

(19) World Intellectual Property Organization  
International Bureau



(43) International Publication Date  
2 July 2009 (02.07.2009)

PCT

(10) International Publication Number  
**WO 2009/080116 A1**

(51) International Patent Classification:  
H04L 29/06 (2006.01)

(21) International Application Number:  
PCT/EP2007/064451

(22) International Filing Date:  
21 December 2007 (21.12.2007)

(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]; Ängermannagatan 119, S-162 64 Vällingby (SE). SKOG, Robert [SE/SE]; Gullvivegränd 7, S-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, 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, HU, IE, IS, IT, LT, LU, LV, MC, MT, NL, 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

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

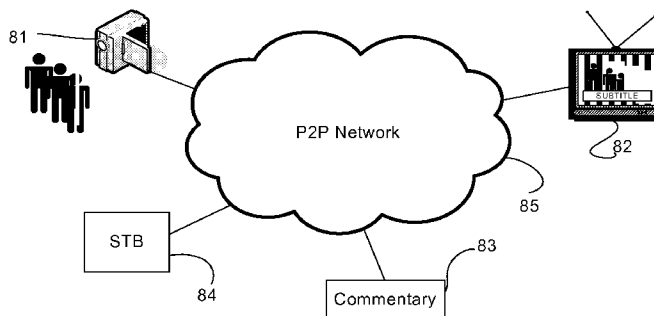


Figure 8

(57) Abstract: A method and apparatus for distributing media over a P2P IPTV network is described. The media is encoded into data fragments. An internal time relative to an epoch time reference point is determined at a media injector. Each data fragment is tagged with a timestamp corresponding to the internal time at the moment the data is encoded. The data fragments are transmitted into the network, and received at a peer node, which stores the data fragments in a buffer. A suitable local time reference point, corresponding to the epoch time reference point, is determined at the peer node, and the frames encoded in the data fragments are passed to a media decoder on the basis of the timestamps of the data fragments, so that each frame is passed to the media decoder at a local time corresponding to the timestamp attached to the data fragment or fragments into which that frame is encoded.

WO 2009/080116 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 distribution of IPTV using a Peer to Peer 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.

5

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 only sends data but does not receive data from the peers. This is illustrated in Figure 3, which is a schematic representation of a simple IPTV P2P network 1. The network 1 includes an IPTV back-end 6 and two STBs STB1 and STB2. Each STB includes a P2P network interface 12, 13 to which is connected a video decoder 9, 11. In this example, STB1 receives the IPTV media stream from both STB2 and the IPTV back-end 6, which injects either streaming content or content from a database 7 using a P2P media injector 8. Note that other network nodes (in addition to nodes in STBs) may be peers in the network.

10  
15

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 generated 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.

20

Compressed video media generally consists of a series of frames containing the information to be displayed on a user's screen. Each frame can be considered as a "picture" displayed on the screen. In most video compression formats, such as in ITU-T VCEG or ISO/IEC MPEG video standards, only the differences between successive pictures are usually encoded. For example, in a scene in which a person walks past a stationary background, only the moving portions of the picture are represented in each frame (either using motion compensation or as image data or as a combination of the two, depending on which representation requires fewer bits to adequately represent the picture). The parts of the scene that are not changing do not need to be sent repeatedly.

30

35

For example, MPEG media streams contain different frames, such as I-frames (“intra” frames), P-frames (“predicted” frames) and B-frames (“bi-predictive” or “bi-directional” 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  
5 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-frames 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. In  
10 some systems, 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.

The media stream includes payload data and metadata. The payload data is the media  
15 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  
20 portions of the media stream containing both the payload data and the metadata.

The latency in the P2P network can potentially give rise to problems when media is being provided by several sources or in large networks with many hops. In the example described above, STB2 may receive some fragments of the media stream for  
25 a particular channel from the media injector 8, and some from STB1. These fragments are stored in a buffer (not shown) in STB2, and re-assembled before being passed to the video decoder (or re-assembled in the video decoder). Problems can arise if the video decoder does not know the order in which the fragments should be re-assembled. In addition, fragments may be transmitted at different speeds, and it is  
30 important that the video decoder can play the media back at the correct speed. Traditionally, video decoders have very limited memory, and must therefore be “fed” frames or fragments at the correct rate for decoding.

