(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
11 September 2009 (11.09.2009)

PCT

(10) International Publication Number
WO 2009/109232 A1

(51) International Patent Classification:
*H04N 7/173* (2006.01) *H04L 29/06* (2006.01)

(21) International Application Number:
PCT/EP2008/052776

(22) International Filing Date:
7 March 2008 (07.03.2008)

(25) Filing Language: English

(26) Publication Language: English

(71) Applicant *(for all designated States except US)*: **TELEFONAKTIEBOLAGET LM ERICSSON (PUBL)** [SE/SE]; S-164 83 Stockholm (SE).

(72) Inventors; and
(75) Inventors/Applicants *(for US only)*: **LJUNGGREN, Andreas** [SE/SE]; Ångermannagatan 119, S-SE-162 64 Vällingby (SE). **SKOG, Robert** [SE/SE]; Gullvivegränd 7, S-SE-165 76 Hässelby (SE).

(74) Agent: **TALBOT-PONSONBY, Daniel**; Marks & Clerk LLP, 4220 Nash Court, Oxford Business Park South, Oxford Oxfordshire OX4 2RU (GB).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, SV, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MT, NL, NO, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

**Declarations under Rule 4.17**:

— *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))*

— *of inventorship (Rule 4.17(iv))*

**Published**:

— *with international search report (Art. 21(3))*

**(54) Title**: METHOD AND APPARATUS FOR DISTRIBUTING MEDIA OVER A COMMUNICATIONS NETWORK

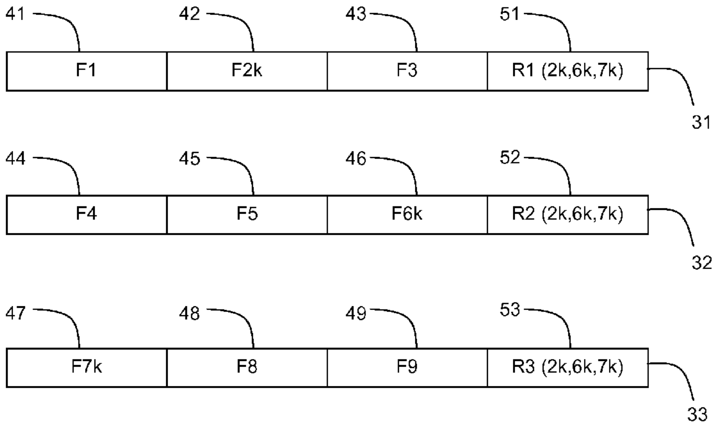



Figure 8

**(57) Abstract**: Method and Apparatus for Distributing Media over a Communications Network A method and apparatus for receiving media content in a communications network. A node in the network receives a sequence of fragments from a media stream. The node also receives fragment data relating to each fragment, the fragment data comprising any of an indicator that the fragment comprises key data relating to the media stream and an indication of when a next fragment comprising key data appears in the sequence. If the node determines that any fragments in the sequence are missing, it uses the fragment data to prioritize which missing fragments are required, andsends a request to a further node requesting the missing fragment having the highest priority.

WO 2009/109232 A1

# Method and Apparatus for Distributing Media over a Communications Network

## TECHNICAL FIELD

The invention relates to the field of distributing media over a communications network, and in particular to the field of distribution of IPTV using a communications network.

## BACKGROUND

TV services broadcast over an IP network are referred to as IPTV. IPTV is typically broadcast using a broadband access network, in which channels are transmitted over a broadband network from a super head-end down to an end-user's set top box (STB).

Linear content delivery, in which all channels in a subscription are simultaneously delivered to a user's set top box (STB), is not suitable for IPTV, as IPTV has limited bandwidth available over a broadband connection. A typical ADSL broadband connection provides a capacity of between 3 and 8 Mbps, and ADSL2 promises to deliver up to 25 Mbps downstream, whereas VDSL can provide a capacity of greater than 30 Mbps. Standard quality MPEG 2 IPTV content requires 2 Mbps per channel, and HDTV will require around 8-10 Mbps per channel. The MPEG 4 standard will approximately halve the bandwidth required to deliver IPTV content with the same quality. Nevertheless, the available bandwidth is a scarce resource, and IPTV solutions must limit the number of channels that can be delivered simultaneously.

Figure 1 illustrates a known way of distributing media in which an IPTV media stream originates in a service provider network 1, is passed to a core network 2, is further passed into a metro network 3, and finally is sent via access networks 4 to each home network 5 that contains an STB that wishes to receive the media stream. Networks can quickly become saturated due to heavy traffic loads. In order to mitigate this problem, content can be multicast to reduce bandwidth demands for broadcast TV distribution. Furthermore, Video on Demand (VoD) services can be handled by VoD cache servers located close to the end-user. However, such caches require additional investment, and many routers would need to be replaced, as existing routers may not support IPTV multicasts.

It is known to distribute an IPTV service using a Peer to Peer (P2P) network, as illustrated in Figure 2. Each STB is a peer in the network. An IPTV media stream can be delivered to a STB from another STB, from a media injector from which the stream originates, or from any other peer in the network.

An IPTV media stream is typically compressed in order to save bandwidth. An example of a compressed media format is MPEG. MPEG media streams contain different frames, such as I-frames, P-frames and B-frames. I-frames do not depend on data contained in the preceding or following frames, as they contain a complete picture. P-frames provide more compression than I-frames because they utilize data contained in the previous I-frame or P-frame. When generating a P-frame, the preceding frame is reconstructed and altered according to incremental extrapolation information. B-frame are similar to P-frames, except that B-frames interpolate data contained in the following frame as well as the preceding frame. As a result, B-frames usually provide more compression than P-frames. Typically, every 15th frame or so is an I-frame. P-frames and B-frames might follow an I-frame as follows: IBBPBBPBBPBB(I). The order and number of frames in the sequence can be varied.

Since B and P frames depend on adjacent frames it is necessary that when the STB receives a new channel, it receives a full I-frame before the new channel can be shown. The average time for switching between channels therefore depends on the length of time between I-frames. Typically, for MPEG-2 IPTV content, the length of time is around 0.5 seconds. For MPEG-4 part 10 IPTV content, the length of time between I-frames can be several seconds.

The media stream includes payload data and metadata. The payload data is the media data itself, and is decoded and shown by the receiver. Payload data typically comprises frames as described above. The metadata includes all other data in the media stream. This may be, for example, data describing the payload data, or information establishing signalling between two peers. In order to facilitate handling of the media stream, the media stream is sent in “fragments”. Fragments are discrete portions of the media stream containing both the payload data and the metadata.

In order to show a channel quickly when a STB connects to a peer, it would be useful if the peer could send the latest I-frame and all subsequent frames in order to allow the

STB to decode the media stream as quickly as possibly. Unfortunately this is not practical, as the media stream is sent in fragments. A buffer containing fragments is illustrated in Figure 3. A fragment may contain both metadata about the media stream, and payload data from the media stream itself.

The P2P network interface (in, for example, a STB) requests fragments from other P2P peers. In Figure 3 the P2P logic is writing fragment number 21 into the buffer and fragment number 17 is sent to the video decoder.

Other fragments are requested from other peers in the P2P network. However, there are situations when one peer does not have the requested fragment, or simply does not respond to the request. In this case, the requesting peer must find the fragment from another peer. Since the data is distributed over a multi-origin network, it is likely that some fragments will be lost due to network faults, or peers quickly joining and leaving the network, in which case it will not be possible to obtain the missing fragment from anywhere. In the worst case scenario, fragments containing parts of key frames (which can be seen as “key fragments”) may be permanently lost. If this happens, many peers will waste time and effort attempting to find such fragments, with no chance of success.

## SUMMARY

The inventors have realised the problems associated with the prior art and devised an apparatus and method to mitigate the problems described above.

In accordance with a first aspect of the present invention there is provided a method of distributing media content in a communications network. A node in the network, which may be a media injector, transmits a sequence of media stream fragments and recovery record fragments. The recovery record fragments contain sufficient information to enable missing media stream fragments to be reconstructed from the media stream fragments adjacent to the missing fragments.

In accordance with a second aspect of the present invention there is provided a method of receiving media content in a communications network. A node in the network receives a sequence of media stream fragments, together with recovery record fragments. The node determines if any media stream fragments in the sequence are missing and, if so, sends a request to a further node requesting the missing media stream fragments. If the missing media stream fragments cannot be recovered, they are reconstructed from adjacent media stream fragments and the recovery record fragments using a suitable reconstruction method such as, for example, Reed Solomon error correction.

