



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2009-0025990
(43) 공개일자 2009년03월11일

(51) Int. Cl.

H04N 7/08 (2006.01) H04N 7/12 (2006.01)

H04N 7/173 (2006.01)

(21) 출원번호 10-2007-0091237

(22) 출원일자 2007년09월07일

심사청구일자 2007년09월07일

(71) 출원인

서강대학교산학협력단

서울 마포구 신수동 1-1 서강대학교

(72) 발명자

남윤석

서울 노원구 상계9동 주공아파트 1204-809

정문열

경기 고양시 일산서구 일산동 후곡마을 307-102

(74) 대리인

리엔목특허법인

전체 청구항 수 : 총 6 항

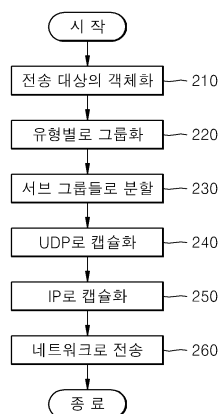
(54) IPTV 방송의 애플리케이션 전송 방법

(57) 요약

본 발명은 인터넷 프로토콜 텔레비전(일명, IPTV)에 관한 것으로서, 특히, 애플리케이션과 A/V 데이터를 포함하는 서비스를 인터넷 프로토콜 망을 통해 클라이언트에 전송하는 인터넷 프로토콜 텔레비전 방송의 애플리케이션 전송 방법에 관한 것이다.

본 발명의 일 실시 예에 따른 애플리케이션 전송 방법은, (A)전송하고자하는 애플리케이션의 계층 구조를 읽어들이고, 상기 애플리케이션을 구성하는 각각의 구성요소에 대해 고유 식별자 정보와 계층 식별자 정보를 부가하여 각각의 구성요소를 객체화시키는 단계, (B)상기 객체화된 애플리케이션 구성요소들을 그룹화하고, 상기 그룹을 UDP 패킷에 전송할 수 있는 최대 데이터 크기에 따라 분할시킨 후, 각각의 분할 그룹에 그에 상응하는 UDP 패킷을 부가하는 단계, 및 (C)상기 UDP 패킷이 부가된 각각의 분할 그룹에 IP 패킷을 더 부가하고, 이를 네트워크 통신망을 통해 클라이언트에게 전송하는 단계를 포함한다.

대 표 도 - 도2



특허청구의 범위

청구항 1

애플리케이션과 A/V 데이터를 포함하는 서비스를 인터넷 프로토콜 망을 통해 클라이언트에 전송하는 IPTV 방송의 애플리케이션 전송 방법에 있어서,

- (A)전송하고자하는 애플리케이션의 계층 구조를 읽어들이고, 상기 애플리케이션을 구성하는 각각의 구성요소에 대해 고유 식별자 정보와 계층 식별자 정보를 부가하여 각각의 구성요소를 객체화시키는 단계;
- (B)상기 객체화된 애플리케이션 구성요소들을 그룹화하고, 상기 그룹을 UDP 패킷이 전송할 수 있는 최대 데이터 크기에 따라 분할시킨 후, 각각의 분할 그룹에 그에 상응하는 UDP 패킷을 부가하는 단계; 및
- (C)상기 UDP 패킷이 부가된 각각의 분할 그룹에 IP 패킷을 더 부가하고, 이를 네트워크 통신망을 통해 클라이언트에게 전송하는 단계를 포함하는 것을 특징으로 하는 IPTV 방송의 애플리케이션 전송 방법.

청구항 2

제1항에 있어서, 상기 (A) 단계는,

- (A1)애플리케이션 실행 파일에 고유 식별자 정보를 부가하고, 상기 애플리케이션 실행 파일과 동일한 디렉터리에 속하는 파일 및 하위 디렉터리 각각에 대해 고유 식별자 정보를 부가하며, 상기 디렉터리에 속하는 파일 및 하위 디렉터리에 동일한 계층 식별자 정보를 부가하는 단계;
- (A2)상기 하위 디렉터리에 속하는 파일 각각에 고유 식별자 정보를 부가하고, 상기 하위 디렉터리의 고유 식별자 정보를 상기 하위 디렉터리에 속하는 파일 각각에 대한 계층 식별자 정보로 부가하는 단계; 및
- (A3)상기 A1 단계 및 A2 단계를 반복적으로 수행하여 상기 애플리케이션 실행 파일의 상위 및 하위에 속하는 파일 및 디렉터리 전체에 대해 객체화하는 단계를 포함하는 것을 특징으로 하는 IPTV 방송의 애플리케이션 전송 방법.

청구항 3

제1항에 있어서, 상기 (A) 단계는,

- (A1)상기 구성요소가 애플리케이션 데이터 또는 애플리케이션 코드인지 여부를 나타내는 객체 유형 정보를 각각의 구성요소에 대해 부가하는 단계;
- (A2)상기 구성요소가 디렉터리인지 여부를 나타내는 디렉터리 판별 정보를 각각의 구성 요소에 대해 부가하는 단계;
- (A3)상기 구성요소가 속하는 디렉터리의 고유 식별자 정보를 나타내는 계층 식별자 정보를 각각의 구성요소에 대해 부가하는 단계;
- (A4)상기 구성요소의 업 데이트 횟수를 나타내는 객체 버전 정보를 각각의 구성요소에 대해 부가하는 단계; 및
- (A5)상기 구성요소의 데이터 크기를 나타내는 데이터 필드 길이 정보를 각각의 구성 요소에 대해 부가하는 단계 중 적어도 어느 하나를 더 포함하는 것을 특징으로 하는 IPTV 방송의 애플리케이션 전송 방법.