Furthermore, it will be appreciated that content may be introduced into the network  
35 from several different sources. For example, it may be desirable to provide commentary on a football match in “pseudo” real time in different languages, with

statistics being relayed at the same time, even though there may be a couple of seconds delay in computing these statistics. This requires the use of multiple media injectors. The user will wish to view the images, hear the commentary and view the statistics concurrently (*i.e.*, as far as the user is concerned, the images, commentary and statistics must be synchronised with one another).

There is thus a need for a system which enables the synchronisation of media streams from different sources and which arrive at different speeds, in order to compensate for high latency and "jitter" in transmission.

### SUMMARY

In accordance with a first aspect of the present invention, there is provided a media injector for injecting streaming media in the form of data fragments into a non-linear network, preferably a Peer-to-Peer IP Television network. The media injector comprises a clock arranged to determine an internal time relative to an epoch time reference point; a tagging processor arranged to tag each data fragment with a timestamp, the timestamp corresponding to the internal time; and a transmitter arranged to transmit the tagged data fragments into the network. The tagging processor is preferably arranged to include the timestamp in a metadata portion of each data fragment. The epoch time reference point can be any arbitrarily chosen point in time, for example the start of that day, or midnight at 28 January 2006.

Preferably, the media injector receives media frames (*e.g.* from a media encoder) and encodes these frames into the data fragments. It is likely that more than one fragment will be required for each frame. The timestamp on any particular fragment may be the internal time at which the frame, of which all or a part is encoded into that fragment, was received by the media injector.

The fragments may also be tagged (*e.g.* in the metadata portion) with a counter to identify the order in which the fragments are encoded.

The tagged data fragments may then be received by a node in the network, and the timestamps can be used to ensure that the received data fragments are decoded at the correct rate. It may be possible to determine the correct order from the timestamps, although it is preferable to use the counter to determine the correct order and the

timestamp to control the rate. In accordance with a second aspect of the present invention there is provided a node for use in a non-linear network delivering streaming media, and preferably a Peer-to-Peer IP Television network. The node comprises a receiver arranged to receive data fragments from the network, each data fragment  
5 tagged with a timestamp providing a time interval relative to an epoch time reference point. A buffer is arranged to store the received data fragments. A processor is arranged to determine, from the timestamps contained in the data fragments, a suitable local time reference point to correspond to the epoch time reference point. A local clock is arranged to determine a local time relative to the local time reference point,  
10 and a traffic shape recovery unit is arranged to pass media frames encoded in the data fragments to a media decoder on the basis of the timestamps of the data fragments, so that each frame is passed to the media decoder at a local time corresponding to the timestamp attached to the data fragment or fragments in which that frame is encoded.

15 The order and/or rate in which the frames are passed to the media decoder may be different to the order and/or rate in which the data fragments are received from the network.

The processor is preferably arranged to determine, from the timestamps of the  
20 received data fragments, a maximum latency time interval. The local time reference point may then be determined as the time of receipt of a first data fragment plus the maximum latency time interval.

The receiver may be arranged to receive data fragments from more than one source.  
25

It is also possible to insert more than one media stream into the network. For example, a first media stream may include video images of, for example, a football game, with a second media stream providing commentary. Alternatively, a second media stream may include subtitles.

30

It will be appreciated that the technique may also be used in top down and store and forward networks.

In accordance with a third aspect of the present invention there is provided a subsidiary  
35 media injector for receiving data fragments of a first media stream from a Peer-to-Peer network, preferably a P2P IPTV network, and inserting data fragments of a second

media stream into the network, the second media stream intended to complement the first media stream. The subsidiary media injector comprises a receiver arranged to receive data fragments of the first media stream from the network, each data fragment tagged with a first timestamp providing a time interval relative to an epoch time reference point. A buffer is arranged to store the received data fragments. A processor is arranged to determine, from the timestamps contained in the data fragments of the first media stream, a suitable local time reference point to correspond to the epoch time reference point, and a local clock is arranged to determine a local time relative to the local time reference point. A tagging processor tags each data fragment of the second media stream with a second timestamp, the second timestamp corresponding to the local time elapsed compared to the local time reference point. A transmitter is arranged to transmit the tagged data fragments of the second media stream into the network.