Optionally, the method further comprises storing the media stream fragments and recovery record fragments in a memory. This is in order to preserve the fragments in a buffer for use by a video decoder when the fragment is required. However, if a fragment is not useable or required, then there is no need to store it.

Optionally, each media fragment may include fragment data relating to each media stream fragment. Such fragment data may comprise an indicator of whether that media stream fragment is a key fragment comprising key data relating to the media stream. The reconstruction of missing fragments may be prioritised on the basis of the fragment data. For example, it may be that only missing key fragments are reconstructed, in which case it is likely that the reconstruction will be on the basis of adjacent key fragments together with the recovery records.

The fragment data is optionally included in metadata contained within the fragment. This is advantageous because both the fragment and the fragment data can be efficiently sent together. Optionally, the fragment data is added to each fragment at a media injector.

As an alternative option, the fragment data is received separately from each fragment. This is advantageous in the case where a node communicates with a further node, and can send the fragment data separately from the fragment itself, without having to re-write the fragment.

Key data optionally comprises complete image data for a single image in the media stream. In the optional case where the media stream is an MPEG media stream, the key data may comprise an I-frame. However, key data can be any data that is crucial to the media stream. Further examples of key data are Icecast with ID3 tagging, Ogg-tagging, x.264 B frames, header information, Closed Caption data, and encryption data.

In some systems, media stream fragments are transmitted in Maximum Transmission Unit packets. If this is the case the recovery record fragments may optionally use up space within each Maximum Transmission Unit packet not used by the media stream fragments. Alternatively, the recovery record fragments may utilise a fixed proportion of bandwidth.

The node is optionally a Set Top Box. In an optional embodiment of the invention, the communications network is a peer to peer communications network, and each node is a peer node.

In accordance with a third aspect of the present invention there is provided a media injector node for distributing media content in a communications network. The node comprises a receiver for receiving a media stream. The node also includes a processor for partitioning the media stream into a sequence of media stream fragments and generating recovery record fragments, the recovery record fragments providing enough information to enable the reconstruction of a missing media stream fragment from adjacent media stream fragments. A transmitter is provided for sending the media stream fragments and recovery record fragments to at least one node located in the communications network.

In accordance with a fourth aspect of the present invention there is provided a node for use in an communications network. A receiver receives a sequence of media stream fragments, and recovery record fragments. A processor determines if any fragments in the sequence are missing. A transmitter transmits a request to a further node for the missing fragments. The processor is also arranged to reconstruct the missing fragments using the recovery record fragments and adjacent media stream fragments.

According to a fifth aspect of the invention, there is provided apparatus for use in distributing media over a communications network, the apparatus comprising means for performing the method as described in the first or second aspect of the invention.

According to a sixth aspect of the invention, there is provided a program for controlling an apparatus to perform the method as described in the first or second aspect of the invention.

According to a seventh aspect of the invention, there is provided a program which, when loaded into an apparatus, causes the apparatus to become an apparatus according to the sixth aspect of the invention.

According to an eighth aspect of the invention, there is provided a program according to the sixth or seventh aspects of the invention, carried on a carrier medium. The carrier medium is optionally a storage medium.

According to a ninth aspect of the invention, there is provided a storage medium containing a program according to any one of the sixth or seventh aspects of the invention.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 illustrates schematically in a block diagram an architecture for the distribution of IPTV;

Figure 2 illustrates schematically in a block diagram an architecture for the distribution of IPTV in a peer to peer network;

Figure 3 illustrates schematically in a block diagram a buffer in a STB containing data fragments;

Figure 4 illustrates schematically in a block diagram a media injector and two Set Top Boxes;

Figure 5 illustrates schematically in a block diagram the signalling required to initiate an IPTV broadcast with a first Set Top Box;

Figure 6 illustrates schematically in a block diagram the signalling required to initiate an IPTV broadcast with a further Set Top Box;

Figure 7 illustrates schematically in a block diagram keep alive messages sent by a Set Top Box;

Figure 8 illustrates schematically the insertion of recovery records into the network;

Figure 9 is a flow diagram illustrating the steps according to an embodiment of the invention;

Figure 10 illustrates schematically in a block diagram a media injector according to an embodiment of the invention; and

Figure 11 illustrates schematically in a block diagram a peer node according to an embodiment of the invention.

## DETAILED DESCRIPTION

The following description sets forth specific details, such as particular embodiments, procedures, techniques, etc. for purposes of explanation and not limitation. In some instances, detailed descriptions of well known methods, interfaces, circuits, and devices are omitted so as not obscure the description with unnecessary detail. Moreover, individual blocks are shown in some of the drawings. It will be appreciated that the functions of those blocks may be implemented using individual hardware circuits, using software programs and data, in conjunction with a suitably programmed digital microprocessor or general purpose computer, using application specific integrated circuitry, and/or using one or more digital signal processors.

Throughout this specification, reference is made to fragments containing key data. An example of key data is the I-frames in the MPEG format. However, it will be appreciated by persons of skill in the art that the invention applies to any key data for the media stream. Examples of key data include any of:

Icecast with ID3 tagging

Ogg-tagging

MPEG I-frames

Possible B frames in x.264

Making sure that the header survives in MJPEG

Header of RTP (if RTP runs on top of P2P)
Closed Caption subtitles
Encryption information

The IPTV P2P requires a media injector in order to introduce the IPTV media stream into the network, although the media injector is not a true peer in the network in the sense that it sends media data but does not receive media data from the peers. This is illustrated in Figure 4, which is a schematic representation of a simple IPTV P2P network. The network includes an IPTV server 6 and two STBs STB1 and STB2. Each STB includes a P2P network interface to which is connected a video decoder 9, 11. In this example, STB2 receives the IPTV media stream from both STB1 and the IPTV Server 6, which injects either streaming content 4 or content from a database 7 using a P2P media injector 8. Note that other network nodes (in addition to STBs) may be peers in the network.

Figure 5 illustrates typical signalling required to initiate an IPTV broadcast with a first STB STB1. The video decoder 9 in STB1 receives an instruction from a user to start channel X. This is relayed to the P2P network interface 2 in STB1, which sends a request to a STB manager 10 in the IPTV back-end to join channel X. The STB Manager 10 returns a peer list to the P2P function in STB1, but no IPTV media stream. The peer list includes the P2P media injector 8. Since the media injector can be considered as a peer in the network, it is termed STB0. The P2P function in STB1 then sends a request to join channel X to STB0. STB0 receives an IPTV media stream from an IPTV media stream source (for example, from the database 7), and sends a peer list and an IPTV media stream comprising fragments of frames to the P2P network interface of STB1. The P2P network interface of STB1 sends the frames to the video decoder 9 in STB1, which can then show the IPTV media stream to the user.

Figure 6 illustrates typical signalling required to initiate an IPTV broadcast with a further STB STB2. It is assumed that STB1 is already receiving an IPTV media stream from STB0. When the user of STB2 wishes to receive channel X, he sends an instruction to logic within STB2, which is relayed to a P2P network interface in STB2. The P2P network interface in STB2 sends a request join channel X to the STB manager 10. The STB manager 10 returns a peer list but no payload to STB2. The peer list includes STB0 and STB1, as these are both possible sources for the IPTV media stream. The

P2P function in STB2 then sends a request to each of STB0 and STB1 to join channel X. STB0 and STB1 each send a peer list and IPTV data stream to the P2P network interface in STB2, which passes the frames of the IPTV media stream to the video decoder.

It is advantageous for all peers in the P2P network to send each other “keep alive” messages, as illustrated in Figure 7, to ensure that each STB is included in the list of peers and can both send and receive IPTV media streams.

Note that the term “IPTV media stream” is used herein to refer to any kind of data having real time requirements, and includes Video on Demand, user defined TV content, interactive TV, interactive or co-operative games, or audio media. The media stream is to be delivered to the user such that the user can observe the media content at a constant rate without interruptions or delays. There is some latency in the P2P network, caused by buffers in each STB and the time it takes to establish communication between peers.

As previously discussed, when data is transferred over a real-time network such as a P2P network or a store and forward network, there always a risk that certain data may be lost. This problem is particularly acute in a P2P network with many nodes which might join and leave at any given point in time. Such packet loss is particularly problematic if key fragments of the traffic are missing. If such a loss occurs in a P2P network, it will generate a waste amount of control traffic from nodes trying to recover the lost fragment to no avail, and this behavior will persist throughout the network.