청구항 4

제1항에 있어서, 상기 (B) 단계는,

상기 객체화된 애플리케이션 구성요소들을 그 유형에 따라 복수 개의 그룹들로 그룹화하고, 각각의 그룹을 UDP 패킷이 전송할 수 있는 최대 데이터 크기에 따라 분할시킨 후, 각각의 분할 그룹에 그에 상응하는 UDP 패킷을 부가하는 단계인 것을 특징으로 하는 IPTV 방송의 애플리케이션 전송 방법.

청구항 5

애플리케이션과 A/V 데이터를 포함하는 서비스를 인터넷 프로토콜 망을 통해 클라이언트에 전송하는 IPTV 방송의 애플리케이션 전송 방법에 있어서,

- (A)전송하고자하는 애플리케이션의 계층 구조를 읽어들이고, 상기 애플리케이션을 구성하는 각각의 구성 요소에 대해 공유 식별자 정보와 계층 식별자 정보를 부가하여 각각의 구성요소를 객체화시키는 단계;
- (B)상기 애플리케이션의 실행과 관련된 이벤트를 기술하는 메타데이터를 복수 개의 파라미터들로 분류하고, 각각의 파라미터에 대해 파라미터 식별자 정보를 부가하여 상기 메타데이터를 객체화시키는 단계;
- (C)상기 객체화된 애플리케이션 구성요소들 및 상기 메타데이터를 그룹화하고, 상기 그룹을 UDP 패킷이 전송할 수 있는 최대 데이터 크기에 따라 분할시킨 후, 각각의 분할 그룹에 그에 상응하는 UDP 패킷을 부가하는 단계; 및
- (D)상기 UDP 패킷이 부가된 각각의 분할 그룹에 IP(Internet Protocol) 패킷을 더 부가하고, 이를 네트워크 통신망을 통해 클라이언트에게 전송하는 단계를 포함하는 것을 특징으로 하는 IPTV 방송 애플리케이션 전송 방법.

청구항 6

제5항에 있어서, 상기 (B) 단계는,

상기 프로그램을 명칭을 나타내는 프로그램 명칭 파라미터;

상기 프로그램의 순위를 나타내는 프로그램 순위 파라미터;

상기 프로그램의 정보를 나타내는 프로그램 정보 파라미터;

상기 프로그램과 상기 애플리케이션의 동기화를 위한 상기 애플리케이션의 시작시간을 나타내는 시작시간 파라미터; 및

상기 프로그램과 상기 애플리케이션의 동기화를 위한 상기 애플리케이션의 종료시간을 나타내는 종료시간 파라미터 중 적어도 어느 하나를 포함하는 메타데이터를 객체화시키는 단계인 것을 특징으로 하는 IPTV 방송 애플리케이션 전송 방법.

명세서

발명의 상세한 설명

기술 분야

- <1> 본 발명은 인터넷 프로토콜 텔레비전(일명, IPTV)에 관한 것으로서, 특히, 애플리케이션과 A/V 데이터를 포함하는 서비스를 인터넷 프로토콜 망을 통해 클라이언트에 전송하는 인터넷 프로토콜 텔레비전 방송의 애플리케이션 전송 방법에 관한 것이다.

배경 기술

- <2> 기술의 발전에 따라 방송 시장과 통신 시장이 하나로 합쳐지면서, 다양한 서비스들이 등장하였는데, 인터넷 프로토콜 텔레비전이 그 중 하나이다. 인터넷 프로토콜 텔레비전(일명, IPTV)은 초고속 인터넷망을 이용하여 제공되는 양방향 텔레비전 서비스를 의미하는 것으로서, 좀 더 포괄적인 의미에서 다채널 방송형 서비스, 콘텐츠 요구형 서비스, 및 쌍방향 통신형 서비스가 IP(Internet Protocol) 망을 통해 동시에 가정의 TV 단말기로 제공되는 서비스 및 시스템을 말한다.
- <3> 상기 인터넷 프로토콜 텔레비전은 기존의 인터넷 네트워크 기술을 상당부분 채용하고 있기 때문에 기존의 지상파 방송이나 위성 방송에서 구현하기 까다로운 양방향 서비스를 손쉽게 제공할 수 있으며, 지상파 방송처럼 전파를 송출하거나 케이블 방송처럼 전용망을 가설할 필요없이, 기존의 인터넷 인프라인 ADSL, VDSL 등을 통한 고화질, 다채널 서비스가 가능하다는 장점이 있다.
- <4> 현재 인터넷 프로토콜 텔레비전의 추세는 ITU-T를 중심으로 국제 표준이 제정중인 상태이며, 상기 국제 표준은 기존의 디지털 방송 전송 규약인 MPEG-2를 대체하고, 인터넷 프로토콜 텔레비전 서비스가 IP 망을 통해 제공된다는 점에 비추어, 방송 스트림을 TCP 또는 UDP를 통해 캡슐화시킨 후 IP 패킷을 통해 방송 스트림을 전송하려는 방향으로 나아가는 추세이다. 특히, 인터넷 프로토콜 텔레비전 방송을 수신하는 다수의 클라이언트가 존재한다는 점에 비추어, 멀티캐스트 방식을 지원하는 UDP를 기본 전송 프로토콜로 삼고 있다.
- <5> 한편, 방송 프로그램에 연동하는 콘텐츠를 제공하기 위해서는 IP 방송 서비스에서 사용하게 될 각종 데이터와

애플리케이션 코드 또한 방송 스트림과 마찬가지로 UDP 방식을 통해 전송해야 한다. 그 이유는, 애플리케이션이 A/V 데이터와 동일한 전송 방식, 즉 UDP를 통해 전송되면 서로 간의 호환성이 좋아질 수 있고, 이로 인해 향후 연동형 애플리케이션을 제작하는 경우 훨씬 유리해질 수 있으며, UDP를 통한 데이터와 애플리케이션의 반복적인 재전송이 TCP에 비해 훨씬 용이하기 때문이다.