According to a fourth aspect of the present invention, there is provided a method of controlling delivery of streaming media in a non-linear network, preferably a Peer-to-Peer IP Television network, comprising: encoding the media into data fragments, each data fragment containing all or part of a media frame; at a media injector connected to the network, determining an internal time relative to an epoch time reference point; at the media injector, tagging each data fragment with a timestamp corresponding to the internal time at the moment the data is encoded; transmitting the data fragments from the media injector into the network; receiving the data fragments at a node connected to the network; storing the data fragments in a buffer of the node; at the node, determining a suitable local time reference point to correspond to the epoch time reference point; recovering the media frames from the data fragments; and passing the media frames to a media decoder on the basis of the timestamps of the data fragments, so that each frame is passed to the media decoder at a local time corresponding to the timestamp attached to the data fragment or fragments in which that frame is encoded.

30

In one embodiment, the data fragments may be received at a subsidiary media injector and stored in a buffer of the subsidiary media injector. A time reference point local to the subsidiary media injector may be determined to correspond to the epoch time reference point. Data fragments of a second media stream may then be tagged with second timestamps corresponding to the time elapsed compared to the time reference point local to the subsidiary media injector, and transmitted into the network.

35

According to a fifth aspect of the present invention, there is provided apparatus for use in a network, the apparatus comprising means for performing a method according to the third aspect of the present invention.

- 5 According to a sixth aspect of the present invention, there is provided a program for controlling an apparatus to perform a method according to the third aspect of the present invention.

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

The program may be carried on a carrier medium, which may be a storage medium or a transmission medium.

According to an eighth aspect of the present invention, there is provided an apparatus programmed by a program according to the sixth or seventh aspect of the present  
15 invention.

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

20 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  
25 of IPTV in a peer to peer network;

Figure 3 illustrates schematically in a block diagram a media injector and two Set Top Boxes;

Figure 4 illustrates schematically in a block diagram the signalling required to initiate an IPTV broadcast with a first Set Top Box;

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

Figure 6 illustrates schematically in a block diagram keep alive messages sent by a Set Top Box;

Figure 7 is a schematic illustration of a media injector and peer node in a P2P IPTV network;

5 Figure 8 is a schematic illustration of a P2P IPTV network having multiple media injectors;

Figure 9 is a schematic block diagram of a STB;

Figure 10 is a flow diagram illustrating the tagging and transmission of data fragments by a media injector; and

10 Figure 11 is a flow diagram illustrating the receiving and re-ordering of data fragments by a STB.

#### DETAILED DESCRIPTION

15 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 to obscure the description with unnecessary detail. Moreover, individual blocks are shown in some of the drawings. It will be appreciated  
20 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.

25 Figure 4 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 12 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 network interface 12 in STB1, but no IPTV media  
30 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 hereinafter referred to as STB0. The network interface 12 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 shown in Figure 3), and sends a peer list and an IPTV media stream  
35 comprising fragments of frames to the network interface 12 of STB1. The network

interface 12 sends the frames to the video decoder 9 in STB1, which can then show the IPTV media stream to the user.

Figure 5 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, she sends an instruction to the logic within STB2, which is relayed to the network interface 13 in STB2. The network interface 13 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 network interface 13 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 network interface 13 in STB2, which passes the frames of the IPTV media stream to the video decoder.

15

It is preferred that all peers in the P2P network send each other "keep alive" messages, as illustrated in Figure 6, to ensure that each STB is included in the list of peers and can both send and receive IPTV media streams.

20 Compensation for high latency and jitter in transmission can be understood with reference to Figure 7, which is a representation of some portions only of a high latency P2P IPTV network. It will be appreciated that further routers and nodes will be present, but these are not shown for the sake of simplicity. The network includes an IPTV server 6 and a Set Top Box STB1, connected via a high latency P2P network 22 (shown schematically). The IPTV server 6 includes a video encoder 21 and media injector 8. Media frames are passed from the video encoder 21 to the media injector 8 for encoding into fragments and transmission into the network. The media injector 8 runs an internal clock, and tags the metadata of each fragment with a timestamp taken from this internal clock. The time can be relative to any arbitrary chosen time (e.g. the time of receipt of the first frame) but is consistent for the media injector. The time used is the time (relative to the arbitrary chosen time) that the frame is received by the media injector. For example, the arbitrary chosen time may be the time the first frame is received by the media injector. In this example, frames are sent to the media injector at T=0, 2, 4, 7, 8, 12 and 13 ms.

