르면, 컬러링은, 컬러링되지 않은 꼭지점들의 수가 가장 높은 레벨보다 더 크거나 같은 경우, 컬러링되지 않은 가장 높은 레벨에서 꼭지점들에 대해 추가적인 컬러링 동작을 수행하는 것을 포함한다. 컬러링은 추가적인 컬러링 동작 이후 여전히 컬러링되지 않은 가장 높은 레벨의 꼭지점들의 레벨들을 가장 높은 레벨 아래의 다음 레벨로 업데이트하는 것을 포함한다.
- [0013] 적어도 일 예시적인 실시예에 따르면, 컬러링은 충돌 그래프의 복수의 꼭지점들이 컬러링될 때까지 수행된다.
- [0014] 적어도 일 예시적인 실시예에 따르면, 방법은 복수의 컬러링된 꼭지점들에 대해 로컬 탐색을 수행하여 충돌 그래프에 대해 사용되는 컬러들의 수를 감소시키는 것을 포함한다.
- [0015] 적어도 일 예시적인 실시예에 따르면, 로컬 탐색을 수행하는 것은 충돌 그래프의 기존의 컬러들로부터 컬러를 선택하는 것, 선택된 컬러를 가지는 꼭지점들을 식별하는 것, 및 식별된 꼭지점들에 링크되는 꼭지점들이 상이한 컬러로 컬러링되지 않은 경우, 선택된 컬러를 기존의 컬러들로부터 선택된 상이한 컬러로 대체하는 것을 포함한다.
- [0016] 위의 방법들이 통신 네트워크 내의 네트워크 엘리먼트(예를 들어, 콘텐츠 소스)에 의해 수행될 수 있다는 것이 이해되어야 한다.

도면의 간단한 설명

- [0017] 예시적인 실시예들은 본원의 하기에 주어진 상세한 설명 및 첨부 도면들로부터 더욱 완전히 이해될 것이며, 여기서 동일한 엘리먼트들은 동일한 참조 부호들에 의해 표현되며, 이는 단지 예시로서 주어지며, 따라서 예시적인 실시예들의 제한이 아니다.
 - 도 1은 적어도 일 예시적인 실시예에 따른 콘텐츠 배포 네트워크를 도시한다.
 - 도 2는 예시적인 실시예에 따른 네트워크 엘리먼트의 예시적인 구조를 예시하는 다이어그램이다.
 - 도 3a-3b는 도 2의 네트워크 엘리먼트의 예시적인 동작들을 예시하는 플로우차트들이다.
 - 도 4는 적어도 일 예시적인 실시예에 따른 전달 페이지에 대한 예시적인 동작들을 예시한다.
 - 도 5는 적어도 일 예시적인 실시예에 따른 무방향 충돌 그래프를 구성하기 위한 예시적인 동작들을 예시한다.
 - 도 6은 적어도 일 예시적인 실시예에 따라 충돌 그래프를 컬러링하기 위한 예시적인 동작들을 예시한다.
 - 도 7은 적어도 일 예시적인 실시예에 따른 컬러링된 충돌 그래프를 예시한다.

발명을 실시하기 위한 구체적인 내용

- [0018] 다양한 예시적인 실시예들은 이제, 일부 예시적인 실시예들이 도시되는 첨부 도면들에 관련하여 더 완전히 기술될 것이다.
- [0019] 상세한 예시적인 실시예들이 본원에 개시된다. 그러나, 본원에 개시되는 특정한 구조적 및 기능적 상세항목들은 예시적인 실시예들을 기술할 목적으로 단지 대표적이다. 그러나, 이 발명은 많은 대안적인 형태들로 구현될 수 있으며, 본원에 설명되는 실시예들로부터 제한되는 것으로서 해석되지 않아야 한다.
- [0020] 따라서, 예시적인 실시예들은 다양한 수정들 및 대안 형태들이 가능하지만, 실시예들은 도면들에서 예시로서 도시되며, 본원에서 상세하게 기술될 것이다. 그러나, 예시적인 실시예들을 개시되는 특정 형태들로 제한하려는 의도가 없다는 것이 이해되어야 한다. 반면, 예시적인 실시예들은 이 개시내용의 범위 내에 드는 모든 수정들, 등가물들 및 대안들을 커버한다. 동일한 번호들은 도면들의 기재 전반에 걸쳐 동일한 엘리먼트들을 지칭한다.
- [0021] 용어들 제1, 제2 등이 다양한 엘리먼트들을 기술하도록 본원에서 사용될 수 있지만, 이러한 엘리먼트들은 이러한 용어들에 의해 제한되지 않아야 한다. 이러한 용어들은 하나의 엘리먼트를 또다른 엘리먼트와 구별하기 위해서만 사용된다. 이 개시내용의 범위로부터 벗어나지 않고, 예를 들어, 제1 엘리먼트는 제2 엘리먼트라 지칭

될 수 있고, 유사하게, 제2 엘리먼트는 제1 엘리먼트라 지칭될 수 있다. 본원에서 사용되는 바와 같이, 용어 "및/또는"은 연관된 열거 항목들 중 하나 이상의 임의의 그리고 모든 조합들을 포함한다.

- [0022] 엘리먼트가 또다른 엘리먼트에 "접속되는" 또는 "커플링되는" 것으로서 지칭될 때, 그것은 다른 엘리먼트에 직접 접속 또는 커플링될 수 있거나, 또는 중재 엘리먼트들이 존재할 수 있다. 반면, 엘리먼트가 "직접 접속되는" 또는 "직접 커플링되는" 것으로서 지칭될 때, 존재하는 중재 엘리먼트들이 없다. 엘리먼트들 간의 관계를 기술하기 위해 사용되는 다른 단어들이 유사한 방식으로 해석될 수 있다(예를 들어, "~사이에" 대 "직접 ~ 사이에", "인접한" 대 "바로 인접한" 등).
- [0023] 본원에서 사용되는 용어는 단지 특정 실시예들을 기술하는 목적이며, 제한적인 것으로 의도되지 않는다. 본원에서 사용되는 바와 같이, 단수 형태("a", "an" 및 "the")는, 문맥이 다른 방식으로 명백하게 지시하지 않는 한, 복수 형태도 포함하도록 의도된다. 용어들 "포함하다(comprises, includes) 및/또는 "포함하는(comprising, including)"이, 본원에서 사용될 때, 언급된 특징들, 정수들, 단계들, 동작들, 엘리먼트들, 및/또는 컴포넌트들의 존재를 특정하지만, 하나 이상의 다른 특징들, 정수들, 단계들, 동작들, 엘리먼트들, 컴포넌트들, 및/또는 이들의 그룹들의 존재 또는 추가를 불가능하게 하지 않는다는 것이 추가로 이해될 것이다.
- [0024] 일부 대안적인 구현예들에서, 주지된 기능들/동작들이 도면들에 주지된 순서를 벗어나 발생할 수 있다는 점에 또한 유의해야 한다. 예를 들어, 연속적으로 도시된 2개의 도면들은 관련된 기능성/동작들에 따라, 실제로 거의 동시에 실행될 수 있거나, 또는 때때로 역순서로 실행될 수 있다.
- [0025] 특정 상세항목들이 예시적인 실시예들의 철저한 이해를 제공하기 위해 후속하는 기재에 제공된다. 그러나, 예시적인 실시예들이 이러한 특정 상세항목들 없이도 구현될 수 있다는 것이 본 기술분야의 통상의 기술자에 의해 이해될 것이다. 예를 들어, 시스템들은, 불필요하게 상세하여 예시적인 실시예들을 모호하게 하지 않기 위해, 블록도들로 도시될 수 있다. 다른 경우들에서, 널리 알려진 프로세스들, 구조들 및 기법들은 예시적인 실시예들을 모호하게 하는 것을 회피하기 위해 불필요하게 상세하지 않게 도시될 수 있다.
- [0026] 후속하는 기재에서, 예시적인 실시예들은, 특정 작업들을 수행하거나 특정 추상적 데이터 타입들을 실행하는, 루틴들, 프로그램들, 객체들, 컴포넌트들, 데이터 구조들 등을 포함하는 프로그램 모듈들 또는 기능 프로세스들로서 구현될 수 있으며, 기존의 네트워크 엘리먼트들(예를 들어, 기지국들, 기지국 제어기들, NodeB들, eNodeB들 등)에 있는 기존의 하드웨어를 사용하여 구현될 수 있는, 작용들 및 동작들의 심볼 표현들에 관련하여(예를 들어, 플로우 차트, 흐름도, 데이터 흐름도, 구조도, 블록도 등의 형태로) 기술될 것이다. 이러한 기존의 하드웨어는 하나 이상의 중앙 처리기(CPU)들, 디지털 신호 프로세서(DSP)들, 주문형 집적 회로들, 필드 프로그래밍 가능 게이트 어레이(FPGA)들, 컴퓨터들 등을 포함할 수 있다.
- [0027] 플로우 차트가 동작들을 순차적 프로세스로서 기술할 수 있지만, 동작들 중 다수가 병렬로, 동시에 또는 일제히 수행될 수 있다. 추가로, 동작들의 순서는 재배열될 수 있다. 프로세스는 그것의 동작들이 완료될 때 종료될 수 있지만, 도면에 포함되지 않은 추가적인 단계들을 또한 포함할 수 있다. 프로세스는 방법, 함수, 절차, 서브루틴, 서브프로그램 등에 대응할 수 있다. 프로세스가 함수에 대응할 때, 그것의 종료는 그 함수를 호출 함수 또는 메인 함수로 리턴시키는 것에 대응할 수 있다.
- [0028] 본원에 개시되는 바와 같이, 용어 "저장 매체" 또는 "컴퓨터 판독가능 저장 매체"는, 판독 전용 메모리(read only memory)(ROM), 랜덤 액세스 메모리(random access memory)(RAM), 자기 RAM, 코어 메모리, 자기 디스크 저장 매체들, 광학 저장 매체들, 플래시 메모리 디바이스들, 및/또는 정보를 저장하기 위한 다른 유형 머신 판독 가능 매체들을 포함한, 데이터를 저장하기 위한 하나 이상의 디바이스들을 나타낼 수 있다. 용어 "컴퓨터-판독 가능 매체"는, 휴대용 또는 고정식 저장 디바이스들, 광학 저장 디바이스들, 및 명령(들) 및/또는 데이터를 저장하고, 보유(contains)하거나 반송할 수 있는 다양한 다른 매체들을 포함할 수 있지만, 이에 제한되지 않는다.
- [0029] 또한, 예시적인 실시예들은, 하드웨어, 소프트웨어, 펌웨어, 미들웨어, 마이크로코드, 하드웨어 기술 언어들, 또는 이들의 임의의 조합에 의해 구현될 수 있다. 소프트웨어, 펌웨어, 미들웨어 또는 마이크로코드로 구현될 때, 필수 작업들을 수행하기 위한 프로그램 코드 또는 코드 세그먼트들은 컴퓨터 판독가능 저장 매체와 같은 머신 또는 컴퓨터 판독가능 매체에 저장될 수 있다. 소프트웨어로 구현될 때, 특수 목적 프로세서 또는 특수 목적 프로세서들이 필수 작업들을 수행할 것이다.
- [0030] 코드 세그먼트는 절차, 함수, 서브프로그램, 프로그램, 루틴, 서브루틴, 모듈, 소프트웨어 패키지, 클래스, 또는 명령들, 데이터 구조들 또는 프로그램 선언문들의 임의의 조합을 나타낼 수 있다. 코드 세그먼트는 정보, 데이터, 인수들, 파라미터들 또는 메모리 콘텐츠를 전달 및/또는 수신함으로써 또다른 코드 세그먼트 또는 하드

웨어 회로에 커플링될 수 있다. 정보, 인수들, 파라미터들, 데이터 등은 메모리 공유, 메시지 전달, 토큰 전달, 네트워크 전송들을 포함하는 임의의 적절한 수단을 통해 전달되고, 포워딩되거나, 전송될 수 있다.