<6> 그러나, 인터넷 프로토콜 텔레비전 방송에서 필요한 애플리케이션 코드 및 애플리케이션 데이터를 직접 UDP로 캡슐화시키는 경우에는 하기와 같은 여러 가지 문제점이 발생한다. 첫째, UDP 특성상 100% 전송 보장이 불가능하며, 수신된 UDP 패킷의 순서가 일정하게 전달된다고 보장이 없다. 둘째, UDP는 TCP와 달리 패킷의 손실이나 지연 도착에 대한 복구 방법이 없다. 이는 IPTV 방송 환경의 특성상 애플리케이션의 반복적인 재전송 및 업 데이트가 빈번하다는 점에 비추어 애플리케이션을 직접 UDP로 캡슐화하는 데에는 문제점이 발생한다.

<7> 특히, 인터넷 프로토콜 텔레비전 방송의 애플리케이션을 종래의 디지털 방송 전송 규약인 MPEG-2 TS를 이용하여 전송하는 경우, 첫째, 애플리케이션을 여러 단계의 복잡한 과정을 거쳐서 MPEG-2 섹션으로 인코딩하여야 한다는 점, 둘째, 애플리케이션의 특정 데이터를 업 데이트하는 경우 애플리케이션 전체를 재전송하여야 한다는 점, 셋째, 특정 애플리케이션에 대한 정보가 여러 곳에 산발적으로 존재한다는 점에 비추어 비효율적이다. 따라서, 인터넷 프로토콜 텔레비전 방송의 특성을 감안하여 애플리케이션을 효율적으로 전송할 수 있는 방법이 필요하다.

발명의 내용

해결 하고자 하는 과제

<8> 본 발명이 해결하고자 하는 기술적 과제는, 애플리케이션의 계층 구조에 따라 각각의 구성 요소에 ID를 부가하고, ID가 부가된 구성 요소들을 바탕으로 그룹을 생성한 후 전송함으로써 애플리케이션 파일의 계층 구조를 손쉽게 전송가능하도록 하는 애플리케이션 전송 방법을 제공하는 데 있다.

<9> 본 발명이 해결하고자 하는 다른 기술적 과제는, 애플리케이션을 전송할 때 애플리케이션 코드와 애플리케이션 데이터의 분리 전송을 가능하게 함으로써 빈번한 업 데이트 주기를 갖는 애플리케이션을 효율적으로 전송가능하도록 하는 애플리케이션 전송 방법을 제공하는 데 있다.

<10> 본 발명이 해결하고자 하는 또 다른 기술적 과제는, 애플리케이션과 관련된 메타데이터를 애플리케이션과 함께 전송가능하도록 함으로써 특정 애플리케이션에 대한 정보가 여러 곳에 흩어져 존재하여 발생하는 불편함을 해결가능하도록 하는 애플리케이션 전송 방법을 제공하는 데 있다.

과제 해결수단

<11> 상기 기술적 과제를 해결하기 위한 본 발명의 일 실시 예에 따른 IPTV 방송의 애플리케이션 전송 방법은,

<12> 애플리케이션과 A/V 데이터를 포함하는 서비스를 인터넷 프로토콜 망을 통해 클라이언트에 전송하는 IPTV 방송의 애플리케이션 전송 방법에 있어서, (A)전송하고자하는 애플리케이션의 계층 구조를 읽어들이고, 상기 애플리케이션을 구성하는 각각의 구성요소에 대해 고유 식별자 정보와 계층 식별자 정보를 부가하여 각각의 구성요소를 객체화시키는 단계, (B)상기 객체화된 애플리케이션 구성요소들을 그룹화하고, 상기 그룹을 UDP 패킷이 전송할 수 있는 최대 데이터 크기에 따라 분할시킨 후, 각각의 분할 그룹에 그에 상응하는 UDP 패킷을 부가하는 단계, 및 (C)상기 UDP 패킷이 부가된 각각의 분할 그룹에 IP 패킷을 더 부가하고, 이를 네트워크 통신망을 통해 클라이언트에게 전송하는 단계를 포함하는 것을 특징으로 한다.

<13> 여기서, 상기 (A) 단계는, (A1)애플리케이션 실행 파일에 고유 식별자 정보를 부가하고, 상기 애플리케이션 실행 파일과 동일한 디렉터리에 속하는 파일 및 하위 디렉터리 각각에 대해 고유 식별자 정보를 부가하며, 상기 디렉터리에 속하는 파일 및 하위 디렉터리에 동일한 계층 식별자 정보를 부가하는 단계, (A2)상기 하위 디렉터리에 속하는 파일 각각에 고유 식별자 정보를 부가하고, 상기 하위 디렉터리의 고유 식별자 정보를 상기 하위 디렉터리에 속하는 파일 각각에 대한 계층 식별자 정보로 부가하는 단계, 및 (A3)상기 A1 단계 및 A2 단계를 반복적으로 수행하여 상기 애플리케이션 실행 파일의 상위 및 하위에 속하는 파일 및 디렉터리 전체에 대해 객체화하는 단계를 포함하는 것이 바람직하다.

<14> 여기서, 상기 (A) 단계는, (A1)상기 구성요소가 애플리케이션 데이터 또는 애플리케이션 코드인지 여부를 나타내는 객체 유형 정보를 각각의 구성요소에 대해 부가하는 단계, (A2)상기 구성요소가 디렉터리인지 여부를 나타내는 디렉터리 판별 정보를 각각의 구성 요소에 대해 부가하는 단계, (A3)상기 구성요소가 속하는 디렉터리의 고유 식별자 정보를 나타내는 계층 식별자 정보를 각각의 구성요소에 대해 부가하는 단계, (A4)상기 구성요소의

업 데이터릿 횡수를 나타내는 객체 버전 정보를 각각의 구성요소에 대해 부가하는 단계, 및 (A5)상기 구성요소의 데이터릿 크기를 나타내는 데이터릿 필드 길이 정보를 각각의 구성 요소에 대해 부가하는 단계 중 적어도 어느 하나를 더 포함하는 것이 바람직하다.