35

It will be appreciated that there need not be a direct correlation between a single media frame and a single data fragment. It is frequently the case that one frame is encoded into many fragments. In this case, each fragment corresponding to a particular frame will be tagged with the same timestamp – *i.e.* the time that the particular frame was received by the media injector 8. The fragments are preferably also tagged with a counter to indicate the order in which they are encoded by the media injector.

The data fragments are then sent through the network 22 and received by the network interface 12 of the STB1. The transmission through the network introduces jitter, bursting and packet drops, and the data fragments take different times to arrive. In this example, it will be noted that the fragment labelled T=4 arrives at the STB1 before the fragment labelled T=2. The STB1 saves all of the fragments into a buffer (not shown), *e.g.* a sliding window buffer or a ring buffer.

Once the first fragment has been received (labelled T=0) the network interface 12 waits a predetermined time interval representing the maximum latency of the system (*i.e.* the greatest expected time between data fragments) before forwarding the frame encoded in the first fragment(s) to the video decoder 9 and starting its own internal clock at local time reference LTD=0. The local clock then mirrors the clock run by the media injector: the network interface knows that the frame from fragment(s) labelled T=2 should be sent to the video decoder 2 ms later than the frame from fragment(s) labelled T=0. Similarly, the frame from fragment(s) labelled T=7 should be sent to the decoder 3 ms after the fragment labelled T=4.

It will be noted that there is no absolute synchronisation between the clock of the STB1 and the clock of the media injector 8: the packets are labelled with a time reference from the media injector's clock, and the local time elapsed after a locally chosen reference point, corresponding to T=0, at the network interface 12 "reproduces" that clock some time interval later. It will also be appreciated that fragments which arrive at the STB1 via other peers in the network can be inserted in the correct order, because they will all have the original timestamp from the media injector 8. In fact, the correct order of fragments can be also be recovered from the counters included in the fragment metadata. However, as previously discussed, it is important that the video decoder 9 receives frames at the correct rate, and this is achieved by ensuring that the frames are passed to the video decoder at a time after T=0 corresponding to their timestamp on their corresponding fragment(s).

There is no need for the media content itself to have any time stamp included with it: the data fragments include the time stamp, and the traffic shape can thus be recovered without any knowledge of the codec.

5

Figure 8 illustrates how an arrangement in which different media injectors 81, 82, 83, transmit data into a P2P network 85. For example, suppose a football match is filmed, and the images are sent into the P2P network from an image media injector 81. A commentator 83 in a different country receives these images and provides a commentary, which is transmitted back into the network. The original image data fragments sent by the image media injector 81 are tagged with a time stamp taken from the clock of the first media injector 81. The commentator 83 deduces a reference point in time, and (as with the STB1 of Figure 7) uses this reference point to run its own local clock as a virtual equivalent of the image media injector's clock. The data fragments containing the commentary are transmitted into the network with timestamps matching those of the first media injector 81, even though the commentator 83 need not be synchronised with the first media injector 81.

In other words, suppose the first image data fragment sent from the image media injector 81 is tagged with a timestamp  $T=0$ . When the commentator peer 83 receives this fragment, it sets its own local time reference to read  $LTD=0$  (as with Figure 7). Any commentary transmitted into the network corresponding to the first image data fragment (or the frame encoded therein) will also be tagged  $T=0$ . The same will apply to frames from the later data image fragments: they will be sorted into the correct order and replayed to the commentator at the correct speed using the local clock. Outgoing commentary data fragments are tagged using the local clock of the commentator 83.

At a STB 84, the user is able to request image data from the image media injector 81 and, separately, commentary in his own language from the commentator 83. Since a commentary data fragment with timestamp  $T=0$  will match an image data fragment with timestamp  $T=0$ , the STB 84 is able to reconstruct the images and commentary in the correct order and at the correct speed, by ensuring that its own local clock also runs from the same reference point (as previously described with reference to Figure 7).

Similarly, other media injectors may also provide other features such as subtitling. The subtitles may have been previously provided, and stored in a database. The user of

the STB 84 can request the subtitles from a separate provider 82 to the images: all that is required to ensure that the two match is that the subtitle provider 82 chooses a reference point for his local clock so that a subtitle data fragment tagged with T=0 matches an image data fragment sent with T=0 from the image media injector 81.