- [0031] 도 1은 적어도 일 예시적인 실시예에 따른 콘텐츠 배포 네트워크를 도시한다.
- [0032] 도 1에 도시된 바와 같이, 콘텐츠 배포 네트워크(CDN)는 복수의 목적지 디바이스들(또는 사용자 디바이스들)(200)에 접속되는 네트워크 엘리먼트(151)를 포함할 수 있다. 네트워크 엘리먼트(151)는 데이터 파일들(예를 들어, 영화 파일들)을 배포하기 위한 콘텐츠 소스(예를 들어, 멀티캐스트 소스)일 수 있다. 목적지 디바이스들(200)은 콘텐츠 소스로부터 데이터를 요청하는 최종 사용자 디바이스들일 수 있다. 예를 들어, 각각의 목적지 디바이스(200)는 사용자가 요청된 데이터에 액세스하도록 하는 디바이스의 일부이거나 또는 디바이스와 연관될 수 있다. 예를 들어, 각각의 목적지 디바이스(200)는 셋톱 박스, 개인용 컴퓨터, 태블릿, 모바일 폰, 또는 오디오 및 비디오를 스트리밍하기 위해 연관되어 사용되는 임의의 다른 디바이스일 수 있다. 목적지 디바이스들(200) 각각은 네트워크 엘리먼트(151)로부터 수신되는 데이터를 저장하기 위한 메모리를 포함할 수 있다. 네트워크 엘리먼트(151) 및 목적지 디바이스(200)의 구조 및 동작은 도 2 및 도 3에 관련하여 하기에 더 상세히 기술될 것이다.
- [0033] 도 2는 예시적인 실시예에 따른 네트워크 엘리먼트의 예시적인 구조를 예시하는 다이어그램이다. 적어도 일 예시적인 실시예에 따르면, 네트워크 엘리먼트(151)는 통신 네트워크(예를 들어, 도 1의 콘텐츠 배포 네트워크(CDN))에서 사용하기 위해 구성될 수 있다. 도 2를 참조하면, 네트워크 엘리먼트(151)는, 예를 들어, 데이터 버스(159), 송신기(152), 수신기(154), 메모리(156), 및 프로세서(158)를 포함할 수 있다. 간략함을 위해 별도의 기체가 본원에 포함되지는 않지만, 각각의 목적지 디바이스(200)가 네트워크 엘리먼트(151)와 동일하거나 유사한 구조를 가질 수 있다는 것이 이해되어야 한다.
- [0034] 송신기(152), 수신기(154), 메모리(156) 및 프로세서(158)는 데이터 버스(159)를 사용하여 서로에 데이터를 송신하고 그리고/또는 서로로부터 데이터를 수신할 수 있다. 송신기(152)는 예를 들어, 데이터 신호들, 제어 신호들, 및 신호 강도/품질 정보를 포함하는 무선 신호들을 하나 이상의 무선 접속들을 통해 통신 네트워크 내의 다른 네트워크 엘리먼트들에 전송하기 위한 하드웨어 및 임의의 필수 소프트웨어를 포함하는 디바이스이다.
- [0035] 수신기(154)는, 예를 들어, 데이터 신호들, 제어 신호들, 및 신호 강도/품질 정보를 포함하는 무선 신호들을 하나 이상의 무선 접속들을 통해 통신 네트워크 내의 다른 네트워크 엘리먼트들에 수신하기 위한 하드웨어 및 임의의 필수 소프트웨어를 포함하는 디바이스이다.
- [0036] 메모리(156)는 자기 저장소, 플래시 저장소 등을 포함한, 데이터를 저장할 수 있는 임의의 디바이스일 수 있다.
- [0037] 프로세서(158)는, 예를 들어, 입력 데이터에 기초하여 특정 동작들을 수행하도록 구성되는, 또는 컴퓨터 관독가능한 코드에 포함되는 명령들을 실행할 수 있는 특수 목적 프로세서를 포함한, 데이터를 프로세싱할 수 있는 임의의 디바이스일 수 있다. 예를 들어, 하기에 기술되는 수정들 및 방법들이 메모리(156)에 저장되고 네트워크 엘리먼트(151) 내의 프로세서(158)에 의해 구현될 수 있다는 것이 이해되어야 한다.
- [0038] 또한, 하기의 수정들 및 방법들이 네트워크 엘리먼트(151)의 전송된 엘리먼트들 중 하나 이상에 의해 수행될 수 있다는 것이 이해되어야 한다. 예를 들어, 수신기(154)는 "수신하는", "취득하는" 등의 단계들을 수행할 수 있고; 송신기(152)는 "전송하는", "출력하는", "송신하는" 등의 단계들을 수행할 수 있고; 프로세서(158)는 "결정하는", "생성하는", "상관시키는", "계산하는" 등의 단계들을 수행할 수 있고; 그리고 메모리(156)는 "저장하는 (storing, saving)" 등의 단계들을 수행할 수 있다.
- [0039] 예시적인 실시예들이 캐싱 페이지(도 3a 및 3b를 참조하여 하기에 기술됨) 및 전달 페이지(도 4-7을 참조하여 하기에 기술됨)에 관한 것임이 이해되어야 한다.
- [0040] 도 3-6은 도 2의 네트워크 엘리먼트의 예시적인 동작들을 예시하는 흐름도들이다. 예를 들어, 도 3a-3b는 통신 네트워크에서 캐싱 방법을 수행하기 위한 예시적인 동작들을 도시한다. 도 4-6은 캐싱 방법이 수행된 이후 데이터 파일들을 전달하기 위한 예시적인 동작들을 도시한다.
- [0041] 도 3a-3b가 하기 알고리즘 1에 관련된 캐싱 배포 방법을 수행하기 위한 것이며, 여기서 각각의 데이터 파일 'f'가 유한 필드의 심볼들로서 표현되며 라이브러리 'F'에 속하는 'B'개의 균일-사이즈 패킷들로 분할된다는 것이 이해되어야 한다.

[0042]

알고리즘 1: 캐싱 알고리즘

```

1  for  $f \in F \rightarrow$ 
2    Each user  $u$  caches a subset of
       $p_{f,u}M_uB$  distinct packets of file  $f$ 
      uniformly at random;
3  endfor
4     $M = \{M_{u,f}, \text{ with } u = 1, \dots, n, \text{ and } f = 1, \dots, m\};$ 
5  return( $M$ );

```

[0043]

end Caching algorithm

[0044]

알고리즘 1에서, $\mathbf{p}_u = [p_{u,1}, \dots, p_{u,m}]$ 은 'u' 목적지 디바이스(200)의 캐싱 분포이며, 여기서, $u = 1, \dots, n$ 에 대해 $\sum_{f=1}^m p_{f,u} = 1, \forall u$ 이고, $0 \leq p_{f,u} \leq 1/M_u, \forall f = 1, \dots, m, u = 1, \dots, n$ 이고, 'm'은 네트워크 엘리먼트(151)에 의해 호스팅되는 파일들의 개수이고, 'M_u'는 목적지 디바이스 'u'(즉, 목적지 디바이스(200))에서의 캐시의 저장 용량이고, $M_{u,f} = p_{f,u}M_uB$ 는 사용자 u에서 캐싱되는 파일 f의 패킷들을 나타낸다. 네트워크 엘리먼트(151)는, 목적지 'u'인 디바이스(200)가 파일 'f'의 $M_{u,f} = p_{f,u}M_uB$ 패킷들을 캐싱하도록 알고리즘 1을 수행한다. 또한, 알고리즘 1의 랜덤화된 속성은, 2개의 목적지들이 주어진 파일 'f'에 대해 동일한 개수의 패킷들을 캐싱하는 경우, 2개의 목적지 디바이스(200) 각각이 동일한 파일 'f'의 상이한 패킷들을 캐싱하도록, 네트워크 엘리먼트(151)가 동작들을 수행하는 것을 허용한다. 알고리즘 1은 하기도 3a-3b에 기술되는 동작들에 따라 네트워크 엘리먼트(151)에 의해 구현될 수 있다.

[0045]

도 3a를 참조하면, 동작(300)에서, 네트워크 엘리먼트(151)는 복수의 데이터 파일들에 대한 인기도를 결정할 수 있다. 데이터 파일들은, 예를 들어, 비디오 및/또는 오디오 파일들일 수 있다. 네트워크 엘리먼트(151)는 목적지 디바이스들(200) 중 적어도 하나로부터의 복수의 데이터 파일들에 대한 요청(예를 들어, 사용자 요청)들에 기초하여 인기도를 결정할 수 있다. 사용자 요청들은 데이터 파일들에 대한 주문형 배포를 형성할 수 있다. 네트워크 엘리먼트(151)는 모든 목적지 디바이스들(200)의 주문형 배포에 따라 인기도를 결정할 수 있다. 이러한 경우, 주문형 배포는 Zipf 배포를 따를 수 있다. 대안적으로, 네트워크 엘리먼트(151)는 매 목적지 디바이스 기반으로 인기도를 결정할 수 있고, 여기서 각각의 목적지 디바이스(200)는 연관된 주문형 배포를 가진다.

[0046]

네트워크 엘리먼트(151)는 목적지 디바이스들(200)로부터의 데이터 파일들에 대한 요청들의 수에 기초하여 인기도를 결정할 수 있다. 예를 들어, 네트워크 엘리먼트(151)는 목적지 디바이스들(200)에 의해 100번 요청되는 데이터 파일을 50번 요청되는 데이터 파일보다 더 높은 인기도를 가지는 것으로서 결정한다. 따라서, 인기도는 어느 데이터 파일들이 목적지 디바이스들(200)의 사용자들에 의해 가장 자주 요청되고 시청되는지에 기초할 수 있다.

[0047]

네트워크 엘리먼트(151)는 각각의 데이터 파일을 복수의 패킷들로 분할할 수 있다. 예를 들어, 네트워크 엘리먼트(151)는 각각의 데이터 파일을 동일한 개수의 패킷들(예를 들어, 3개의 패킷들)로 분할할 수 있다. 따라서, 동작(310)에서, 네트워크 엘리먼트(151)는 동작(300)에서 결정된 인기도에 기초하여 복수의 데이터 파일들의 랜덤 패킷들을 적어도 하나의 목적지 디바이스에 송신할 수 있다. 예를 들어, 네트워크 엘리먼트(151)는, 랜덤 패킷들이 각각의 목적지 디바이스(200)에 저장(또는 캐싱)되도록 각각의 데이터 파일의 랜덤 패킷들을 목적지 디바이스들(200)에 송신할 수 있다.