- <15> 여기서, 상기 (B) 단계는, 상기 객체화된 애플리케이션 구성요소들을 그 유형에 따라 복수 개의 그룹들로 그룹화하고, 각각의 그룹을 UDP 패킷이 전송할 수 있는 최대 데이터 크기에 따라 분할시킨 후, 각각의 분할 그룹에 그에 상응하는 UDP 패킷을 부가하는 단계인 것이 바람직하다.
- <16> 상기 기술적 과제를 해결하기 위한 본 발명의 다른 실시 예에 따른 IPTV 방송의 애플리케이션 전송 방법은,
- <17> 애플리케이션과 A/V 데이터를 포함하는 서비스를 인터넷 프로토콜 망을 통해 클라이언트에 전송하는 IPTV 방송의 애플리케이션 전송 방법에 있어서, (A)전송하고자하는 애플리케이션의 계층 구조를 읽어들이고, 상기 애플리케이션을 구성하는 각각의 구성 요소에 대해 공유 식별자 정보와 계층 식별자 정보를 부가하여 각각의 구성요소를 객체화시키는 단계, (B)상기 애플리케이션의 실행과 관련된 이벤트를 기술하는 메타데이터를 복수 개의 파라미터들로 분류하고, 각각의 파라미터에 대해 파라미터 식별자 정보를 부가하여 상기 메타데이터를 객체화시키는 단계, (C)상기 객체화된 애플리케이션 구성요소들 및 상기 메타데이터를 그룹화하고, 상기 그룹을 UDP 패킷이 전송할 수 있는 최대 데이터 크기에 따라 분할시킨 후, 각각의 분할 그룹에 그에 상응하는 UDP 패킷을 부가하는 단계, 및 (D)상기 UDP 패킷이 부가된 각각의 분할 그룹에 IP(Internet Protocol) 패킷을 더 부가하고, 이를 네트워크 통신망을 통해 클라이언트에게 전송하는 단계를 포함하는 것을 특징으로 한다.
- <18> 여기서, 상기 (B) 단계는, 상기 프로그램을 명칭을 나타내는 프로그램 명칭 파라미터, 상기 프로그램의 순위를 나타내는 프로그램 순위 파라미터, 상기 프로그램의 정보를 나타내는 프로그램 정보 파라미터, 상기 프로그램과 상기 애플리케이션의 동기화를 위한 상기 애플리케이션의 시작시간을 나타내는 시작시간 파라미터, 및 상기 프로그램과 상기 애플리케이션의 동기화를 위한 상기 애플리케이션의 종료시간을 나타내는 종료시간 파라미터 중 적어도 어느 하나를 포함하는 메타데이터를 객체화시키는 단계인 것이 바람직하다.

효 과

- <19> 본 발명에 따른 애플리케이션 전송 방법은, 애플리케이션을 구성하는 각각의 파일 및 디렉터리에 식별자 정보를 부가하여 전송함으로써 간단하게 데이터를 캡슐화시키는 것이 가능하고, 애플리케이션 코드와 애플리케이션 실행 코드의 분리 전송을 보장함으로써 특정 파일을 업-데이트할 경우 기존에 비해 전송과정에서의 대역폭 낭비 및 시간적 낭비를 줄일 수 있는 장점이 있다. 또한, 애플리케이션과 관련된 메타데이터를 애플리케이션과 함께 전송함으로써 애플리케이션 실행에 관한 정보를 하나의 메타데이터 객체를 통해 얻을 수 있으므로 효율적이다.

발명의 실시를 위한 구체적인 내용

- <20> 본 발명과 본 발명의 동작상의 이점 및 본 발명의 실시에 의하여 달성되는 목적을 충분히 이해하기 위해서는, 본 발명의 바람직한 실시 예를 나타내는 첨부 도면 및 도면에 기재된 내용을 참조하여야 한다. 이하, 첨부된 도면을 참조하여 본 발명의 바람직한 실시 예를 상세히 설명한다. 본 발명을 설명함에 있어, 관련된 공지 구성 또는 기능에 대한 구체적인 설명이 본 발명의 요지를 불필요하게 흐릴 수 있다고 판단되는 경우에는 그 상세한 설명을 생략한다.
- <21> 도 1은 인터넷 프로토콜 텔레비전이 수신하는 서비스를 나타내는 도면이다.
- <22> 인터넷 프로토콜 텔레비전은 종래의 디지털 방송이 제공하는 A/V 데이터뿐만 아니라 여러 가지 종류의 애플리케이션 및 프로그램 등에 관한 정보들을 제공한다. 도 1을 참조하면, 서비스 리스트에는 복수 개의 서비스들이 포함되며, 각각의 서비스에는 데이터 스트림, 애플리케이션 코드, 애플리케이션 데이터, 이벤트, A/V 데이터 등에 포함된다.
- <23> 서비스는 디지털 방송에서 제공하는 하나의 특정 채널을 지칭하며, 데이터 스트림은 다른 채널로부터 참조할 수 있는 A/V 데이터 등을 지칭하며, 애플리케이션 코드는 방송 환경에서 동작하는 프로그램 소스 코드를 지칭하며, 애플리케이션 데이터는 날씨 또는 교통 정보와 같이 애플리케이션에 사용되는 데이터를 지칭하며, 이벤트는 9시 뉴스나 10시 드라마와 같은 특정 프로그램을 지칭하며, A/V 데이터는 기존의 디지털 방송에서 전송하는 영상을 지칭한다.
- <24> 한편, 인터넷 프로토콜 텔레비전 방송의 동작 흐름을 설명하면 다음과 같다. 클라이언트는 서버로부터 전송되는 서비스 리스트에 포함된 하나의 서비스를 선택하여 참가 메시지를 전송한다. 서버는 클라이언트가 선택한 서비

스를 전송하며, 전송된 서비스에 대한 정보는 수신기(셋-톱 박스)에서 해석된다. 이 경우 수신기는 IP 패킷으로 캡슐화된 A/V 데이터를 UDP로 캡슐화되기 이전 단계로 복원하고, 애플리케이션 또한 UDP로 캡슐화되기 이전 단계로 복원한다.