The problem can be solved by dedicating a certain amount of the bandwidth to recovery records. Standard methods of error recovery such as Reed-Solomon error correction (as described, for example, in http://en.wikipedia.org/wiki/Reed-Solomon_error_correction) can then be applied. The additional bandwidth may be a fixed amount, (*e.g.* 5% of all traffic). Alternatively each Maximum Transmission Unit (MTU) that is transmitted may be “padded out” with a set of recovery records whether or not it includes a key fragment.

An MTU is the largest packet that a given layer of communications protocol can pass onwards. In the context of IP, a “path MTU” is also defined, as the smallest MTU of any of the IP hops of a “path” between a source and destination. Put another way, the path MTU is the largest packet size that traverses this path without suffering fragmentation. When IPTV fragments are transmitted, in some circumstances fragments may be grouped together to form MTU packets, but there may well be some free space in each MTU packet which is not taken up by the fragments themselves. The free space should therefore be filled with recovery records. For example, one MTU in Ethernet is 1500 bytes, but one MPEG-TS block is normally 188 bytes. It is thus possible to fit seven MPEG-TS fragments inside one Ethernet MTU, leaving a 184 byte “slack space” that can be used for recovery records.

This can be understood with reference to Figure 8, which shows the transmission of nine fragments 41-49 within three MTU packets 31, 32, 33, each of which includes three fragments. Of the nine fragments 41-49, three are key fragments (fragments F2k, F6k and F7k 42, 46, 47) . Each of the three MTU packets 31, 32, 33 is also padded with a recovery record (R1, R2, R3) 51, 52, 53, each of which spans the key fragments found in fragments 1-9. In other words, each of the recovery records R1, R2, R3 contains information about fragments F2k, F6k and F7k. Under the principles of error recovery such as Reed-Solomon encoding, it is not necessary for the recovery records to contain complete fragments. In this example each recovery record 51, 52, 53 is half the size of any given key fragment. The recovery records are transmitted into the network by the media injector at the same time as the fragments.

Since error recovery details 51, 52, 53, for all of the key fragments are included in all of the MTUs 31, 32, 33, if one of the MTU’s is lost in the network, any key fragments included in that MTU can be recovered via the Reed Solomon recovery of remaining key fragments and recovery records.

For example, if the middle MTU 32 is lost from the network, key fragment F6k 46 will also be lost. However, the built in redundancy and use of standard Reed Solomon recovery techniques enables the recovery of F6k from the data contained in the key fragments either side of it which have been successfully received, and from the recovery records. In this case, therefore, fragment F6k can be reconstructed from fragments F2k, F7k and recovery records R1, R3. The missing key fragment F6k can then be retransmitted utilizing standard P2P methodology, thus minimizing any impact on the network and the end user experience.

If fragments are not transmitted in MTU packets, the same principle can still be applied. In addition to the media fragments themselves, a media injector inserts recovery record fragments which contain enough information for key fragments to be reconstructed using information from the key fragments either side of them, together with the information in the recovery records.

It will be appreciated that the error recovery can be extended to fragments other than key fragments, but it is the recovery of key fragments that is important to the experience of the end-user. Other non-key fragments will only give small glitches on the screen, whereas a missing fragment containing, for example, an I-frame, would invalidate subsequent fragments (such as those containing P and B frames) until the next key data fragment falls in the sequence.

Figure 9 is a flow diagram illustrating the insertion of data into the network by a media injector, and the reconstruction of a key fragment if it goes missing in the network.

Referring to Figure 10, there is shown a flow diagram illustrating the basic steps of the invention. The following numbering refers to the numbering in Figure 10:

S1. A video encoder encodes media as frames, which are passed to a media injector.

S2. The media injector encodes the frames as data fragments, and sends the data fragments into the P2P network. The media injector also inserts recovery records into the network.

S3. A node that wishes to receive the channel from the media stream receives the fragments and, in addition to decoding and showing the media stream to the user, stores the fragments in a buffer.

S4. The node determines whether any fragments, which are shortly due to be sent to the video decoder for rendering, are missing.

S5. The node sends a request for the missing fragments.

S6. If the missing fragments are not received after a predetermined delay period, the receiving node reconstructs the missing fragments using the information contained in adjacent fragments, together with the recovery records.

Referring to Figure 10, there is illustrated schematically a media injector. The media injector 23 comprises a media stream source 24, which may come from a Video on Demand server, a real time transmission or any other source. The media injector 23 further comprises a processor 25 for converting media frames into media stream fragments, and generating recovery record fragments in addition to the media stream fragments. The media stream fragments containing key frames are optionally tagged with key frame indicators. The media injector 23 further comprises a transmitter for sending the media stream fragments and recovery record fragments into the P2P network.

Referring to Figure 11, there is illustrated a STB 13. The STB 13 comprises a receiver 27 for receiving media stream fragments and recovery record fragments and a memory 28 for storing the media stream in a buffer. The STB 13 further comprises a processor 29 for determining which fragments of the media stream stored in the memory 28 comprise a key frame indicator, and which also identifies if any key fragments containing key frames are missing from the memory. The STB 13 further comprises a transmitter 30 for requesting missing key fragments. If the missing key fragments are still not received, the processor reconstructs them (for example using Reed Solomon error correction) from adjacent key fragments and the recovery record fragments.

The above description describes a scenario in which a media injector provides a media stream to STBs, each STB being a peer in the P2P network. It will be appreciated that the network may include other peers that are not STBs, for example routers, mobile telephones, web cameras and so on.

The above description describes the invention in a P2P network, although it will be appreciated by those of skill in the art that it may equally apply to a client-server based network as illustrated in Figure 1. In this instance, the STB can request missing frames from a server rather than peers in a network.

Thus the invention, at least in some embodiments, provides the utilisation of spare bandwidth or a dedicated portion of the bandwidth to introduce redundancy of key fragments, enabling reconstruction of such fragments in the case of permanent failure. Redundancy is added on top of the existing traffic, allowing for recovery of selected portions of the streams. The amount of control traffic to recover any given key fragment will be localized in a network sub graph. This provides stronger resilience towards churn in the P2P network.

Although various embodiments have been shown and described in detail, the claims are not limited to any particular embodiment or example. None of the above description should be read as implying that any particular element, step, or function is essential such that it must be included in the claims' scope. The scope of protection is defined by the claims.

CLAIMS:

1. A method of distributing media content in a communications network, the method comprising:

at a node in the network, transmitting a sequence of media stream fragments and recovery record fragments, the recovery record fragments containing sufficient information to enable missing media stream fragments to be reconstructed from the media stream fragments adjacent to the missing fragments.

2. The method of claim 1, wherein the node is a media injector.

3. A method of receiving media content in a communications network, the method comprising:

at a node in the network, receiving a sequence of media stream fragments, and recovery record fragments;

determining if any media stream fragments in the sequence are missing;

if it is determined that media stream fragments in the sequence are missing, sending a request to a further node requesting the missing media stream fragments;

if the missing media stream fragments cannot be recovered, reconstructing the missing media stream fragments from adjacent media stream fragments and the recovery record fragments.

4. The method of claim 3, further comprising receiving fragment data relating to each media stream fragment, the fragment data comprising an indicator of whether that media stream fragment is a key fragment comprising key data relating to the media stream, and prioritising the missing fragments on the basis of the fragment data.

5. The method of claim 4, wherein missing media stream fragments are reconstructed only if they are key fragments, and wherein missing key fragments are reconstructed from adjacent key fragments and the recovery records.

6. The method of claim 4 or 5, wherein key data comprises complete image data for a single image in the media stream.

7. The method of any of claims 4 to 6, wherein the media stream is an MPEG media stream and the key data comprises an I-frame.

8. The method of any of claims 3 to 7, further comprising storing the media stream fragments and recovery record fragments in a memory.

9. The method of any of claims 3 to 8, wherein the node is a Set Top Box.

10. The method of any preceding claim, wherein the media stream fragments are transmitted in Maximum Transmission Unit packets, and wherein the recovery record fragments use space within each Maximum Transmission Unit packet not used by the media stream fragments.

11. The method of any of claims 1 to 9, wherein the recovery record fragments utilise a fixed proportion of bandwidth.

12. The method of any preceding claim, wherein reconstruction of missing fragments is carried out using Reed-Solomon error correction.

13. The method of any preceding claim, wherein the communications network is a peer to peer communications network, and each node is a peer node.

14. A media injector node for distributing media content in a communications network, the node comprising:

a receiver for receiving a media stream;

a processor for partitioning the media stream into a sequence of media stream fragments and generating recovery record fragments, the recovery record fragments providing enough information to enable the reconstruction of a missing media stream fragment from adjacent media stream fragments; and

a transmitter for sending the media stream fragments and recovery record fragments to at least one node located in the communications network.