[0048]

네트워크 엘리먼트(151)는 각각의 목적지 디바이스(200)가 결정된 인기도 및 입력 파라미터들(예를 들어, 목적지 디바이스들의 개수, 인기도 분포, 각각의 목적지 디바이스의 캐시 사이즈, 네트워크 엘리먼트(151)에서의 데

이터 파일 라이브러리의 사이즈 등)에 기초하여 데이터 파일들 중 적어도 하나에 대한 주어진 개수의 랜덤 패킷들을 수신하도록, 랜덤 패킷들을 송신할 수 있다. 예를 들어, 네트워크 엘리먼트(151)는, 목적지 디바이스들(200)이 동일한 사이즈의 캐시 및 동일한 주문 분포를 가지는 경우(예를 들어, 목적지 디바이스들이 동일함), 동일한 개수의 패킷들을 각각의 목적지 디바이스(200)에 송신할 수 있다. 일 예에서, 2개의 목적지 디바이스들(1 및 2) 및 10개의 패킷들로 분할되는 2개의 파일들(A 및 B)이 존재한다고 가정한다. (i) 목적지 디바이스들(1 및 2)이 동일한 빈도수로 파일 A 및 파일 B를 요청하고, 파일 A가 파일 B보다 더 큰 빈도수를 가지고 두 목적지들 모두에 의해 요청되는 경우, 그리고 (ii) 2개의 목적지 디바이스들(1 및 2)이 동일한 캐시 사이즈, 예를 들어, 패킷들의 견지에서 6개 유닛들을 가지는 경우, 네트워크 엘리먼트(151)는 양 목적지 디바이스들(1 및 2) 모두가 파일 A의 4개 패킷들 및 파일 B의 2개 패킷들을 캐싱하도록, 캐싱 방법을 수행할 것이다.

[0049] 동작(300)에서 네트워크 엘리먼트(151)가 목적지 디바이스 기반으로 인기도를 결정한 경우, 네트워크 엘리먼트(151)는 동작(310)에서 매 목적지 기반으로 랜덤 패킷들을 송신할 수 있다. 예를 들어, 네트워크 엘리먼트(151)는, 목적지 디바이스들(200)이 상이한 사이즈의 캐시들 또는 상이한 주문 배포들을 가지는 경우 각각의 목적지에 상이한 개수의 패킷들을 송신할 수 있다. 이 경우, 위의 예를 참조하면, 목적지 디바이스(1)는 파일 A의 7개 패킷들 및 파일 B의 3개 패킷들을 수신하는 반면, 목적지 디바이스(2)는 파일 A의 2개 패킷들 및 파일 B의 5개 패킷들을 수신할 수 있다. 이는 목적지 디바이스(1)가 파일 B보다 훨씬 더 많이 파일 A를 요청하고, 패킷들의 견지에서 10개 유닛들의 전체 캐시 사이즈를 가지는 반면, 목적지 디바이스(2)가 파일 B보다 훨씬 더 많이 파일 A를 요청하며 패킷들의 견지에서 7개 유닛들의 전체 캐시 사이즈를 가진다는 사실로 인한 것일 수 있다.

[0050] 도 3b는, 원하는 경우, 동작들(300 및 310) 사이에서 수행될 수 있는 네트워크 엘리먼트(151)의 예시적인 동작들을 예시한다. 예를 들어, 도 3a의 동작(300)에 후속하여, 네트워크 엘리먼트(151)는 결정된 인기도에 기초하여 데이터 파일들을 순위 매길 수 있다. 예를 들어, 동작(301)에서, 네트워크 엘리먼트(151)는 동작(300)에서 결정된 인기도를 사용하여 가장 인기있는 데이터 파일로부터 가장 덜 인기있는 데이터 파일로 데이터 파일들을 순위 매길 수 있다.

[0051] 동작(302)에서, 네트워크 엘리먼트(151)는, 각각의 데이터 파일에 대해, 순위에 기초하여 다수의 랜덤 패킷들을 선택할 수 있다. 예를 들어, 네트워크 엘리먼트(151)는 각각의 데이터 파일의 각자의 순위 및 네트워크의 입력 파라미터들(예를 들어, 목적지 디바이스들의 개수, 인기도 분포, 각각의 목적지 디바이스의 캐시 사이즈, 네트워크 엘리먼트(151)에서의 데이터 파일 라이브러리의 사이즈 등) 중 적어도 하나에 따라 각각의 목적지 디바이스에 대해 그리고 데이터 파일들 각각에 대해, 상이한 개수의 랜덤 패킷들을 선택한다. 동작(302) 이후, 네트워크 엘리먼트(151)는 도 3a의 동작(310)으로 역으로 진행하여, 각각의 데이터 파일에 대해 선택된 개수의 랜덤 패킷들을 송신할 수 있다.

[0052] 동작(302)이, 네트워크 엘리먼트(151)가 순위 매겨진 데이터 파일들을 적어도 하나의 임계값에 기초하여 적어도 제1 서브세트 및 제2 서브세트로 분할하는 것을 포함할 수 있다는 것이 이해되어야 한다. 적어도 하나의 임계값은 실험적 증거에 기초하고 그리고/또는 사용자 정의될 수 있다. 제1 서브세트는 제2 서브세트보다 더 높게 순위 매겨진 데이터 파일들을 포함할 수 있다. 따라서, 동작(310)에서, 네트워크 엘리먼트(151)는 제1 서브세트 내의 데이터 파일들에 대해서만 선택된 개수의 랜덤 패킷들을 송신할 수 있다. 이는 목적지 디바이스들(200)에서 패킷들의 더욱 효율적인 캐싱을 허용할 수 있다.

[0053] 도 3a 및 3b에 관련하여 기술되는 동작들이, 더 인기 있는 파일들의 더 많은 패킷을 캐싱하고, 그리고 목적지 디바이스들(200)에 의해 총체적으로 캐싱되는 각각의 파일의 별개의 패킷들의 양을 증가(또는 대안적으로, 최대화)시키고, 그리고 종래의 멀티캐스팅 기법들에 비해 전송들의 수를 감소시키는 방식의 능력으로 인해 네트워크의 개선된 성능을 허용한다는 것이 이해되어야 한다.

[0054] 전술된 캐싱 방법들과 함께, 이 출원은 데이터 파일들의 요청된 패킷들이 목적지 디바이스들(200)에 전달되는 전달 페이즈에 대한 예시적인 방법들을 개시한다. 전달 페이즈에 대한 예시적인 방법은 그래프 이론에 기초한다.

[0055] 도 4는 적어도 일 예시적인 실시예에 따른 전달 페이즈에 대한 예시적인 동작들을 예시한다. 도 4에 관련하여, 동작(400)에서, 네트워크 엘리먼트(151)는 데이터 파일들의 패킷들에 대한 요청들을 목적지 디바이스들(200)(또는 사용자 디바이스들)로부터 수신한다. 네트워크 엘리먼트(151)가 전술된 캐싱 방법을 이미 수행했기 때문에, 각각의 목적지 디바이스(200)는 캐싱 방법의 결과로서 캐싱(또는 저장)되지 않은 패킷들만을 요청한다. 따라서, 전달 페이즈는 요청된 파일들의 유실 부분(들), 즉, 목적지 디바이스(200)의 메모리로부터 유실되는 패

킷들을 각각의 목적지 디바이스(200)에 제공하는 것으로 구성된다.

- [0056] 동작(410)에서, 네트워크 엘리먼트(151)는 충돌 그래프를 구성한다. 예를 들어, 네트워크 엘리먼트(151)는, 각각의 목적지 디바이스(200)에 의해 요청되는 각각의 패킷이 충돌 그래프의 복수의 꼭지점들 내의 별개의 꼭지점에 의해 표현되도록 복수의 꼭지점들로 충돌 그래프를 점유한다. 따라서, 동일한 패킷이 K개의 상이한 사용자들에 의해 요청되더라도, 패킷은 충돌 그래프에서 K개의 상이한 꼭지점들로서 표현된다. 다시 말해, 충돌 그래프 내의 각각의 꼭지점은 목적지 디바이스(200) 및 요청된 패킷의 고유한 쌍과 연관된다. 따라서, 충돌 그래프의 각각의 꼭지점이 목적지 디바이스(200)와 연관되며, 목적지 디바이스(200)에 의해 요청되는 패킷을 나타낸다고 할 수 있다. 또한, 네트워크 엘리먼트(151)는 복수의 꼭지점들 중 어느 것이 동일한 요청된 패킷을 나타내며, 어느 요청된 패킷들이 목적지 디바이스(200)에 속하는 캐시들에 저장되는지에 기초하여 충돌 그래프를 구성할 수 있다. 동작(410)은 도 5 및 도 6을 참조하여 하기에 더 상세히 기술된다.
- [0057] 여전히 도 4를 참조하면, 동작(420)에서, 네트워크 엘리먼트(151)는 복수의 꼭지점들을 원래 레벨들에 할당할 수 있다. 각각의 원래 레벨은 패킷들에 대한 요청들의 수 및 패킷들을 저장하는 사용자 목적지(200) 캐시들의 수를 표시할 수 있다. 예를 들어, 각각의 레벨은 패킷을 요청하는 사용자 목적지들의 수와 요청된 패킷을 저장하는 사용자 디바이스 캐시들의 수의 합산일 수 있다. 따라서, 패킷이 4개의 목적지 디바이스(200)에 의해 요청되고, 목적지 디바이스들(200) 중 3개가 패킷을 저장하고 있는 경우, 그 패킷과 연관된 꼭지점에 대한 원래 레벨은 3+4=7이다.
- [0058] 여전히 도 4를 참조하면, 동작(420)에서, 네트워크 엘리먼트(151)는 또한 라벨들을 복수의 꼭지점들에 할당할 수 있다. 각각의 라벨은 목적지 디바이스들(200)이 꼭지점에 대응하는 패킷을 요청하는 것, 및 목적지 디바이스(200) 캐시들이 패킷을 저장하는 것을 표시할 수 있다. 예를 들어, 각각의 라벨은 패킷을 요청하는 목적지 디바이스들(200)에 그리고 요청되는 패킷을 저장하는 목적지 디바이스들(200)에 대응하는 인덱스들의 세트일 수 있다. 예를 들어, 패킷이 4개의 목적지 디바이스들(200)(예를 들어, 사용자 1, 사용자 3, 사용자 5, 사용자 10)에 의해 요청되고, 목적지 디바이스들(200) 중 3개(예를 들어, 사용자 2, 사용자 4, 사용자 8)가 요청되는 패킷을 저장하는 경우, 요청되는 패킷과 연관된 꼭지점에 대한 인덱스들의 세트는 $K_{v_1} = \{1,3,5,10;2,4,8\}$ 이다. 여기서, 라벨의 길이는 7이다(즉, 패킷을 요청하는 디바이스들의 개수와 패킷을 저장하는 목적지 디바이스 캐시들의 수의 합산).
- [0059] 동작(430)에서, 네트워크 엘리먼트(151)는 할당된 레벨들에 기초하여 꼭지점들을 순서화할 수 있다. 예를 들어, 네트워크 엘리먼트(151)는 컬러링 동작을 위한 준비로 가장 높은 레벨로부터 가장 낮은 레벨로 꼭지점들을 순서화할 수 있다.
- [0060] 동작(445)에서, 네트워크 엘리먼트(151)는 충돌 그래프 상에서 요청된 패킷들을 라벨링하는 방식으로 복수의 꼭지점들을 컬러링한다. 예를 들어, 네트워크 엘리먼트(151)는 동작(430)에서 결정되는 순서에 기초하여 복수의 꼭지점들을 컬러링한다. 동작(445)은 도 6에 관련하여 하기에 더 상세하게 논의된다.
- [0061] 동작(480)에서, 네트워크 엘리먼트(151)는 동일한 컬러를 가지는 꼭지점들에 의해 표현되는 요청된 패킷들을 결합시킨다. 예를 들어, 네트워크 엘리먼트(151)는 동일한 컬러를 가지는 꼭지점들에 의해 표현되는 패킷들에 대해 배타적-OR(XOR) 연산들(또는 유한 필드 상의 다른 선형 조합 연산들)을 수행한다.
- [0062] 동작(490)에서, 네트워크 엘리먼트(151)는 결합된 패킷들을 송신한다. 예를 들어, 네트워크 엘리먼트(151)는 멀티캐스트 전송을 통해 목적지 디바이스들(200)에 결합된 패킷들을 송신한다. 전송 이전에 패킷들을 결합시킴으로써 적어도 일 예시적인 실시예에 따른 전달 방법이, 네트워크 엘리먼트(151)의 전송들의 수를 감소시킬 수 있고, 이것이 소모를 감소시키고 네트워크 효율성을 증가시킬 수 있다는 것이 이해되어야 한다. 목적지 디바이스들(200)이 XOR 연산들의 세트(또는 다른 선형 조합들의 세트)를 사용하여 전송된 결합된 패킷들을 수신하고 디코딩할 수 있다는 것이 이해되어야 한다. 예를 들어, 목적지 디바이스들(200)은 그것의 캐싱된 패킷들을 키로서 사용하여 결합된 패킷들을 디코딩할 수 있다.
- [0063] 도 5는 적어도 일 예시적인 실시예에 따라 충돌 그래프를 구성하기 위한 예시적인 동작들을 예시한다. 예를 들어, 도 5는 도 4로부터의 동작(410)을 더 상세하게 논의한다. 도 5는 무방향 충돌 그래프를 구성하기 위한 예시적인 동작들을 예시한다. 무방향 충돌 그래프는 이 출원에서 계층적 그리디 컬러링(hierarchical greedy coloring)(HgC) 방식이라 지칭되는 컬러링 방식과 함께 사용될 수 있다. 이 컬러링 방식은 도 6에 관련하여 하기에 더 상세히 기술된다.