- <25> 도 2는 본 발명의 일 실시 예에 따른 애플리케이션 전송 방법을 나타내는 흐름도이다.
- <26> 전송하고자 하는 애플리케이션의 계층 구조를 읽어들이고 후, 애플리케이션(애플리케이션 코드 및 애플리케이션 데이터를 포함)을 구성하는 파일 및 디렉터리 각각에 대해 고유 식별자 정보와 계층 식별자 정보를 부가하여 객체화시킨다(210). 상기 고유 식별자 정보는 각각의 객체를 다른 객체와 구별하기 위한 정보이며, 상기 계층 식별자 정보는 각각의 객체가 존재하는 위치를 구별하기 위한 정보이다. 또한, 애플리케이션과 관련된 메타데이터(특히, 프로그램 관련 정보)에 식별자 정보를 부가하여 상기 메타데이터를 객체화시킨다(210)
- <27> 각각의 객체가 애플리케이션 코드, 애플리케이션 데이터, 또는 메타데이터인지의 여부에 따라 그 유형별로 그룹화시킨다(220). 또한, 상기 220 단계 이후에 각 그룹의 특징을 나타내는 헤더를 부가하는 단계가 더 포함될 수 있다. 이와 같이 유형별로 그룹화시키는 이유는, 특정 애플리케이션 파일만을 업-데이트할 경우, 해당 파일만을 새롭게 교체하여 전송하기 위함이다. 다만, 특정 애플리케이션을 맨 처음 전송할 때에는 애플리케이션 코드와 애플리케이션 데이터를 함께 전송해야 하므로 상기 220 단계는 생략하는 것이 바람직하다.
- <28> 각각의 그룹을 UDP 패킷이 전송할 수 있는 최대 크기에 맞게 복수 개의 서브 그룹들로 분할한다(230). 또한, 상기 220 단계 이후에 각 서브 그룹의 특징을 나타내는 헤더를 부가하는 단계가 더 포함될 수 있다. 일반적으로, UDP 패킷은 64kb까지 전송할 수 있으므로 상기 최대 크기는 64kb인 것이 바람직하다. 따라서, 각 서브 그룹의 크기가 64kb 미만인 경우에는, 상기 230 단계는 생략될 수 있다.
- <29> 각각의 서브 그룹을 UDP로 캡슐화한다(240). UDP 전송방식은 다수의 시청자에게 안정적이고 빠른 송출이 가능한 장점이 있으므로, 본 발명에 따른 애플리케이션도 UDP를 이용하여 전송하는 것이 바람직하다. UDP 헤더에는 송신지 포트, 수신지 포트, 패킷의 길이, 체크-섬에 관한 정보가 포함될 수 있다. UDP 헤더가 부가된 각각의 서브 그룹을 IP로 캡슐화한다(250). 그 후 네트워크를 통하여 클라이언트에게 애플리케이션 또는 메타데이터를 전송한다(260).
- <30> 도 3은 본 발명에 따른 애플리케이션 객체화 단계에서 애플리케이션에 부가되는 데이터를 나타내는 테이블이다.
- <31> 본 발명에 따른 애플리케이션 전송 방법은 전송하고자 하는 애플리케이션(애플리케이션 코드 및 애플리케이션 데이터를 포함)의 계층 구조를 읽어들이고 후, 상기 애플리케이션을 구성하는 파일 및 디렉터리 각각에 대해 ID를 부가하여 객체화시키는 단계를 포함한다. 상기 ID에는 각각의 구성요소를 식별하기 위한 고유 식별자 정보뿐만 아니라 상기 구성요소의 상하 관계를 식별하기 위한 계층 식별자 정보를 포함된다.
- <32> 도 3을 참조하면, 애플리케이션 코드 및 애플리케이션 데이터 각각에 대해 이들을 식별할 수 있는 Object ID가 부가된다. 또한, 애플리케이션 코드 및 애플리케이션 데이터 각각에 대해 이들이 속한 디렉터리를 식별할 수 있는 Parent ID가 부가된다. 뿐만 아니라, Object ID가 부여된 구성요소에는 파일인지 또는 디렉터리인지를 구별하기 위한 Directory Flag와 업-데이트의 횟수를 나타내는 Version이 부가될 수 있다.
- <33> 이와 같은 Object ID 및 Parent ID의 배정은 파일 또는 디렉터리의 특성에 상관없이 무작위로 수행된다. 다만, 애플리케이션을 구성하는 파일 및 디렉터리의 복원을 간편하게 하기 위해 애플리케이션 실행 파일에 Object ID와 Parent ID로 각각 1을 부가하는 것이 바람직하다. 그 경우, 수신기에서는 Object ID와 Parent ID가 각각 1인 실행 파일을 중심으로 애플리케이션의 계층 구조를 파악할 수 있으므로 애플리케이션의 복원을 손쉽게 수행할 수 있는 장점이 있다.
- <34> 도 3을 다시 참조하면, AA.class에 Object ID와 Parent ID로 각각 1을 부가한 후, AA.class와 같은 디렉터리에 속한 파일들(BB.class와 TT.txt)과 디렉터리(<data>)에 각각 Object ID를 임의로 부가한다. 이 경우 동일한 디렉터리에 속한 파일 및 디렉터리는 동일한 Parent ID를 가지므로, 이들의 Parent ID도 1이 된다. 그 후, 디렉터리 <data>에 속한 파일들(Ⅱ.jpg와 XX.xml)에 각각 Object ID를 부가하고, 이들의 Parent ID를 디렉터리 <data>의 Object ID로 부가한다.
- <35> 한편, 디렉터리와 파일의 구별을 위해 각각의 객체에 Directory Flag를 부가할 수 있다. Directory Flag가 1인 경우에는 해당 객체가 디렉터리임을 나타내고, 0인 경우에는 해당 객체가 파일임을 나타낸다. 한편, 업-데이트 횟수를 나타내기 위해 각각의 객체에 Version을 부가할 수 있다. Version이 1인 경우에는 해당 객체가 처음으로 전송됨을 나타내고, 그 이후부터는 재전송됨을 나타낸다. 이와 같이 업-데이트 횟수를 나타낼 수 있는 이유는