15. The media injector node of claim 14, wherein the processor is arranged to add, to each media stream fragment that contains key data relating to the media stream, metadata indicating that the fragment comprises key data relating to the media stream.

16. A node for use in an communications network, the node comprising:

a receiver for receiving a sequence of media stream fragments, and recovery record fragments;

a processor for determining if any fragments in the sequence are missing; and

a transmitter for transmitting a request to a further node requesting the missing fragments;

wherein the processor is arranged to reconstruct the missing fragments using the recovery record fragments and adjacent media stream fragments.

17. The node of claim 16, wherein the receiver is arranged to receive fragment data relating to each media stream fragment, the fragment data comprising an indicator of whether that media stream fragment is a key fragment comprising key data relating to the media stream, and wherein the processor is arranged to prioritise the missing fragments on the basis of the fragment data.

18. The node according to claim 16 or 17, wherein the node is a Set Top Box.

19. An apparatus for use in distributing media over a communications network, the apparatus comprising means for performing the method as claimed in any one of claims 1 to 13.

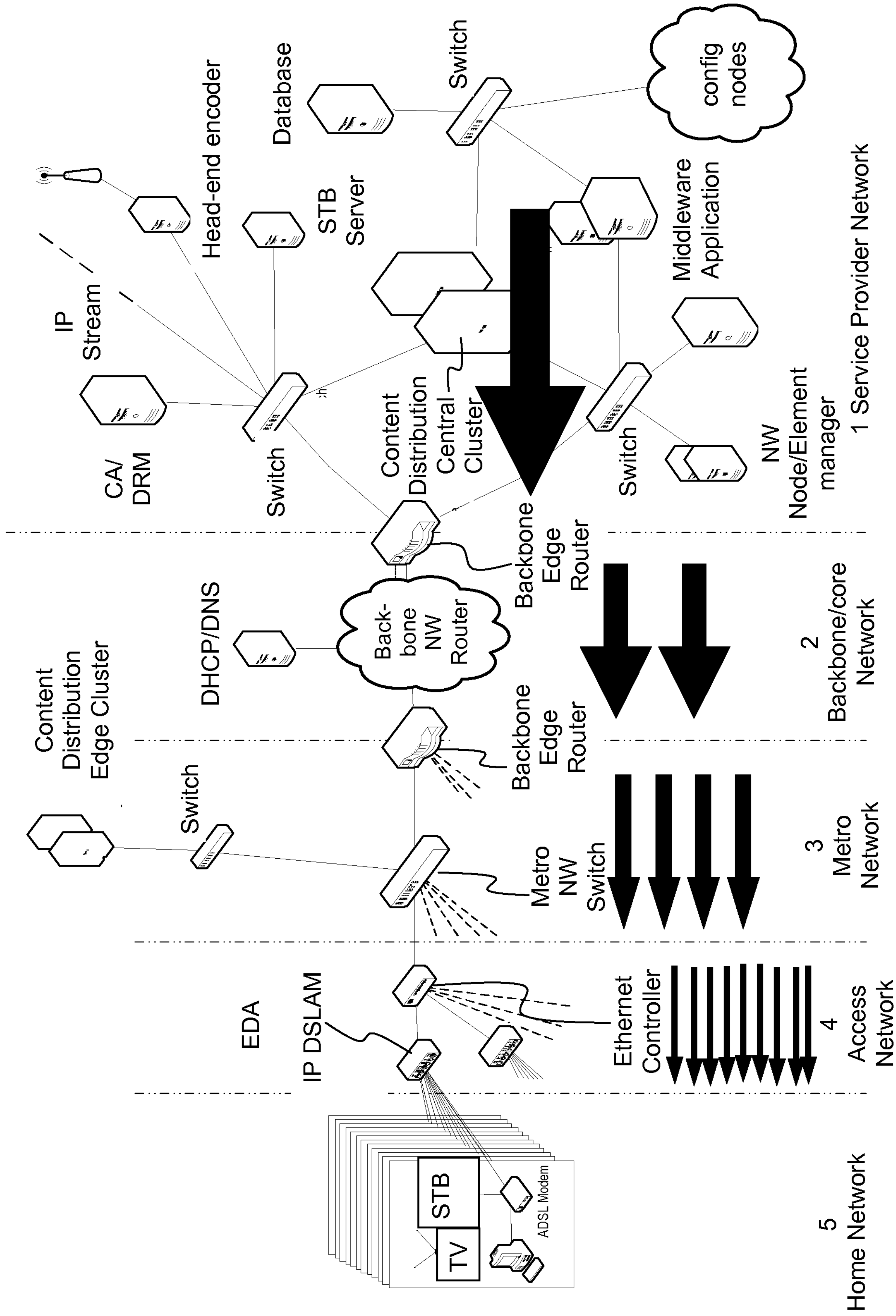
20. A program for controlling an apparatus to perform a method as claimed in any one of claims 1 to 13.

21. A program which, when loaded into an apparatus, causes the apparatus to become an apparatus as claimed in claim 19.

22. A program as claimed in claim 20 or 21, carried on a carrier medium.

23. A program as claimed in claim 22, wherein the carrier medium is a storage medium.

24. A storage medium containing a program as claimed in any one of claims 20 to 23.



IP Stream
Head-end encoder
STB Server
Database
Switch
config nodes
CA/ DRM
Switch
Content Distribution Central Cluster
Middleware Application
Switch
NW Node/Element manager
1 Service Provider Network
Backbone Edge Router
DHCP/DNS
Back-bone NW Router
Backbone Edge Router
2 Backbone/core Network
Content Distribution Edge Cluster
Switch
Metro NW Switch
3 Metro Network
EDA
IP DSLAM
Ethernet Controller
4 Access Network
STB
TV
ADSL Modem
5 Home Network