- [0064] 도 5를 참조하면, 동작(511)에서, 네트워크 엘리먼트(151)는 동작(410)에서 채워지는(populate) 복수의 꼭지점들 중 2개의 꼭지점들(V_i' 및 V_j')을 분석한다. 동작(513)에서, 네트워크 엘리먼트가 꼭지점들(V_i 및 V_j)이 동일한 요청된 패킷을 나타낸다고 결정하는 경우, 네트워크 엘리먼트(151)는 동작(515)에서 2개의 꼭지점들 사이에 링크(또는 에지)를 생성하지 않는다. 이후, 네트워크 엘리먼트(151)는 동작(531)으로 진행하여 충돌 그래프 내의 모든 꼭지점들이 분석되었는지를 결정한다.
- [0065] 동작(513)에서, 네트워크 엘리먼트(151)가 꼭지점들(V_i 및 V_j)이 동일한 요청된 패킷을 나타내지 않는다고 결정하는 경우, 네트워크 엘리먼트(151)는 동작(517)으로 진행하여, 꼭지점(V_i)에 의해 표현된 패킷을 요청하는, 꼭지점(V_i)과 연관된 목적지 디바이스(200)의 캐시(또는 메모리)를 체크한다. 동작(519)에서, 네트워크 엘리먼트(151)가 꼭지점(V_j)을 나타내는 패킷이 꼭지점(V_i)에 의해 표현된 패킷을 요청하는 목적지 디바이스(200)의 캐시에서 이용가능하지 않다고 결정하는 경우, 네트워크 엘리먼트(151)는 동작(521)에서 꼭지점(V_i)과 꼭지점(V_j) 사이에 링크를 생성한다. 이후, 네트워크 엘리먼트(151)는 동작(531)으로 진행하여, 충돌 그래프 내의 모든 꼭지점들이 분석되었는지를 결정한다.
- [0066] 동작(519)에서, 네트워크 엘리먼트(151)가, 꼭지점(V_j)을 나타내는 패킷이 꼭지점(V_i)과 연관되며 꼭지점(V_i)에 의해 표현되는 패킷을 요청하는 목적지 디바이스(200)의 캐시에서 이용가능하다고 결정하는 경우, 네트워크 엘리먼트(151)는 동작(523)에서 꼭지점(V_j)과 연관되며 꼭지점(V_j)에 의해 표현되는 패킷을 요청하는 목적지 디바이스(200)의 캐시를 체크한다. 동작(525)에서, 꼭지점(V_i)을 나타내는 패킷이 꼭지점(V_j)과 연관되며 꼭지점(V_j)에 의해 표현되는 패킷을 요청하는 목적지 디바이스(200)의 캐시에서 이용가능하지 않은 경우, 네트워크 엘리먼트(151)는 동작(431)으로 진행하여 충돌 그래프 내의 모든 꼭지점들이 분석되었는지를 결정하기 이전에 동작(527)에서 꼭지점들(V_i 및 V_j) 사이에 링크를 생성한다.
- [0067] 동작(525)에서, 네트워크 엘리먼트(151)가, 꼭지점(V_i)을 나타내는 패킷이 꼭지점(V_j)과 연관되며 꼭지점(V_j)에 의해 표현되는 패킷을 요청하는 목적지 디바이스(200)의 캐시에서 이용가능하다고 결정하는 경우, 네트워크 엘리먼트(151)는 동작(529)에서 꼭지점들(V_i 및 V_j) 사이에 링크를 설정한다. 이후, 네트워크 엘리먼트(151)는 동작(531)으로 진행하여, 충돌 그래프 내의 모든 꼭지점들이 분석되었는지를 결정한다.
- [0068] 일단 네트워크 엘리먼트(151)가 충돌 그래프 내의 모든 꼭지점들을 분석하면, 네트워크 엘리먼트(151)는 동작(533)에서 구성된 충돌 그래프를 리턴시킨다.
- [0069] 도 5의 견지에서, 도 4의 구성 동작(410)이, (i) 제1 꼭지점과 제2 꼭지점이 동일한 패킷을 나타내지 않고, (ii) 제1 꼭지점에 의해 표현되는 패킷이 제2 꼭지점과 연관되며 제2 꼭지점에 의해 표현되는 패킷을 요청하는 사용자 디바이스의 캐시에 저장되지 않거나, 또는 제2 꼭지점에 의해 표현되는 패킷이 제1 꼭지점과 연관되며 제1 꼭지점에 의해 표현되는 패킷을 요청하는 사용자 디바이스의 캐시에 저장되지 않는 경우, 복수의 꼭지점들 중 제1 꼭지점과 제2 꼭지점 사이에 링크를 생성하는 동작에 의해 요약될 수 있다는 것이 이해되어야 한다.
- [0070] 도 6은 적어도 일 예에 따라 충돌 그래프를 컬러링하기 위한 예시적인 동작들을 예시한다. 도 6의 동작들은 하기에 논의되는 알고리즘에 관련된다.

Algorithm 1 HgLC₁

```

1:  $\mathcal{C} = \emptyset$ ;
2:  $c = \emptyset$ ;
3: choose  $a \in [0, 1]$ ;
4: choose  $b \in [0, 1]$ ;
5: for all  $i = n, n-1, \dots, 2, 1$  do
6:   for all  $v \in G_i$  and  $|K_v| = i$  do
7:      $\mathcal{I} = \{v\}$ ;
8:     for all  $v' \in G_i \setminus \mathcal{I}$  with  $|K_{v'}| = |K_v|$  do
9:       if {There is no edge between  $v'$  and  $\mathcal{I}$ } then
10:         $\mathcal{I} = \mathcal{I} \cup v'$ ;
11:       end if
12:     end for
13:     if  $|\mathcal{I}| = i$  then
14:       Color all the vertices in  $\mathcal{I}$  by  $c \notin \mathcal{C}$ ;
15:        $c[\mathcal{I}] = c$ ,  $\mathcal{C} = \mathcal{C} \cup c$ ;
16:        $G_i = G_i \setminus \mathcal{I}$ ;
17:     end if
18:   end for
19:   for all Randomly pick a  $v \in W_1 \subset G_i$ , with
20:      $W_1 = \{v \in G_i : \min_{u \in Q_i} |K_u| \leq |K_v| \leq \min_{u \in Q_i} |K_u| + [a \cdot (\max_{u \in Q_i} |K_u| - \min_{u \in Q_i} |K_u|)]\}$  do
21:      $\mathcal{Q}_i = G_i \setminus \mathcal{I}$ ;
22:     for all Randomly pick a  $v' \in W_2 \subset Q_i$ , with
23:        $W_2 = \{v' \in Q_i : \min_{u' \in Q_i} |K_{u'}| \leq |K_{v'}| \leq \min_{u' \in Q_i} |K_{u'}| + [b \cdot (\max_{u' \in Q_i} |K_{u'}| - \min_{u' \in Q_i} |K_{u'}|)]\}$  do
24:         if {There is no edge between  $v'$  and  $\mathcal{I}$ } then
25:            $\mathcal{I} = \mathcal{I} \cup v'$ ;
26:            $Q_i = Q_i \setminus \{v'\}$ ;
27:         else
28:            $Q_i = Q_i \setminus \{v'\}$ ;
29:         end if
30:       end for
31:       if  $|\mathcal{I}| \geq i$  then
32:         Color all the vertices in  $\mathcal{I}$  by  $c \notin \mathcal{C}$ ;
33:          $c[\mathcal{I}] = c$ ,  $\mathcal{C} = \mathcal{C} \cup c$ ;
34:          $G_i = G_i \setminus \mathcal{I}$ ;
35:       else
36:          $G_i = G_i \setminus \{v\}$ ,  $G_{i-1} = G_{i-1} \cup \{v\}$ ;
37:       end if
38:     end for
39:   end for
40: end for
41:  $c = \text{LocalSearch}(\mathcal{H}_C, W, c, \mathcal{C})$ ;

```

[0071]

[0072]

위의 알고리즘("HgC 알고리즘"으로 지칭됨)에서, $f(v)$ 를 충돌 그래프에서 꼭지점(v)에 의해 표현되는 패킷이라고 하자. U 를 사용자 목적지 디바이스들의 세트들로서 정의하고, $K_v = \{\forall u \in U : f(v) \in Q_u \cup C_u\}$ 이며, 여기서, Q_u 는 사용자 목적지 디바이스(u)에 의해 요청되는 모든 패킷들의 세트이고, C_u 는 사용자 목적지 디바이스(u)에 의한 모든 캐싱된 패킷들의 세트이다. K_v 를 꼭지점 " v "에 연관된 라벨이라 지칭한다. 예를 들어, 패킷 A_1 ($f(v_1) = A_1$)을 요청하는 꼭지점 v_1 이 사용자 목적지 디바이스(1) 및 사용자 목적지 디바이스(2)에 의해 요청되고, 사용자 목적지 디바이스(3) 및 사용자 목적지 디바이스(4)에서 캐싱되는 경우, $K_{v_1} = \{1,2;3,4\}$ 이다.