객체의 분리전송이 가능하기 때문이다.

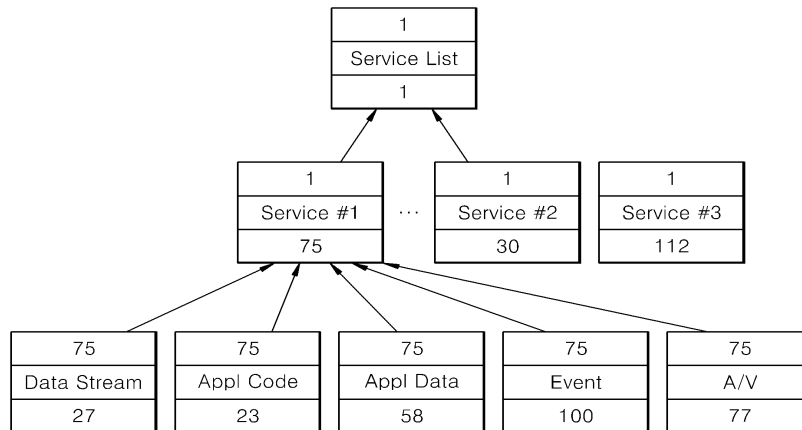
- <36> 도 4는 본 발명에 따른 메타데이터 객체화 단계에서 메타데이터를 이루는 파라미터를 나타내는 테이블이다.
- <37> 애플리케이션과 관련된 메타데이터는 연동 프로그램에 관한 정보, 동기화를 위한 시간 정보 등을 포함할 수 있다. 특히, 상술한 이벤트는 이러한 애플리케이션과 연동하는 프로그램에 관한 정보를 나타낸다. 따라서, 이러한 프로그램 관련 메타데이터를 애플리케이션과 함께 또는 별도로 전송하기 위해서는 메타데이터를 그룹화시켜 하나의 객체로 형성할 필요가 있다. 도 3은 애플리케이션과 관련된 프로그램 메타데이터를 나타낸다.
- <38> 도 4를 참조하면, 애플리케이션과 관련된 메타데이터에는 Object Update, Event Name, Event Priority, Trigger Time, Expire Time, 및 Event Info 등과 같은 복수 개의 파라미터들이 포함될 수 있다. Object Update는 메타데이터의 업-데이트 정보를 나타내며, Event Name는 애플리케이션과 연동하는 이벤트의 이름을 나타내며, Event Priority는 연동하는 이벤트와 다른 이벤트들과의 우선 순위를 나타내며, Event Info는 이벤트 자체에 대한 기본 정보를 나타낸다.
- <39> 한편, Trigger Time은 이벤트와 연동하는 애플리케이션이 특정 시간에 화면에 나타나는 시작시간을 나타내며, Expire Time은 이벤트와 연동하는 애플리케이션이 특정 시간에 화면에 사라지는 종료시간을 나타낸다. 그 밖에 애플리케이션 제작자 의도에 따라 새로운 메타데이터를 전송하고 싶은 경우, User define에 파라미터를 직접 지정하여 사용할 수 있다. 각각의 파라미터의 길이는 가변적이며, 각각의 파라미터를 다른 파라미터와 구별하기 위한 파라미터 ID가 부가된다. 또한, 파라미터의 수와 파라미터 전체 데이터 필드 길이를 나타내는 정보가 부가될 수 있다.
- <40> 도 5는 본 발명에 따른 객체화 단계에서 각각의 객체에 부가되는 데이터를 나타내는 테이블이다.
- <41> 도 5를 참조하면, 애플리케이션을 구성하는 각각의 구성요소 또는 애플리케이션과 관련된 메타데이터가 객체화되는 과정에서 각각의 객체에 부가되는 정보의 체계가 도시되어 있다. Object Type은 객체화 대상이 애플리케이션 데이터, 애플리케이션 코드, 또는 이벤트 메타데이터인지를 나타낸다. 따라서 각각의 객체는 객체 유형을 구분할 수 있는 데이터가 부가된 상태에서 전송되며, 이로 인해 객체 유형별로 분리 전송이 가능해진다.
- <42> 애플리케이션의 데이터 필드는 파일 또는 디렉터리이며, 이들 각각에 Object Type, Directory Fla, Object ID, Parent ID, Object Vision, Data field Length에 관한 정보가 각각의 파일 또는 디렉터리에 부가된다. Directory Flag는 해당 객체가 디렉터리인지 여부를 나타내며, Object ID는 파일 또는 디렉터리에 부가된 고유 식별자 정보를 나타내며, Parent ID는 상위 디렉터리에 부가된 계층 식별자 정보를 나타내며, Object Version은 해당 객체의 전송 횟수를 나타내며, Data field Length는 실제 전송하는 데이터의 길이를 나타낸다.
- <43> 애플리케이션과 관련된 이벤트 메타데이터의 데이터 필드는 복수 개의 파라미터들로 이루어진 정보이며, Object Type, Parameter Number, Data field Length에 관한 정보가 이벤트 메타데이터에 부가된다. Parameter Number는 해당 객체의 데이터 필드에 포함된 파라미터의 수를 나타낸다. 한편, 애플리케이션은 데이터필드 내에 파일 또는 디렉터리를 포함하지만, 이벤트 메타데이터는 데이터 필드 내에 파라미터들로 이루어진 애플리케이션과 프로그램의 연결 정보를 포함한다.
- <44> 도 6은 본 발명에 따른 그룹화 단계에서 각각의 그룹에 부가되는 데이터를 나타내는 테이블이다.
- <45> 본 발명에 따른 애플리케이션 전송 방법은 각각의 애플리케이션 파일 또는 디렉터리에 대해 그 구조 및 형태를 파악할 수 있는 식별자 정보를 부가한 후 하나의 그룹 또는 복수 개의 그룹으로 그룹화하는 단계를 포함한다. 본 발명에 따른 애플리케이션 전송 방법은 애플리케이션과 관련된 메타데이터에 대해 상기 메타데이터를 이루는 복수 개의 파라미터들 각각에 대해 파라미터 식별자 정보를 부가한 후, 하나의 그룹으로 그룹화하는 단계를 포함한다. 또한, 상기 그룹화 단계에는 각각의 그룹의 특성을 파악할 수 있는 헤더 정보를 부가하는 단계를 더 포함한다.
- <46> 도 6을 참조하면, 상기 헤더 정보에는 ID, Type, Version Number, Object Number, Body Size가 포함된다. ID는 각 그룹을 다른 그룹과 식별하기 위한 ID를 나타내며, Type은 해당 그룹에 포함된 데이터의 종류, 즉, 애플리케이션 데이터, 애플리케이션 코드, 또는 이벤트 메타데이터인지를 나타내며, Version Number는 특정 애플리케이션 데이터 전체를 업-데이트할 때 업-데이트 횟수를 나타내며, Object Number는 그룹에 포함된 총 객체의 수를 나타내며, Body Size는 그룹에 포함된 데이터의 크기를 나타낸다.
- <47> 도 7은 본 발명에 따른 그룹화 단계에서 그룹 유형에 따라 그룹에 부가되는 데이터를 나타내는 도면이다.