Figure 1

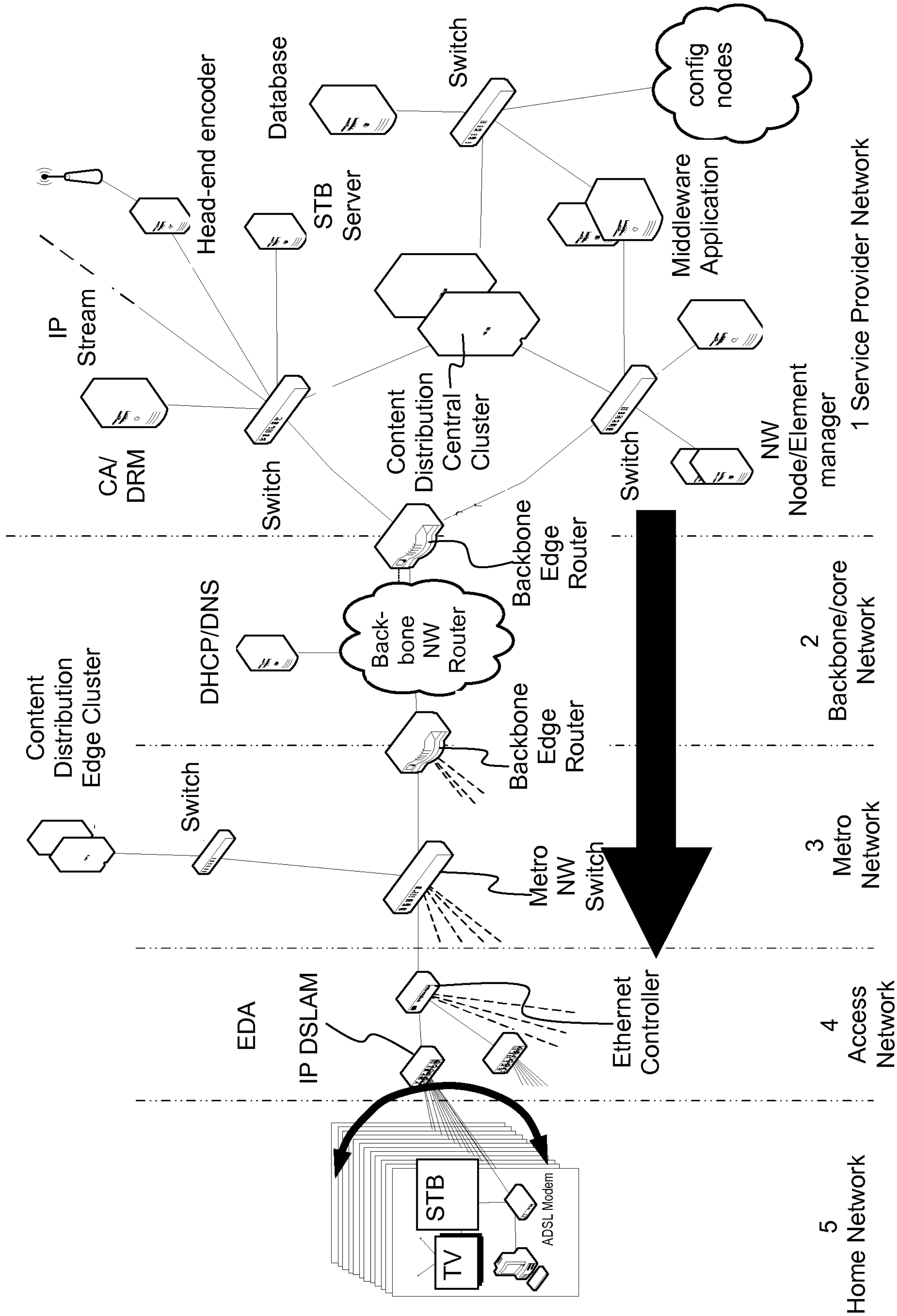

IP
Stream
Head-end encoder
Database
Switch
config
nodes
STB
Server
CA/
DRM
Switch
Content
Distribution
Central
Cluster
Middleware
Application
Switch
NW
Node/Element
manager
1 Service Provider Network
Backbone
Edge
Router
DHCP/DNS
Back-
bone
NW
Router
2
Backbone/core
Network
Content
Distribution
Edge Cluster
Switch
Backbone
Edge
Router
Metro
NW
Switch
3
Metro
Network
EDA
IP DSLAM
Ethernet
Controller
4
Access
Network
STB
TV
ADSL Modem
5
Home Network

Figure 2

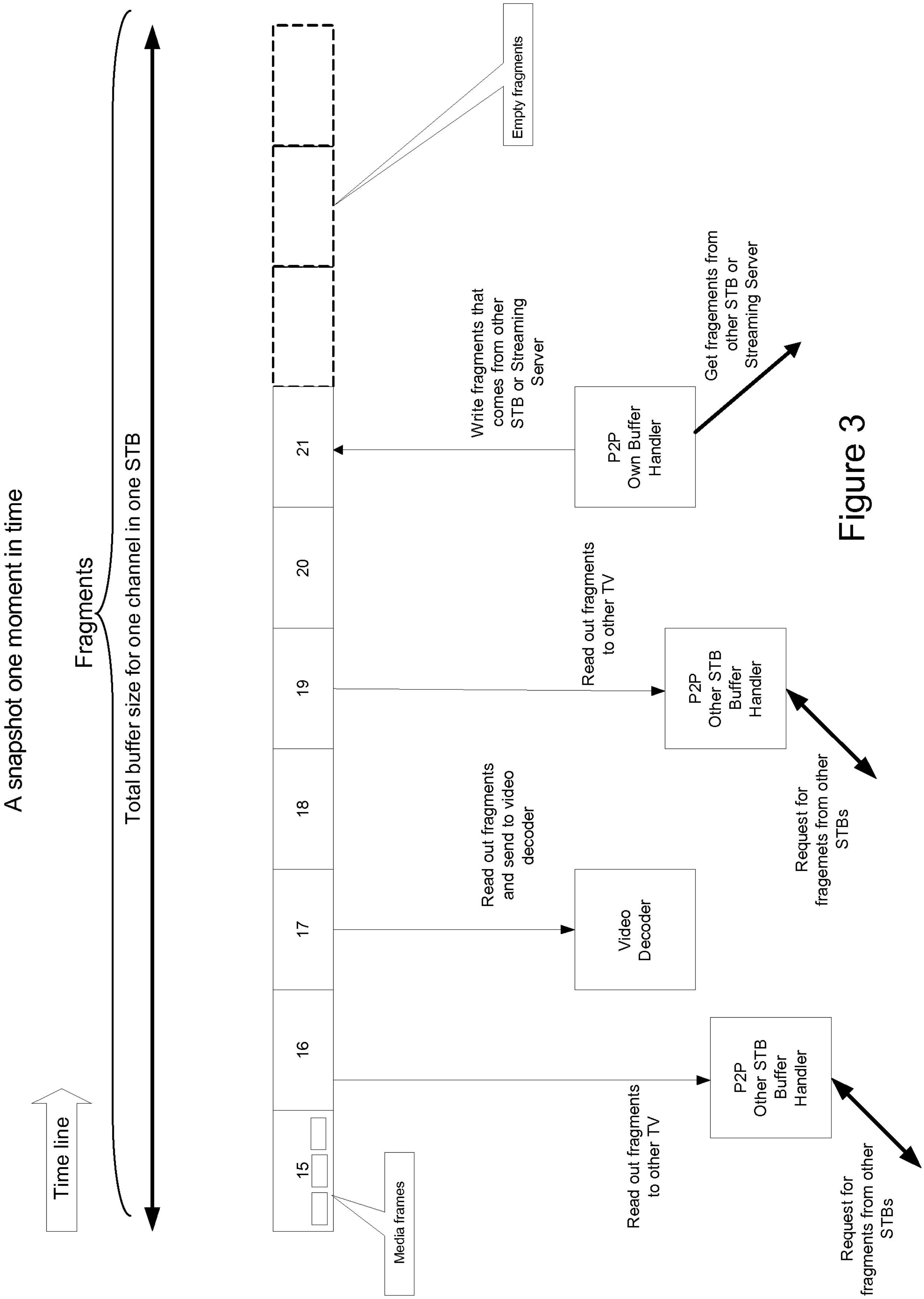

A snapshot one moment in time
Fragments
Total buffer size for one channel in one STB
Time line
15
16
17
18
19
20
21
Media frames
Empty fragments
Read out fragments to other TV
P2P Other STB Buffer Handler
Request for fragments from other STBs
Read out fragments and send to video decoder
Video Decoder
Read out fragments to other TV
P2P Other STB Buffer Handler
Request for fragemets from other STBs
Write fragments that comes from other STB or Streaming Server
P2P Own Buffer Handler
Get fragements from other STB or Streaming Server

Figure 3

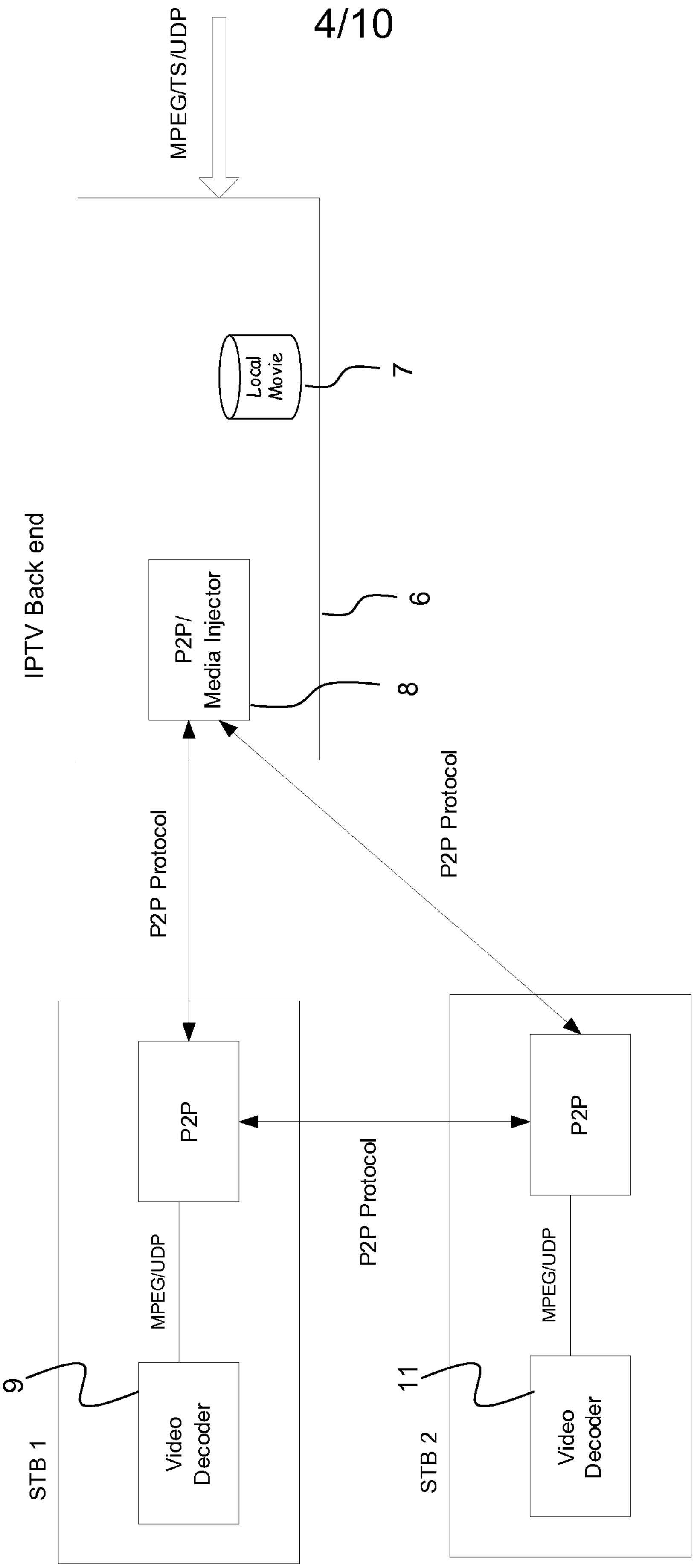

MPEG/TS/UDP
IPTV Back end
Local Movie
7
P2P/ Media Injector
6
8
P2P Protocol
P2P Protocol
STB 1
Video Decoder
9
MPEG/UDP
P2P
P2P Protocol
STB 2
Video Decoder
11
MPEG/UDP
P2P