[0073]

$G_i = \{v : |K_v| = i\}$ 라 하자. G_i 를 원래 계층들(또는 레벨들)로서 간주한다. 우리는 가장 높은 계층 또는 레벨인 계층(또는 레벨) n 으로부터 시작한다. 먼저, 이들이 동일한 $|K_v|=n$ 을 가지고, 이러한 세트의 카디널리티가 n 과 동일하며, 충돌 그래프 내에 이러한 세트의 임의의 2개 꼭지점들 사이에 링크가 존재하지 않는 경우(즉, 그리디 방식으로 사이즈 " n "을 가지는 독립 세트에 대한 탐색), G_n 내의 꼭지점들의 서브세트를 동일한 컬러로 컬러링한다. (동일한 $|K_v|$ 를 가지며, 충돌 그래프 내의 링크에 의해 접속되지 않는 2개의 노드들은 이들이 동일한 K_v 를 가진다는 선언과 등가라는 점에 주목한다.) 이후, G_n 내의 컬러링되지 않은 꼭지점들의 나머지

를 G_{n-1} 과 병합하여(즉, $G_{n-1} \equiv G_{n-1} \cup G_n$, HgC 알고리즘의 라인 35) 새로운 계층(또는 레벨) $(n-1)$ 을 초래한다.

[0074]

계층(또는 레벨) $(n-1)$ 에서, $|K_v| = n-1$ 이도록 하는 모든 K_v 에 대해, 이들이 이들이 $|K_v| = n-1$ 을 가지고, 이러한 세트의 카디널리티가 n 과 같으며, 이러한 세트의 임의의 2개의 꼭지점들 사이에 링크들이 존재하지 않는 경우,

먼저, G_{n-1} 내의 꼭지점들의 서브셋을 동일한 컬러로 컬러링한다. 이후, G_{n-1} 내의 꼭지점들의 나머지를 컬러링하려고 시도한다.

[0075] 우리가 G_{n-1} 내의 꼭지점들의 나머지를 컬러링하기 위해 사용하는 기준은 다음과 같다: 먼저, HgC 알고리즘의 라인 19에 보여지는 $W_1 \in G_{n-1}$ 로부터 꼭지점(v)을 랜덤으로 선택하며, 여기서, W_1 은 "작은" $|K_v|$, $v \in G_{n-1}$ 또는 H_{CQ} 내의 "큰 정도"를 가지고 꼭지점들의 세트를 표기하고, 여기서, 값 $a \in [0,1]$ 은 W_1 의 크기를 제어한다. 예를 들어, $a = 0$ 인 경우, W_1 은 가장 작은 $|K_v|$, $v \in G_{n-1}$ 을 가지는 꼭지점을 표기한다.

[0076] 둘째, 선택된 꼭지점(v) 및 꼭지점들 $v' \in W_2 \subseteq G_i \setminus \{v\}$ 을 동일한 컬러로 컬러링하려고 시도하며, 여기서, W_2 는, $v' \in G_i \setminus \{v\}$ 인 그것의 $|K_{v'}|$ 가 그리디 방식으로 $|K_v|$ 에 "가까운" "i" 레벨에서의 컬러링되지 않은 꼭지점들의 세트로서 정의된다. 파라미터 $a \in [0,1]$ 와 유사하게, 이러한 근접성은 알고리즘 내의 라인 22에 도시된 바와 같이 또다른 파라미터 $b \in [0,1]$ 에 의해 캡처된다. 예를 들어, $b = 0$ 인 경우, $|K_{v'}| - |K_v|$ 가 최소화되도록 꼭지점(v')으로부터 시작한다. 여기서, 그리디 방식으로, 제i 계층(또는 레벨) 내의 적어도 "i"의 크기를 가지는 독립 세트를 찾는다.

[0077] 이러한 제2 컬러링 절차 이후, 컬러링되지 않은 꼭지점들을, 이 경우 G_{n-2} 인 다음 계층(또는 레벨)의 꼭지점들과 연합시킨다. 이후, 모든 계층들(또는 레벨들)에 대해 동일한 절차를 반복한다.

[0078] 마지막으로, HgC 알고리즘의 라인 39에서 최종 충돌 그래프에 대해 사용되는 컬러들의 개수를 추가로 감소시키도록 LocalSearch라 명명되는 함수를 사용한다. LocalSearch의 상세항목들은 하기의 알고리즘에 의해 주어진다.

Algorithm 2 LocalSearch(H_G, W, c, C)

```

1: for all  $c \in C$  do
2:   Let  $\mathcal{J}_c$  be the set of vertices whose color is  $c$ ;
3:   Let  $\mathcal{B} = \emptyset$ ;
4:   Let  $\hat{c} = c$ ;
5:   for all  $i \in \mathcal{J}$  do
6:      $\mathcal{A} = \emptyset$ ;
7:     for all  $j \in \mathcal{N}(i)$  do
8:        $\mathcal{A} = \mathcal{A} \cup c[i]$ ;
9:       if  $C \setminus \mathcal{A} \neq \emptyset$  then
10:         $c'$  is randomly picked from  $C \setminus \mathcal{A}$ ;
11:         $c[i] = c'$ ;
12:         $\mathcal{B} = \mathcal{B} \cup \{i\}$ ;
13:       end if
14:     end for
15:     if  $|\mathcal{B}| = |\mathcal{J}_c|$  then
16:        $c = \hat{c}$ ;
17:        $C = C \setminus c$ ;
18:     end if
19:   end for
20: end for
21: return  $c$ ;

```

[0079] 여기서, $\mathcal{N}(j)$ 가 꼭지점 "j"의 이웃 꼭지점들(꼭지점 "j"를 배제시킴)을 표기한다고 하자. 명료함을 위해, A_1 을 요청하는 사용자 디바이스와 A_1 을 캐싱하는 사용자 목적지 디바이스들을 분리하기 위해 썬표를 사용한다. 또한, c 는 컬러링을 보여주는 원하는 벡터이다. HgC 알고리즘의 복잡도는 $O(n^3 B^2)$ 이다.

[0081] 도 6에 예시되는 동작들이 $a=0$ 및 $b=0$ 에 대한 HgC 알고리즘에 관련되며, 도 4의 동작(445)의 일부로서 수행되는 동작들에 대응한다는 것이 이해되어야 한다. 초기에, 네트워크 엘리먼트(151)는 'V'를 충돌 그래프 내의 꼭지점들로서 정의하고, V_1 을 컬러링된 꼭지점들의 세트로서, 그리고 V_2 를 컬러링되지 않은 꼭지점들의 세트로서

정의한다. 컬러링 동작들이 시작하기 전에, 세트 V_1 은 비어 있고, $V_2 = V$ 이다.