5 Effectively, subsidiary media injectors 82, 83 synchronise their local clocks so that they run from the same reference point as the image media injector 81.

Figure 9 illustrates a set-top box 31, which may be any of the STBs shown in the other figures. In addition to the functions normally found in a set-top box known in the art,

10 the set-top box 31 comprises a buffer 33 for storing fragments received from media injectors and/or other set-top boxes or other peers, as discussed above. The set-top box 31 also comprises a transmitter unit 35 and a receiver unit 37, connected to the buffer 33, for communicating with the other set-top boxes in the network. The set-top box 31 also comprises a control unit for controlling the functions of the transmitter unit

15 35, the receiver unit 37 and the buffer 33. In particular, the control unit is able to identify time stamps on data fragments stored in the buffer, and ensure that frames are passed to a video decoder (not shown) in the correct order and at the correct speed.

The receiver may be implemented as a software module in a television set, which will

20 then be able to receive IPTV from the network and display it to the user. Alternatively, the set-top box is implemented as a software module, for example in a personal computer or other terminal having data processing capabilities. The stream can then be forwarded from the set-top box to any display unit, including a television set, or the computer's own display for display to the user.

25

Figure 10 is a flow chart illustrating the actions carried out by the media injector 8 in the IPTV server 6 shown in Figure 7.

S1: Media frames are received from the video encoder 21.

30

S2: The frames are encoded into data fragments.

S3: Each data fragment is tagged with a timestamp, the timestamp reflecting the time, relative to a reference epoch time, that a frame, of which at least a part is

35 encoded into that fragment, was received by the media injector.

S4: The tagged data fragments are transmitted into the network.

Figure 11 is a flow chart illustrating the actions carried out by the P2P network interface 12 in the STB1 shown in Figure 7.

5

S5: Data fragments are received from the network.

S6: A point in time is chosen as a local reference for the internal clock, corresponding to the reference epoch time of the media injector 8. The local clock then  
10 “tracks” the media injector clock.

S7: Data fragments are re-ordered using the re-calibrated internal clock, and frames are sent to the video decoder in the right order and at the right speed.

15 It will be appreciated that variations from the above described embodiments may still fall within the scope of the claims. For example, the set top boxes have all been described as including “video decoders” but it will be appreciated that decoding for any form of media may be envisaged.

20 It will also be appreciated that the above described embodiments have been described with reference to P2P IPTV networks, but the invention may also be used in other non-linear environments, for example top down networks and store and forward networks.

25 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 media injector for injecting streaming media in the form of data fragments into a non-linear network, comprising:
  - 5 a clock arranged to determine an internal time relative to an epoch time reference point;
  - a tagging processor arranged to tag each data fragment with a timestamp, the timestamp corresponding to the internal time; and
  - a transmitter arranged to transmit the tagged data fragments into the network.
- 10 2. The media injector of claim 1, wherein the tagging processor is arranged to include the timestamp in a metadata portion of each data fragment.
3. The media injector of claim 1 or 2, wherein the media injector is arranged to  
15 receive media frames and encode said frames into the data fragments, the timestamp of each fragment identifying the internal time at which the frame encoded into that fragment was received by the media injector.
4. The media injector of claim 1, 2 or 3, further arranged to tag each data fragment  
20 with a counter to identify the order in which the fragments are encoded.
5. The media injector of any preceding claim, wherein the network is a Peer-to-Peer IP Television network.
- 25 6. The media injector of any of claims 1 to 4, wherein the network is a top down or store and forward network.
7. A node for use in a non-linear network delivering streaming media, comprising:
  - a receiver arranged to receive data fragments from the network, each data  
30 fragment tagged with a timestamp providing a time interval relative to an epoch time reference point;
  - a buffer arranged to store the received data fragments;
  - a processor arranged to determine, from the timestamps contained in the data fragments, a suitable local time reference point to correspond to the epoch time  
35 reference point;

a local clock arranged to determine a local time relative to the local time reference point; and

a traffic shape recovery unit arranged to pass media frames encoded in the data fragments to a media decoder on the basis of the timestamps of the data fragments, so that each frame is passed to the media decoder at a local time  
5 corresponding to the timestamp attached to the data fragment or fragments in which that frame is encoded.