Figure 4

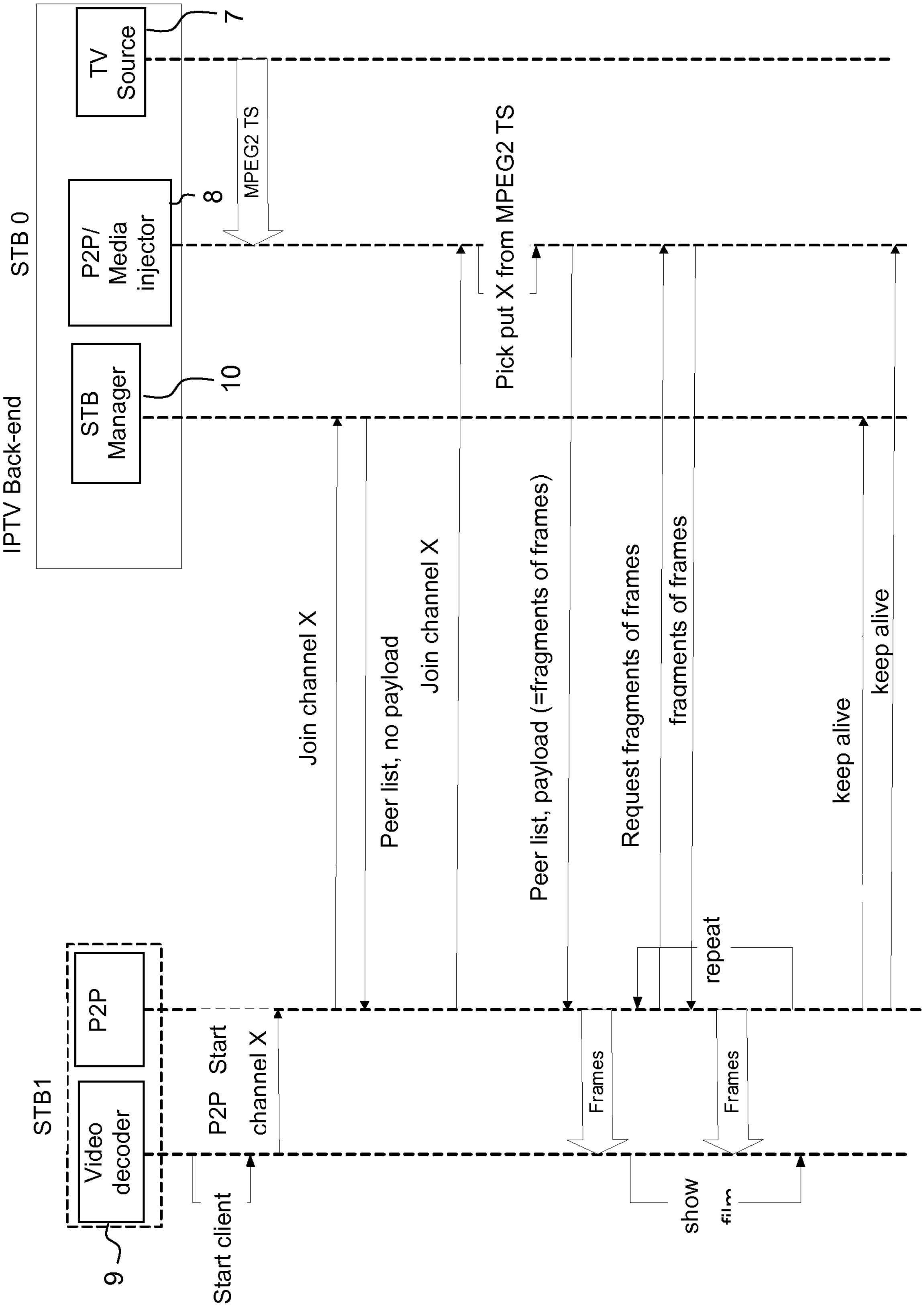

IPTV Back-end
STB 0
TV Source
7
P2P/ Media injector
8
STB Manager
10
MPEG2 TS
STB1
Video decoder
9
P2P
Start client
P2P Start channel X
Join channel X
Peer list, no payload
Join channel X
Pick put X from MPEG2 TS
Peer list, payload (=fragments of frames)
Request fragments of frames
fragments of frames
repeat
Frames
Frames
show
keep alive
keep alive

Figure 5

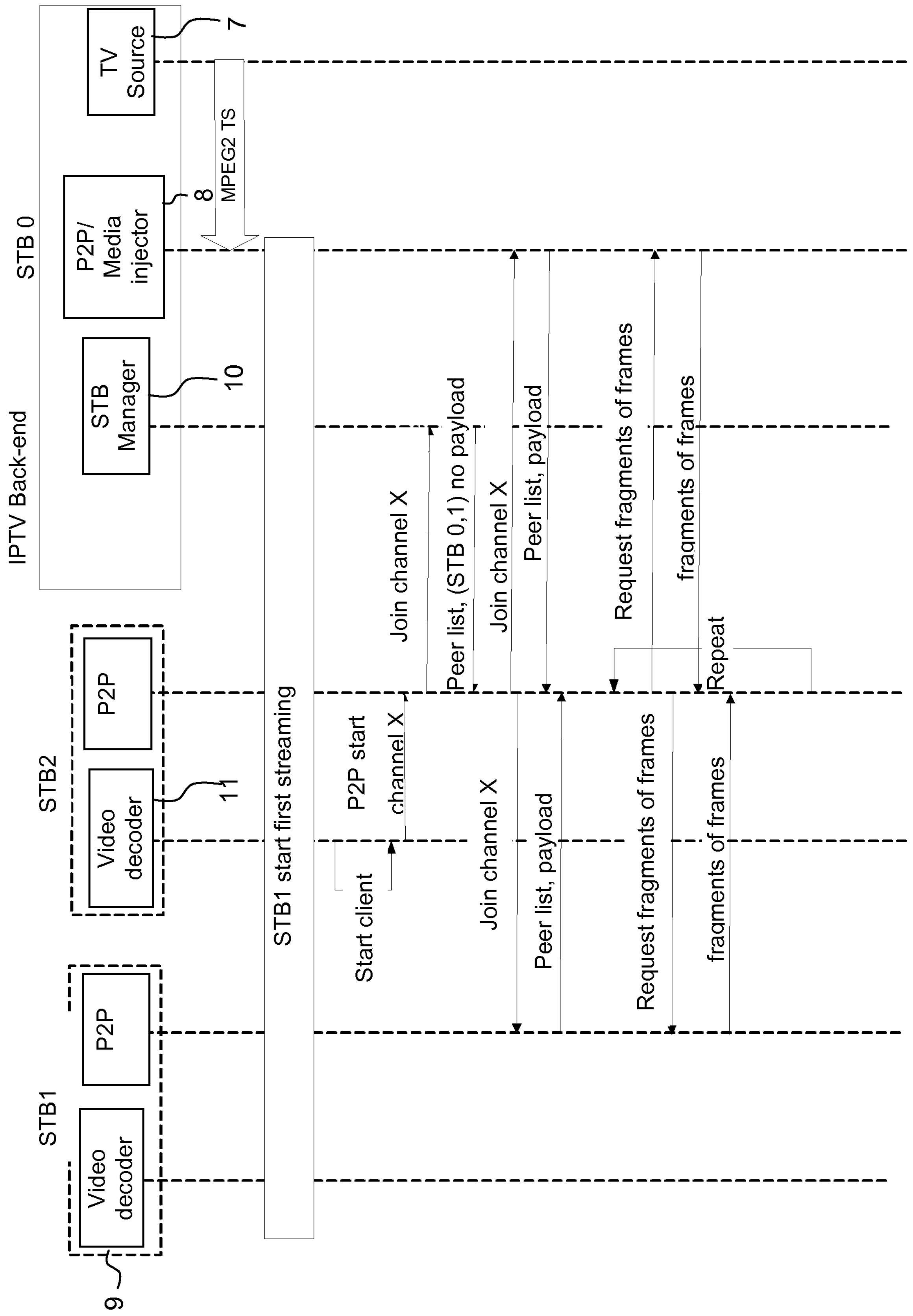

IPTV Back-end
STB 0
TV Source
7
P2P/ Media injector
8
MPEG2 TS
STB Manager
10
STB2
P2P
Video decoder
11
STB1
P2P
Video decoder
9
STB1 start first streaming
Start client
P2P start channel X
Join channel X
Peer list, (STB 0,1) no payload
Join channel X
Peer list, payload
Request fragments of frames
fragments of frames
Repeat
Join channel X
Peer list, payload
Request fragments of frames
fragments of frames