- [0082] 동작(600)에서, 네트워크 엘리먼트(151)는 레벨 $j = n$ 을 초기화시키며, 여기서, n 은 할당되는 레벨들 중에서 가장 높은 레벨을 표기한다(동작(430)에서 꼭지점들이 이들의 할당된 레벨들에 기초하여 순서화되었음을 상기하라). 동작(605)에서, 네트워크 엘리먼트(151)는 분석을 위해 레벨 'j'을 선택한다.
- [0083] 동작(610)에서, 네트워크 엘리먼트(151)는 j 와 동일한 라벨 길이를 가지는 레벨 j 에 속하는 꼭지점 'v'을 선택하고, 그 꼭지점을 분석된 것으로서 마킹한다. 동작(615)에서, 네트워크 엘리먼트(151)는 꼭지점들의 세트 'L'를 형성한다. 네트워크 엘리먼트(151)는 i) 꼭지점(v)과 동일한 라벨 길이를 가지며(즉, 동일한 $|K_v|$ 을 가지며), ii) 컬러링되지 않고, iii) 서로 링크되지 않고, iv) 꼭지점(v)에 링크되지 않는, 레벨 j 내의 꼭지점들을 가지는 세트(L)를 형성한다. 동작(620)에서, 네트워크 엘리먼트(151)는 (꼭지점 v를 포함하는) 세트 L 내의 꼭지점들의 수가 레벨 j 의 수와 동일하지는 않는 경우, 네트워크 엘리먼트(151)는 컬러를 세트(L) 내의 꼭지점들에 할당하지 않으며, 동작(635)으로 진행하여, 레벨 j 내의 모든 꼭지점들이 분석되었는지를 체크한다. 세트(L) 내의 꼭지점들의 개수가 j 와 같은 경우, 네트워크 엘리먼트(151)는 동작(625)에서 새로운 컬러(예를 들어, 충돌 그래프에서 아직 사용되지 않은 컬러)를 선택한다. 동작(630)에서, 네트워크 엘리먼트(151)는 새로운 컬러를 세트(L) 내의 꼭지점들에 할당하며, 세트(V_1) 내에 세트(L) 내의 꼭지점들을 포함시키고, 세트(V_2)로부터 세트(L) 내의 꼭지점들을 제거한다.
- [0084] 이후, 동작(635)에서, 네트워크 엘리먼트(151)는 j 의 라벨 길이를 가지는 레벨 j 에서의 모든 꼭지점들이 분석되었는지를 결정한다. 분석되었다고 결정하지 않은 경우, 네트워크 엘리먼트(151)는 동작(605)으로 리턴한다. 분석되었다고 결정하는 경우, 네트워크 엘리먼트(151)는 동작(640)으로 진행한다.
- [0085] 동작(640)에서, 네트워크 엘리먼트(151)는 레벨 j 내의 어느 꼭지점들이 컬러링되지 않았는지를 결정하고, 이러한 컬러링되지 않은 꼭지점들을 분석되지 않은 것으로서 마킹하고, 그것의 라벨이 레벨 j 보다 더 크거나 같은 길이를 가지는 꼭지점 'w'을 선택하고, 꼭지점 'w'을 분석된 것으로서 마킹한다. 이는 $a = 0$ 일 때, G_{n-1} 내의 꼭지점들의 나머지의 컬러링과 관련한 위의 기재와 등가이다.
- [0086] 동작(645)에서, 네트워크 엘리먼트(151)는 세트(L)를 빈 세트로 리셋시키고, 서로 링크되지 않으며 꼭지점 "w"에 링크되지 않은 j 에 가장 가까운(또는 더 크거나 같은) 길이를 가지는 라벨을 가지는 레벨 j 내의 여전히 컬러링되지 않은 꼭지점들을 포함하도록 세트(L)를 형성한다. 동작(650)에서, 네트워크 엘리먼트(151)는 세트(L) 내의 꼭지점들의 수(꼭지점 "w"를 포함함)가 " j "보다 더 크거나 같은지를 결정한다. 그렇지 않은 경우, 네트워크 엘리먼트(151)는 어떠한 꼭지점들도 컬러링하지 않고, 동작(665)으로 진행하여, 레벨 j 에 있는 모든 꼭지점들이 분석되었는지를 결정한다.
- [0087] 동작(650)에서, 네트워크 엘리먼트(151)가 세트(L) 내의 꼭지점들의 전체 수가 j 보다 더 크거나 같다고 결정하는 경우, 네트워크 엘리먼트(151)는 동작(655)에서 새로운 컬러(예를 들어, 충돌 그래프에서 아직 사용되지 않은 컬러)를 선택한다. 동작(660)에서, 네트워크 엘리먼트(151)는 세트(L) 내의 꼭지점들에 새로운 컬러를 할당하고, 세트(V_1)에 세트(L) 내의 꼭지점들을 포함시키고, 세트(V_2)로부터 세트(L) 내의 꼭지점들을 제거한다. 동작(665)에서, 네트워크 엘리먼트(151)는 레벨 j 에서의 모든 꼭지점들이 분석되었는지를 결정한다. 분석되지 않았다고 결정되는 경우, 네트워크 엘리먼트(151)는 동작(640)으로 리턴한다. 분석되었다고 결정되는 경우, 동작(668)에서, 네트워크 엘리먼트(151)는 레벨 j 내의 모든 컬러링되지 않은 꼭지점들을 다음 더 낮은 레벨로 이동시켜, 레벨 $j-1$ 에 원래 속하는 모든 꼭지점들과 레벨 j 에 속하는 컬러링되지 않은 꼭지점들과의 연합에 의해 주어지는, 새로운 레벨 $j-1$ 을 생성한다. 다음으로, 동작(670)에서, 네트워크 엘리먼트(151)는 레벨(j)이 1(또는 할당된 레벨들 중에서 가장 낮게 할당된 레벨)과 같은지를 결정한다.
- [0088] 레벨 j 가 1(또는 할당된 레벨들 중에서 가장 낮게 할당된 레벨)과 같은 경우, 네트워크 엘리먼트(151)는 동작(677)으로 진행한다. 그렇지 않은 경우, 동작(675)에서, 네트워크 엘리먼트(151)는 레벨 j 를 $j-1$ 로 설정하고, 동작(605)으로 리턴한다. 동작(675)에서, 충돌 그래프의 모든 꼭지점들에 컬러가 할당됨을 보장하기 위해, 네트워크 엘리먼트(151)가 레벨 j 에 남아 있는 임의의 컬러링되지 않은 꼭지점들을 다음 가장 낮은 레벨(즉, $j-1$)로 배치한다는 것이 이해되어야 한다. 동작(677)에서, 네트워크 엘리먼트는 충돌 그래프에서 사용되는 컬러들의 전체 개수를 감소시키려는 시도로 로컬 탐색을 수행한다. 예를 들어, 네트워크 엘리먼트(151)는 충돌 그래프의 기존의 컬러들로부터 한 컬러를 선택하고, 선택된 컬러를 가지는 꼭지점들을 식별하고, 식별된 꼭지점들에 링크된 꼭지점들이 상이한 컬러로 컬러링되지 않은 경우, 선택된 컬러를 기존의 컬러들로부터 선택된 상이한 컬러로 대체한다. 따라서, 선택된 컬러는 기존의 컬러들의 세트로부터 제거된다. 이후, 네트워크 엘리먼트

(151)는 680에서 컬러링된 충돌 그래프를 해(solution)로서 리턴시킨다.

[0089] 도 6에 도시된 동작들의 견지에서, 도 6의 동작들이, 네트워크 엘리먼트(151)가 가장 높은 레벨을 가지는 컬러링되지 않은 꼭지점을 선택하는 것, 및 (i) 서브세트 내의 꼭지점들이 원하는 길이를 가지는 할당된 레벨을 가지고 — 원하는 길이는 입력 파라미터들 및 선택된 꼭지점들의 라벨의 길이에 기초함 —, (ii) 서브세트 내의 꼭지점들이 선택된 꼭지점에 대한 링크를 가지지 않고, (iii) 서브세트 내의 꼭지점들이 서로 사이에 링크를 가지지 않고, (iv) 서브세트의 카디널리티가 레벨의 값 마이너스 1과 같은 경우, 선택된 꼭지점에 그리고 복수의 꼭지점들 내의 컬러링되지 않은 꼭지점들의 서브세트에 동일한 컬러를 할당하는 것에 대응할 수 있다는 것이 또한 이해되어야 한다. 도 6의 동작들이 네트워크 엘리먼트(151)가, 가장 높은 레벨의 모든 꼭지점들이 선택될 때까지 동일한 컬러 동작들을 선택하고 할당하는 것을 반복적으로 수행하는 것, 및 여전히 컬러링되지 않은 가장 높은 레벨의 꼭지점들을 가장 높은 레벨 아래의 다음 레벨로 업데이트하는 것에 대응한다는 것이 추가로 이해되어야 한다. 추가로, 도 6의 동작들이, 충돌 그래프의 모든 꼭지점들이 컬러링될 때까지 네트워크 엘리먼트(151)에 의해 수행된다는 것이 이해되어야 한다. 레벨들이 또한 계층들로서 지칭될 수 있으며, 레벨(또는 계층) "j"에 할당되는 꼭지점들의 세트가 G_j 로 표기되며, 따라서, 레벨 j 내에 남아 있는 모든 컬러링되지 않은 꼭지점들을 레벨 j-1로 배치하는 동작이 세트 G_{j-1} 을 형성하는 것과 등가라는 것이 이해되어야 한다. G_{j-1} 을 형성하는 이 동작이 동작(668)과 등가라는 것이 이해되어야 한다.

[0090] 도 7은 적어도 일 예시적인 실시예에 따른 컬러링된 충돌 그래프를 예시한다. 예를 들어, 도 7은 도 5에 따라 구성되며, HgC 방식 및 도 6의 동작들에 따라 컬러링되는 충돌 그래프를 예시한다.

[0091] 도 7에서, $n = 3$ 개 목적지 디바이스들(또는 사용자 디바이스들)(200)(즉, $U = \{1, 2, 3\}$), 및 $m = 3$ 개 데이터 파일들(A, B, C)(즉, $F = \{A, B, C\}$)을 가지는 공유형 링크 네트워크를 고려한다. 각각의 데이터 파일은 4개의 패킷들로 분할된다. 예를 들어, 데이터 파일(A)은 4개의 패킷들 $\{A1, A2, A3, A4\}$ 을 가진다. 캐싱 페이지에 대해(즉, 도 3a 및 3b, 및 관련 기재를 참조), 사용자 디바이스 1이 $\{A1, B1, B2, B3, B4, C2\}$ 를 캐싱하고, 사용자 디바이스 2가 $\{A1, A2, A3, A4, B1, C2, C3\}$ 를 캐싱하고, 사용자 디바이스 3이 $\{A1, A2, A3, B1, B2, B3\}$ 를 캐싱한다고 하자. 이후, 사용자 디바이스들 $\{1, 2, 3\}$ 은 각자 데이터 파일들 $\{A, B, C\}$ 을 요청한다. 등가적으로, 사용자 디바이스 1은 A2, A3, A4를 요청하고; 사용자 디바이스 2는 B2, B3, B4를 요청하고; 사용자 디바이스 3은 C1, C2, C3, C4를 요청한다. 이후, $K_{A2} = \{1, 2, 3\}$ (침표는 패킷 A2를 캐싱한 사용자 디바이스들로부터 패킷 A2를 요청하는 사용자 디바이스를 분리시키기 위해 사용된다. 여기서, 패킷 A2는 사용자 디바이스 1에 의해 요청되는 반면, 패킷 A2는 사용자 디바이스들 2 및 3에 의해 캐싱된다); $K_{A3} = \{1, 2, 3\}$; $K_{A4} = \{1, 2\}$; $K_{B2} = \{2, 1, 3\}$; $K_{B3} = \{2, 1, 3\}$; $K_{B4} = \{2, 1\}$; $K_{C2} = \{3, 1, 2\}$; $K_{C3} = \{3, 2\}$; $K_{C4} = \{3\}$ 을 가진다(여기서, C4는 사용자 디바이스 3에 의해 요청되며 어디에서도 캐싱되지 않는다).

[0092] HgC 방식은 다음과 같이(HgC 알고리즘 및 도 6과 관련하여) 작용한다. $j = n = 3$, $G_3 = \{A2, A3, B2, B3, C2\}$ 에 대해, 꼭지점 $v = A2$ 라 하면, B2 및 C2가 세트(L)일 수 있으며, 따라서 세트 $L = \{A2, B2, C2\}$ 임이 발견될 수 있다. 이제, $|L| = n = 3$ 이기 때문에, A2, B2, C2를 블랙으로 컬러링한다. 이후, $3 = G_3 \setminus L = \{A3, B3\}$ 이다. 후속하는 루프에서, $|L| = n = 3$ 을 가지는 세트(L)를 발견할 수 없기 때문에, 알고리즘의 라인 21로 이동한다. 이후, $|L| \geq n = 3$ 을 가지는 세트(L)를 발견할 수 없기 때문에, 이후, $G_2 = G_2 \cup \{A3\}$ 을 수행하고, 이후 $G_2 = G_2 \cup \{B3\}$ 을 수행한다. 따라서, $G_2 = \{A3, A4, B3, B4, C3\}$ 를 획득한다. 이제, 알고리즘의 라인 3(즉, 다음 루프 시작)으로 간다. $j = j - 1 = n - 1 = 2$ 에 대해, 먼저 꼭지점 $v = A4$ 을 선택하고, 이후 $L = \{A4, B4\}$ 을 찾는다. $\{A4, B4\}$ 를 블루로 컬러링한다(도 4 참조). 이제, $G_2 = G_2 \setminus \{A4, B4\} = \{A3, B3, C3\}$ 이다. 이후, 알고리즘의 라인 21에서, $K_{C3} = \{3, 2\}$ 를 가지는 C3인, K_v 의 가장 작은 길이를 가지는 꼭지점을 발견하고, 이후, $L = \{C3, B3\}$ 를 발견한다. $L = \{C3, B3\}$ 를 레드로 컬러링한다. 이제, $G_2 = G_2 \setminus \{C3, B3\} = \{A3\}$ 이다. $L \geq 2$ 가 존재하지 않기 때문에, 이후 $G_1 = G_1 \cup \{A3\} = \{C4, A3\}$ 를 수행한다. 이후 다음 루프 $j = j - 1 = n - 2 = 1$ 로 간다. 이후, $L = \{C4\}$ 임을 알 수 있고, $\{C4\}$ 를 퍼플로 컬러링한다. 이후, $G_1 = G_1 \setminus \{C4\} = \{A3\}$ 이다. 따라서, $L = \{A3\}$ 임을 발견할 수 있으며, $\{A3\}$ 를 브라운으로 컬러링한다. 완전히 컬러링된 충돌 그래프가 도 7에 도시된다.