- <48> 상술한 바와 같이, 각각의 그룹에는 ID, Type, Version Number, Object Number, Body Size와 같은 헤더 정보가 부가된다. 다만, 각 그룹에 포함된 정보가 애플리케이션 코드, 애플리케이션 데이터, 또는 메타데이터인지 여부에 따라 상기 ID의 형태가 달라질 수 있다. 여기서 메타데이터는 애플리케이션과 관련된 프로그램(이벤트) 정보를 나타낸 것으로 애플리케이션 이벤트로 표시하기로 한다. 도 7을 참조하면, 각각 다른 ID 형태를 갖는 세 개의 헤더가 도시되어 있다.
- <49> 즉, 애플리케이션 코드 헤더의 경우 Appl ID와 Appl Type, 애플리케이션 데이터는 Parent Appl ID와 Data Type, 애플리케이션 이벤트는 Parent Appl ID와 Event Type을 포함하는 헤더를 가진다. 이와 같이 하나의 애플리케이션을 분리하여 전송하되, 애플리케이션 코드를 전송하는 그룹에 부여한 Appl ID를 해당 애플리케이션에서 사용되거나 관련된 정보(애플리케이션 데이터 및 애플리케이션 이벤트)를 전송하는 그룹의 Parent ID로 부여하면 수신기에서 복원이 편리하다.
- <50> 도 8은 본 발명에 따른 그룹 분할 단계에서 각각의 서브 그룹에 부가되는 데이터를 나타내는 테이블이다.
- <51> 상술한 바와 같이, 하나의 그룹은 UDP 패킷의 전송 기준을 충족시키기 위해서 복수 개의 서브 그룹들로 분할된다. 이때, 각각의 서브 그룹은 그 크기가 다른 적어도 하나 이상의 객체들을 포함하기 때문에, 각 서브 그룹의 길이는 가변적이다. 즉, 각각의 객체는 전송시에 분리되지 않아야 한다. 각각의 서브 그룹의 크기는 9byte인 것이 바람직하며, 각각의 서브 그룹에는 도 7에 도시된 헤더가 부가될 수 있다. 또한, 각 서브 그룹의 크기는 64kb를 넘지 않아야 한다.
- <52> 도 8을 참조하면, 헤더 정보에는 Sync Byte, Packet ID, Total Packets, Current Packet Number, Payload Size를 포함할 수 있다. Sync Byte는 하나의 그룹을 전송하기 위해 필요한 정보를 나타내며, Packet ID는 현재 서브 그룹의 식별자 정보를 나타내며, Total Packets는 하나의 그룹에서 나누어진 총 서브 그룹의 수를 나타내며, Current Packet Number는 현재 서브 그룹의 순위 번호를 나타내며, Payload Size는 현재 서브 그룹에 포함된 객체들의 길이를 나타낸다.
- <53> 도 9는 본 발명에 따른 애플리케이션 분리 전송 과정을 나타내는 도면이다.
- <54> 도 9를 참조하면, 애플리케이션 제작 단계, 애플리케이션 객체화 단계, 애플리케이션 전송 단계, 애플리케이션 복원 단계, 및 애플리케이션 실행 단계가 예시적으로 나타나있다. 상술한 바와 같이 애플리케이션은 애플리케이션 코드와 애플리케이션 데이터를 포함하고, 애플리케이션은 제작업체에 의해 제작된 후, 인터넷 프로토콜 텔레비전 방송 서비스를 제공할 수 있는 방송국에 의해 송출된다. 송출된 애플리케이션은 네트워크를 통해 클라이언트의 단말기로 전송된다.
- <55> 각각의 애플리케이션에는 고유 식별자 정보와 계층 식별자 정보가 부가되어 객체화된 후, 객체 유형별로 그룹화된다. 그 후, 각각의 그룹에는 그룹 유형에 따른 헤더 정보가 부가된다. 또한, 애플리케이션 메타데이터를 의미하는 애플리케이션 이벤트도 파라미터 식별자 정보가 부가되어 객체화되고, 그 유형에 따른 헤더 정보가 부가된다. 각각의 그룹은 네트워크를 통해 분리전송되며, 수신기에는 상기 고유 식별자 정보와 계층 식별자 정보를 이용하여 애플리케이션을 복원하고, 상기 메타데이터를 이용하여 애플리케이션을 프로그램과 연동하여 실행시킨다.
- <56> 이상 도면과 명세서에서 최적 실시 예가 개시되었다. 여기서 특정한 용어들이 사용되었으나, 이는 단지 본 발명을 설명하기 위한 목적으로 사용된 것이지 의미 한정이나 특허 청구범위에 기재된 본 발명의 범위를 제한하기 위하여 사용된 것은 아니다. 그러므로 본 기술 분야에서 통상의 지식을 가지는 자라면 이로부터 다양한 변형 및 균등한 타 실시 예가 가능하다는 점을 이해할 것이다. 따라서 본 발명의 진정한 기술적 보호 범위는 첨부된 특허 청구범위의 기술적 사상에 의해 정해져야 할 것이다.
- 도면의 간단한 설명**
- <57> 도 1은 인터넷 프로토콜 텔레비전이 수신하는 서비스를 나타내는 도면이다.
- <58> 도 2는 본 발명의 일 실시 예에 따른 애플리케이션 전송 방법을 나타내는 흐름도이다.
- <59> 도 3은 본 발명에 따른 애플리케이션 객체화 단계에서 애플리케이션에 부가되는 데이터를 나타내는 테이블이다.
- <60> 도 4는 본 발명에 따른 메타데이터 객체화 단계에서 메타데이터를 이루는 파라미터를 나타내는 테이블이다.