Figure 6

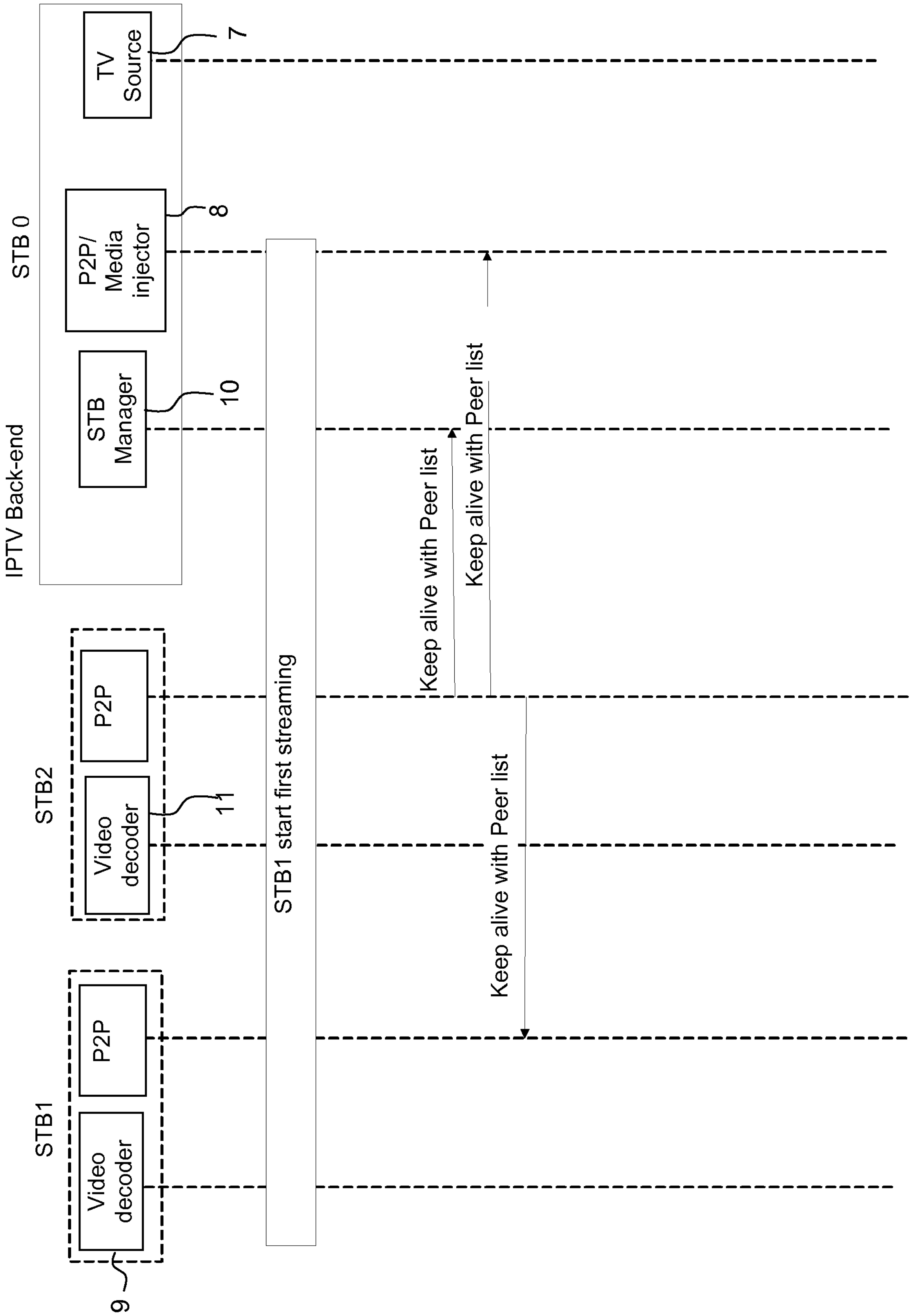

IPTV Back-end
STB 0
TV Source
7
P2P/ Media injector
8
STB Manager
10
STB2
P2P
Video decoder
11
STB1
P2P
Video decoder
9
STB1 start first streaming
Keep alive with Peer list
Keep alive with Peer list
Keep alive with Peer list

Figure 7

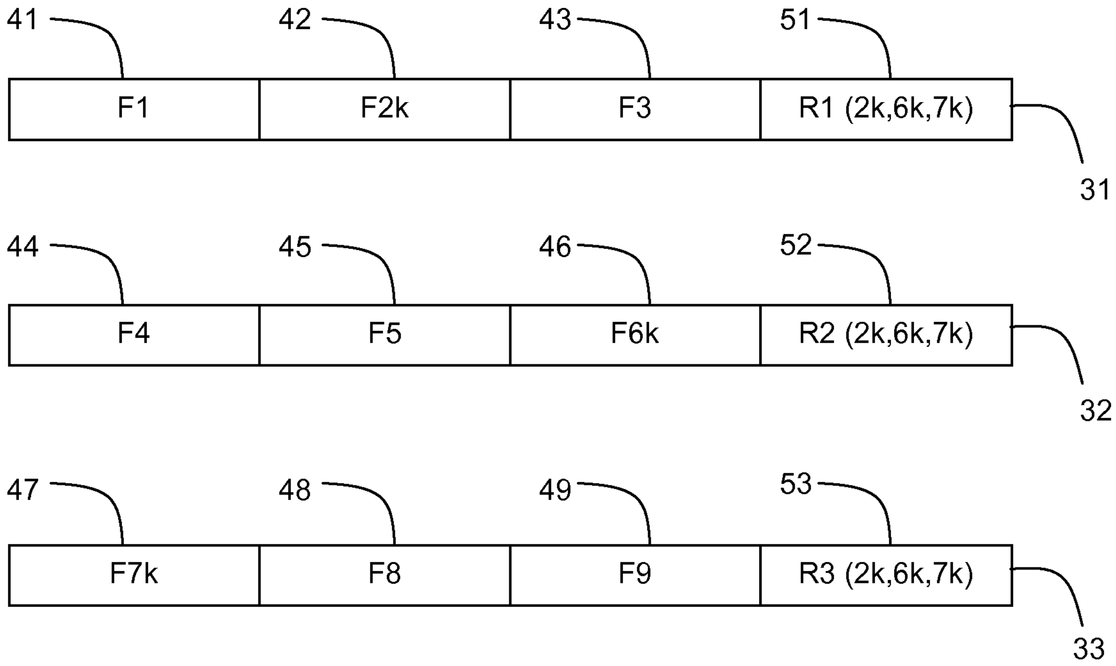

41
42
43
51
F1
F2k
F3
R1 (2k,6k,7k)
31
44
45
46
52
F4
F5
F6k
R2 (2k,6k,7k)
32
47
48
49
53
F7k
F8
F9
R3 (2k,6k,7k)
33