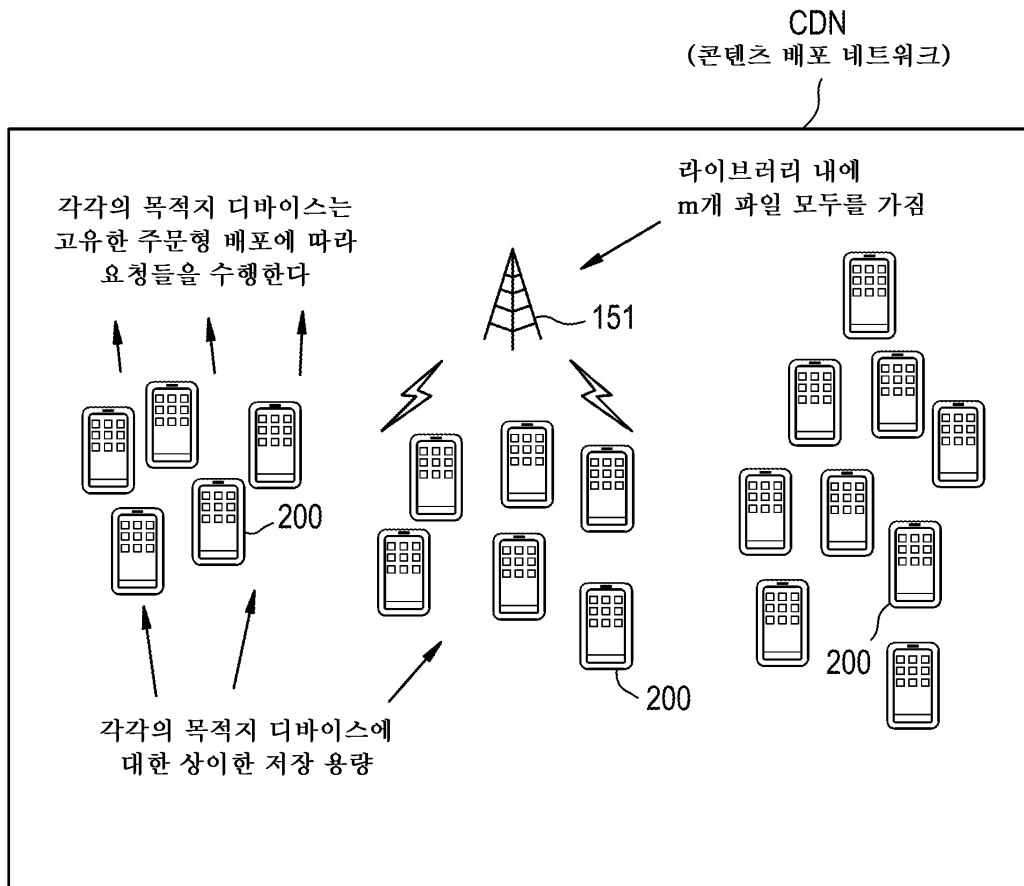
[0093] 전술된 동작들은, 예시적인 실시예들이 목적지 디바이스들(200)에서 더 인기 있는 파일들의 더 많은 패킷들을 캐싱하고, 목적지 디바이스들(200)에 의해 총체적으로 캐싱되는 각각의 파일의 별개의 패킷들의 양을 증가(또는 대안적으로, 최대화)시키고, 그리고 데이터 파일들의 요청되는 패킷들의 전체 세트 내에서의 코딩된 멀티캐스트 전송들을 허용하는 능력을 허용하기 때문에, 네트워크의 개선된 성능을 허용한다는 것이 이해되어야 한다. 또

한, 전송 이전에 요청되는 패킷들을 결합시킴으로써, 적어도 일 예시적인 실시예에 따른 전달 방법들 및/또는 디바이스들이 네트워크 엘리먼트(151)의 전송들의 횟수를 감소시킬 수 있으며, 이것이 소모를 감소시키고 네트워크 효율성을 개선할 수 있다는 것이 이해되어야 한다. 예를 들어, 관심 있는 일부 시나리오에서, 전송된 방법들 및 디바이스들은 전송 횟수를 10의 인자까지 감소시킬 수 있다. 제한되지 않은 복잡도를 가지는 경우, 이득이 제한되지 않을 수 있다. 또한, 각각의 사용자 목적지의 메모리(또는 캐시) 사이즈가 대역폭 곱셈기로서 사용될 수 있다는 것이 이해되어야 한다.

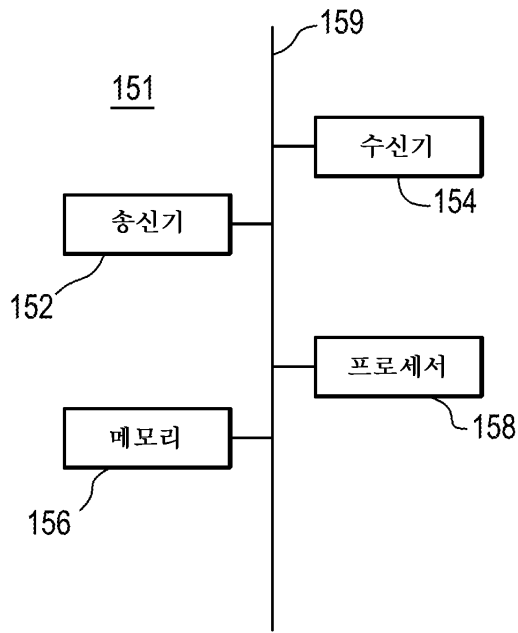
[0094] 예시적인 실시예들의 변형들은 예시적인 실시예들의 사상 및 범위로부터의 이탈로서 간주되지 않아야 한다. 모든 이러한 변형들은, 본 기술분야의 통상의 기술자에게 명백할 바와 같이, 이 개시내용의 범위 내에 포함되도록 의도된다.