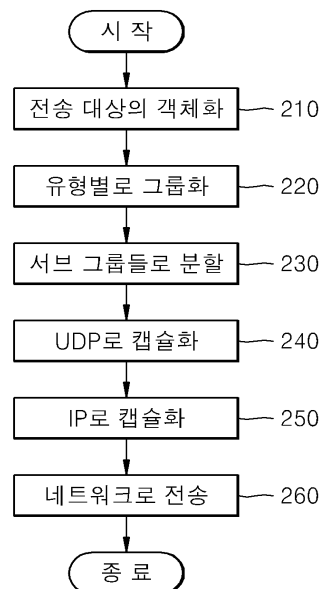
- <61> 도 5는 본 발명에 따른 객체화 단계에서 각각의 객체에 부가되는 데이터를 나타내는 테이블이다.
- <62> 도 6은 본 발명에 따른 그룹화 단계에서 각각의 그룹에 부가되는 데이터를 나타내는 테이블이다.
- <63> 도 7은 본 발명에 따른 그룹화 단계에서 그룹 유형에 따라 그룹에 부가되는 데이터를 나타내는 도면이다.
- <64> 도 8은 본 발명에 따른 그룹 분할 단계에서 각각의 서브 그룹에 부가되는 데이터를 나타내는 테이블이다.
- <65> 도 9는 본 발명에 따른 애플리케이션 분리 전송 과정을 나타내는 도면이다.

도면

도면1



도면2



도면3

	Object ID	Directory Flag	Parent ID	Version
AA.Class	1	False	1	1
BB.Class	30	False	1	1
TT.txt	112	False	1	1
<data> ll.jpg	23	False	75	1
<data> XX.xml	58	False	75	1
<data>	75	True	1	1

도면4

Name	ID	Comment
Object Update	0	업데이트 정보
Name	1	Event 이름
Priority	2	Event 순위
Trigger Time	3	동기화를 위한 시작 시간
Expire Time	4	동기화를 위한 종료 시간
Event Info	5	Event 정보
User define	6-127	For future use

도면5

Syntax	bits	Type	Comment
Object Type	3	uimsbf	Appl Data, Appl Code, Event
Directory flag	1	uimsbf	Object가 Directory 인지를 판별
Object ID	7	uimsbf	Object identifier
Parent ID	7	uimsbf	상위 Object ID를 지칭
Object Version	4	uimsbf	현재 Object의 Version
Reserved bit	3	uimsbf	For future use
Parameter Number	8	uimsbf	현재 Object의 파라미터 개수
Data field Length	23	uimsbf	Object data field 길이

도면6

Syntax	bits	Type	Comment
ID	12	uimsbf	Group Identifier
Type	3	uimsbf	Body type
Version Number	3	uimsbf	Structure version
Object Number	8	uimsbf	포함된 Object의 개수
Body Size	30	uimsbf	Body field 길이

도면7

Application code header

Appl ID	Appl Type	Version Number	Object Number	Body Size
12bit	3bit	3bit	8bit	30bit

Application data header

Parent Appl ID	Data Type	Version Number	Object Number	Body Size
12bit	3bit	3bit	8bit	30bit

Application event header

Parent Appl ID	Event Type	Version Number	Object Number	Body Size
12bit	3bit	3bit	8bit	30bit

도면8

Syntax	bits	Type	Comment
Sync Byte	8	uimsbf	Packet sync byte
Packet ID	16	uimsbf	Packet Identifier
Total Packets	16	uimsbf	전체 패킷의 수
Current Packet Number	16	uimsbf	현재 패킷의 번호
Payload Size	16	uimsbf	해당 패킷의 Payload 길이

도면9