Figure 8

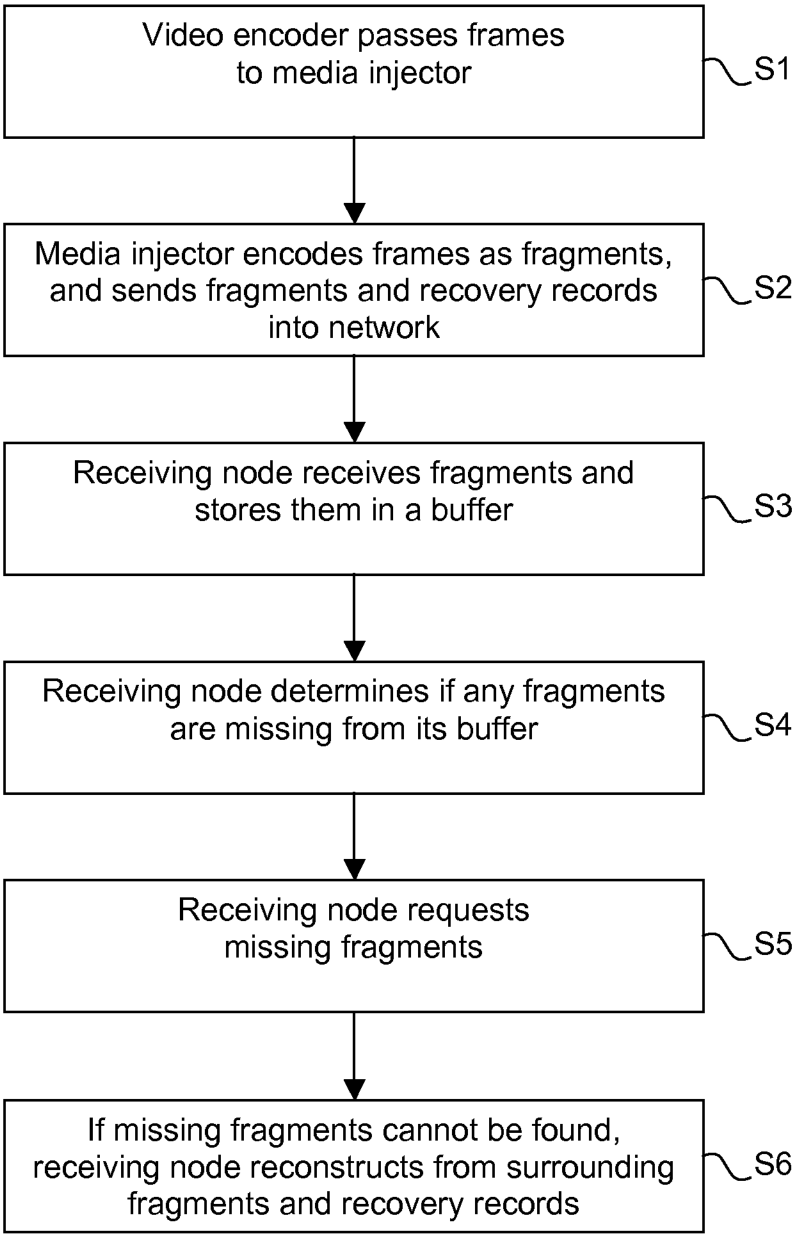

Video encoder passes frames to media injector
S1
Media injector encodes frames as fragments, and sends fragments and recovery records into network
S2
Receiving node receives fragments and stores them in a buffer
S3
Receiving node determines if any fragments are missing from its buffer
S4
Receiving node requests missing fragments
S5
If missing fragments cannot be found, receiving node reconstructs from surrounding fragments and recovery records
S6

Figure 9

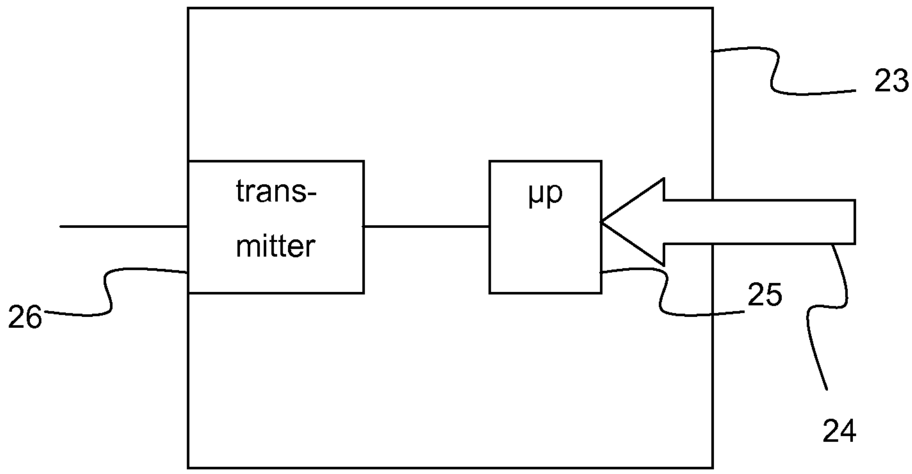



Figure 10

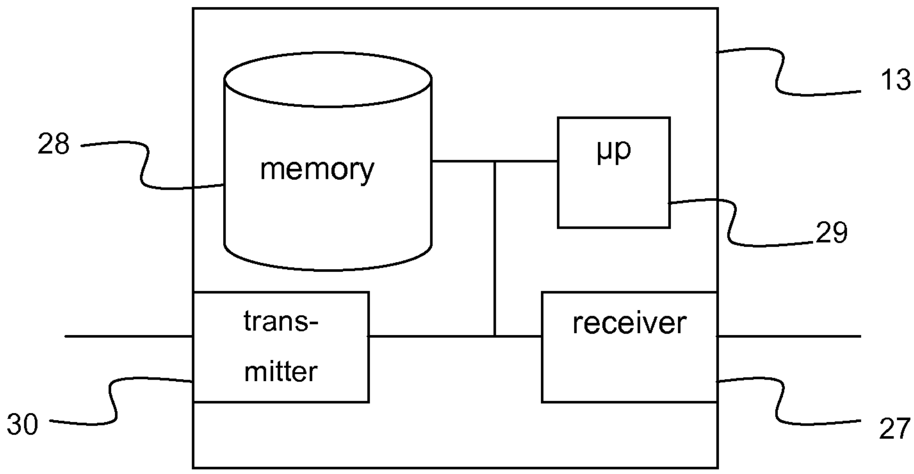



Figure 11

# INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2008/052776

**A. CLASSIFICATION OF SUBJECT MATTER**
INV. H04N7/173 H04L29/06

According to International Patent Classification (IPC) or to both national classification and IPC

**B. FIELDS SEARCHED**

Minimum documentation searched (classification system followed by classification symbols)
H04N H04L

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

EPO-Internal

**C. DOCUMENTS CONSIDERED TO BE RELEVANT**

| Category* | Citation of document, with indication, where appropriate, of the relevant passages | Relevant to claim No. |
|---|---|---|
| X | WU ET AL: "Stochastic analysis of the interplay between object maintenance and churn" COMPUTER COMMUNICATIONS, ELSEVIER SCIENCE PUBLISHERS BV, AMSTERDAM, NL, vol. 31, no. 2, 18 January 2008 (2008-01-18), pages 220-239, XP022426917 ISSN: 0140-3664 | 1-3, 8-14,16, 18-24 |
| Y | page 222, left-hand column, paragraph 2.2 - page 223, left-hand column, paragraph 3 | 4-7,15, 17 |
| Y | WO 2004/044710 A (SUPRACOMM INC [US]; RIDEOUT NEIL [CA]) 27 May 2004 (2004-05-27) page 11, line 16 - page 12, line 13 page 43, line 6 - page 44, line 9 | 4-7,15, 17 |

-/--

[X] Further documents are listed in the continuation of Box C.

[X] See patent family annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

| Date of the actual completion of the international search | Date of mailing of the international search report |
|---|---|
| 18 September 2008 | 25/09/2008 |

| Name and mailing address of the ISA/ | Authorized officer |
|---|---|
| European Patent Office, P.B. 5818 Patentlaan 2 NL – 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016 | Schoeyer, Marnix |

# INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2008/052776

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

| Category* | Citation of document, with indication, where appropriate, of the relevant passages | Relevant to claim No. |
|---|---|---|
| A | AKAR G B ET AL: "Transport Methods in 3DTV-A Survey" IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, IEEE SERVICE CENTER, PISCATAWAY, NJ, US, vol. 17, no. 11, 1 November 2007 (2007-11-01), pages 1622-1630, XP011195152 ISSN: 1051-8215 the whole document ----- | 1-24 |

# INTERNATIONAL SEARCH REPORT

**Information on patent family members**

International application No
PCT/EP2008/052776

| Patent document cited in search report | Publication date | Patent family member(s) | Publication date |
|---|---|---|---|
| WO 2004044710 A | 27-05-2004 | AU 2003295515 A1 | 03-06-2004 |
| | | CA 2505936 A1 | 27-05-2004 |