도면

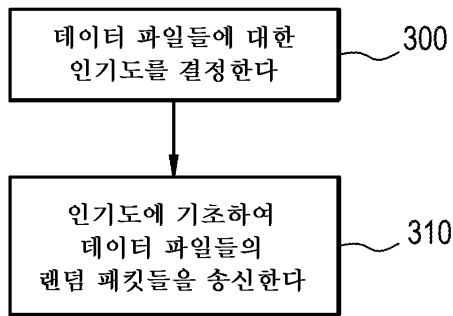
도면1



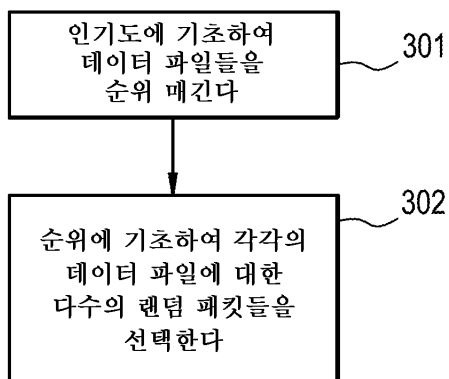
도면2



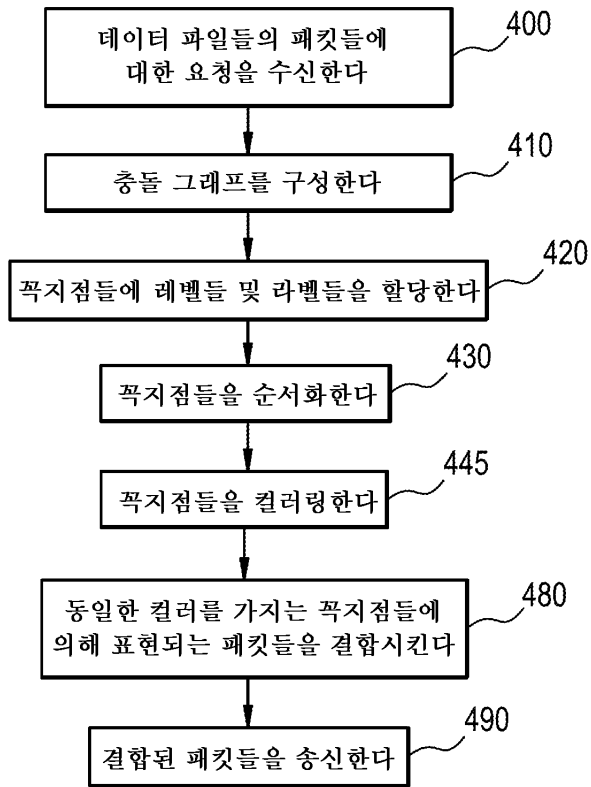
도면3a



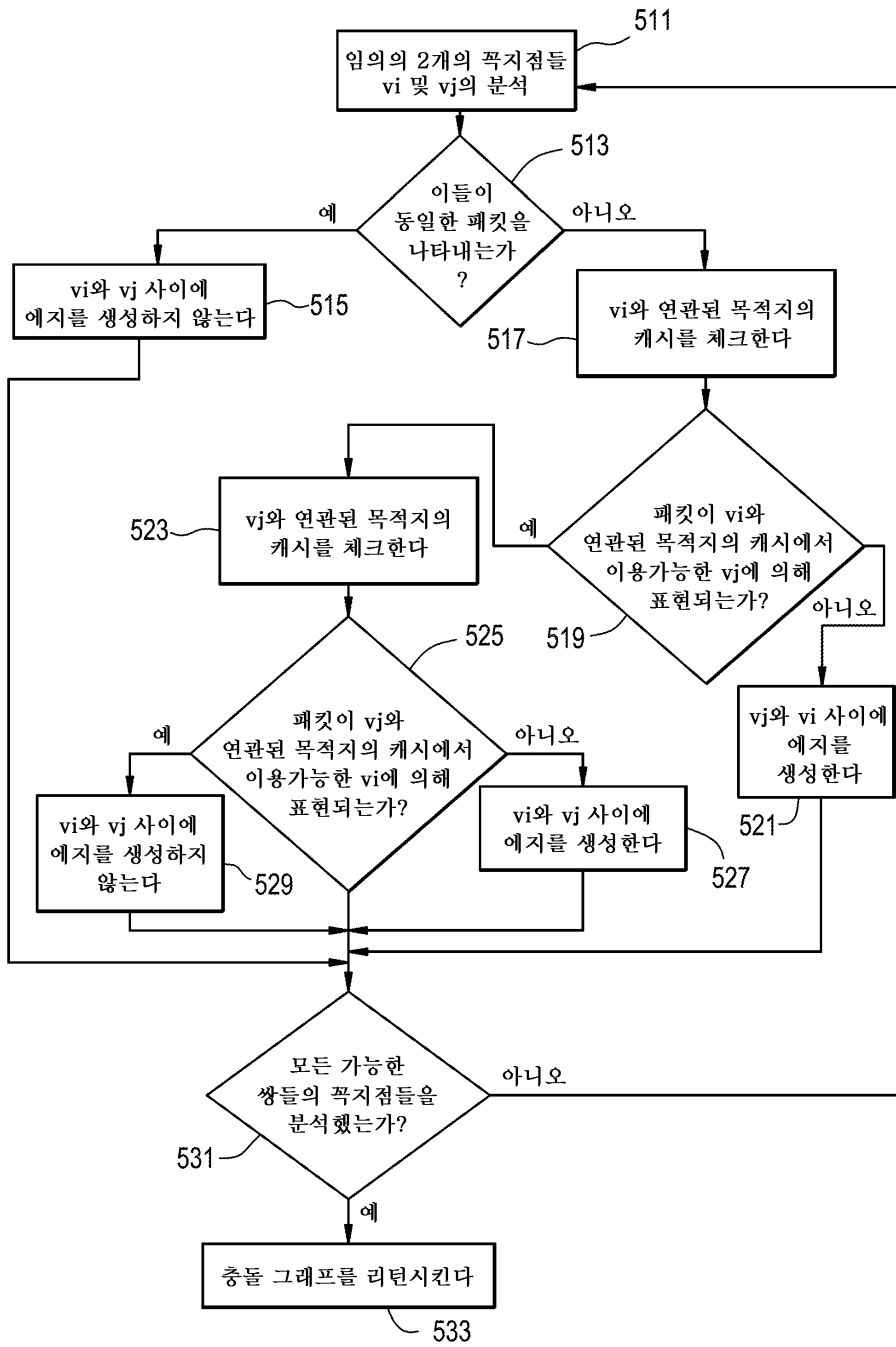
도면3b



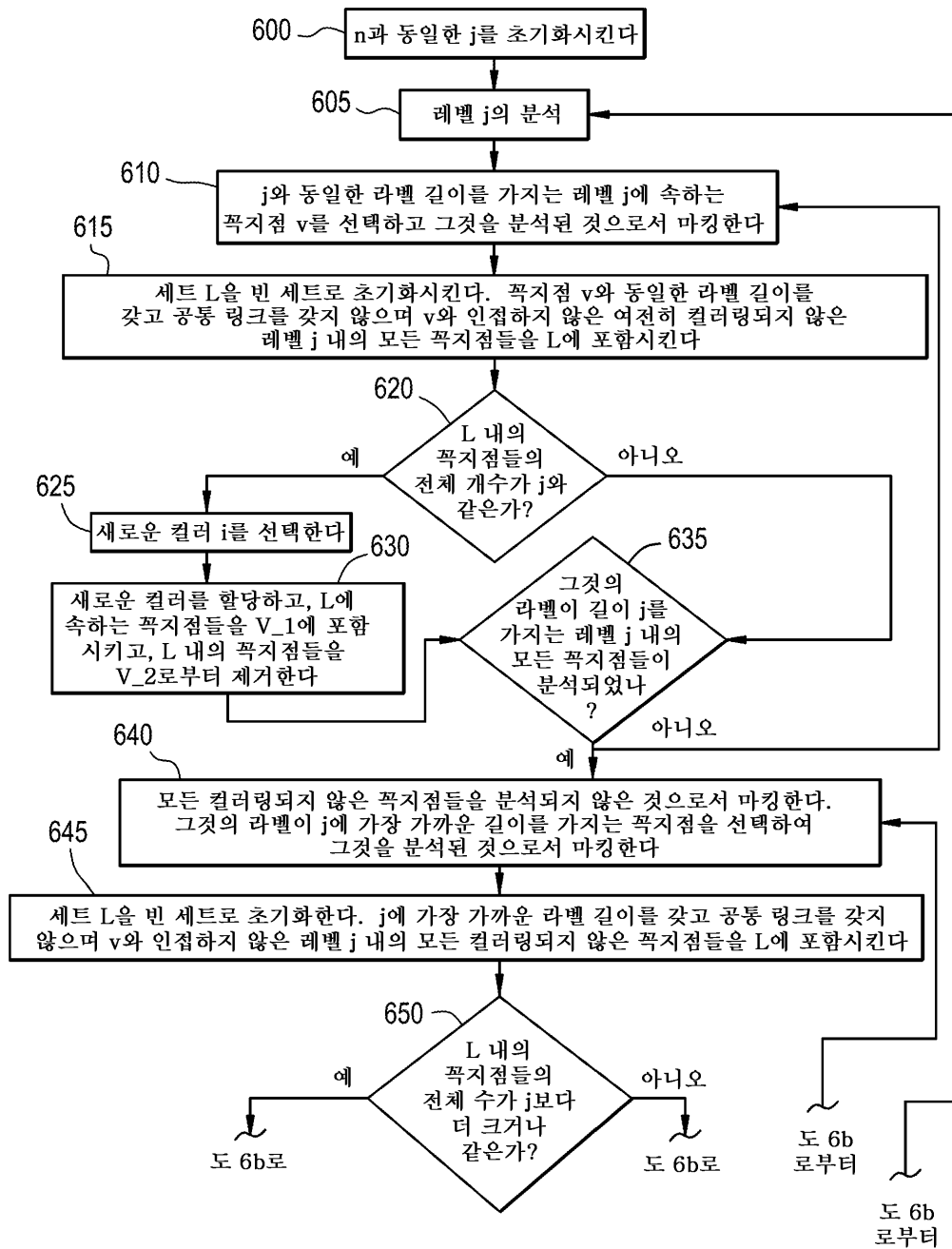
도면4



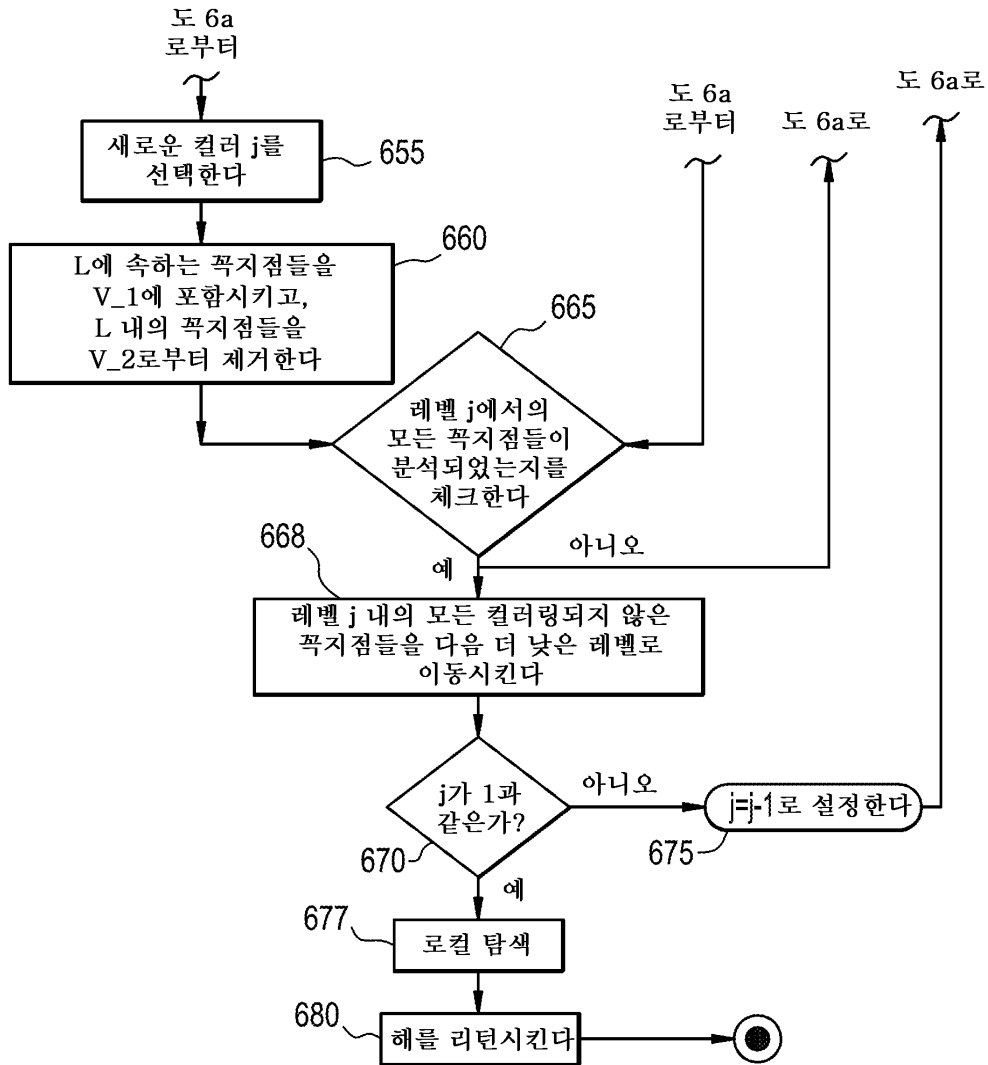
도면5



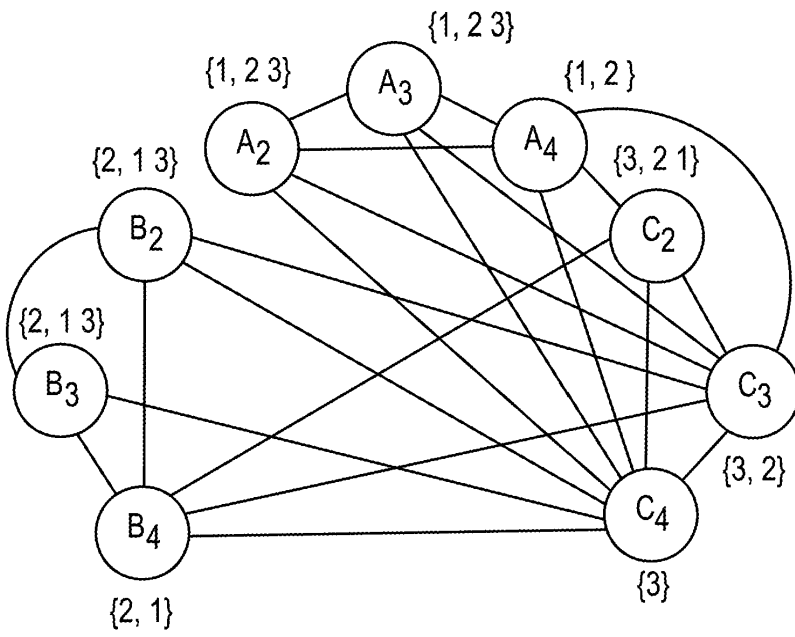
도면6a



도면6b



도면7



A₂, B₂, C₂ = 블랙

B₃, C₃ = 레드

A₄, B₄ = 블루

A₃ = 브라운

C₄ = 퍼플