8. The node of claim 7, wherein the order in which the frames are passed to the media decoder is different to the order in which the data fragments are received from  
10 the network.

9. The node of claim 7 or 8, wherein the rate at which the frames are passed to the media decoder is different to the rate at which the data fragments are received from  
15 the network.

10. The node of claim 7, 8 or 9, wherein the processor is arranged to determine, from the timestamps of the received data fragments, a maximum latency time interval, and the processor is further arranged to determine the local time reference point as  
20 being the time of receipt of a first data fragment plus the maximum latency time interval.

11. The node of any of claims 7 to 10, wherein the receiver is arranged to receive data fragments from more than one source.  
25

12. The node of any of claims 7 to 11, wherein the timestamp is included in a metadata portion of each data fragment.

13. The node of any of claims 7 to 12, wherein the network is a Peer-to-Peer IP  
30 Television network.

14. The node of any of claims 7 to 12, wherein the network is a top down or store and forward network.

35 15. A subsidiary media injector for receiving data fragments of a first media stream from a Peer-to-Peer network and inserting data fragments of a second media stream

into the network, the second media stream intended to complement the first media stream, the subsidiary media injector comprising:

5 a receiver arranged to receive data fragments of the first media stream from the network, each data fragment tagged with a first timestamp providing a time interval relative to an epoch time reference point;

a buffer arranged to store the received data fragments;

a processor arranged to determine, from the timestamps contained in the data fragments of the first media stream, a suitable local time reference point to correspond to the epoch time reference point;

10 a local clock arranged to determine a local time relative to the local time reference point;

a tagging processor arranged to tag each data fragment of the second media stream with a second timestamp, the second timestamp corresponding to the local time elapsed compared to the local time reference point; and

15 a transmitter arranged to transmit the tagged data fragments of the second media stream into the network.

16. The subsidiary media injector of claim 15, wherein the second media stream includes commentary.

20

17. The subsidiary media injector of claim 15, wherein the second media stream includes subtitles.

18. The subsidiary media injector of claim 15, 16 or 17, wherein the network is a  
25 Peer-to-Peer IP Television network.

19. A method of controlling delivery of streaming media in a non-linear network, comprising:

30 encoding the media into data fragments, each data fragment containing all or part of a media frame;

at a media injector connected to the network, determining an internal time relative to an epoch time reference point;

at the media injector, tagging each data fragment with a timestamp corresponding to the internal time at the moment the data is encoded;

35 transmitting the data fragments from the media injector into the network;

receiving the data fragments at a node connected to the network;

- storing the data fragments in a buffer of the node;  
at the node, determining a suitable local time reference point to correspond to the epoch time reference point;  
recovering the media frames from the data fragments; and  
5 passing the media frames to a media decoder on the basis of the timestamps of the data fragments, so that each frame is passed to the media decoder at a local time corresponding to the timestamp attached to the data fragment or fragments in which that frame is encoded.
- 10 20. The method of claim 19, wherein the timestamps are included in metadata portions of the data fragments.
21. The method of claim 19 or 20, further comprising:  
receiving the data fragments at a subsidiary media injector;  
15 storing the data fragments in a buffer of the subsidiary media injector;  
determining a time reference point local to the subsidiary media injector to correspond to the epoch time reference point;  
tagging data fragments of a second media stream with second timestamps corresponding to the time elapsed compared to the time reference point local to the  
20 subsidiary media injector; and  
transmitting the data fragments of the second media stream into the network.
22. The method of claim 19, 20 or 21, wherein the network is a Peer-to-Peer IP Television network.
- 25 23. The method of claim 19, 20 or 21, wherein the network is a top down or store and forward network.
24. An apparatus for use in a network, the apparatus comprising means for  
30 performing a method as claimed in any of claims 19 to 23.
25. A program for controlling an apparatus to perform a method as claimed in any one of claims 19 to 23.
26. A program which, when loaded into an apparatus, causes the apparatus to become an apparatus as claimed in claim 24.

27. A program as claimed in claim 25 or 26, carried on a carrier medium.

28. A program as claimed in claim 27, wherein the carrier medium is a storage medium.

29. A program as claimed in claim 27, wherein the carrier medium is a transmission  
5 medium.

30. An apparatus programmed by a program as claimed in any one of claims 25 to  
29.

31. A storage medium containing a program as claimed in any one of claims 25 to  
28.

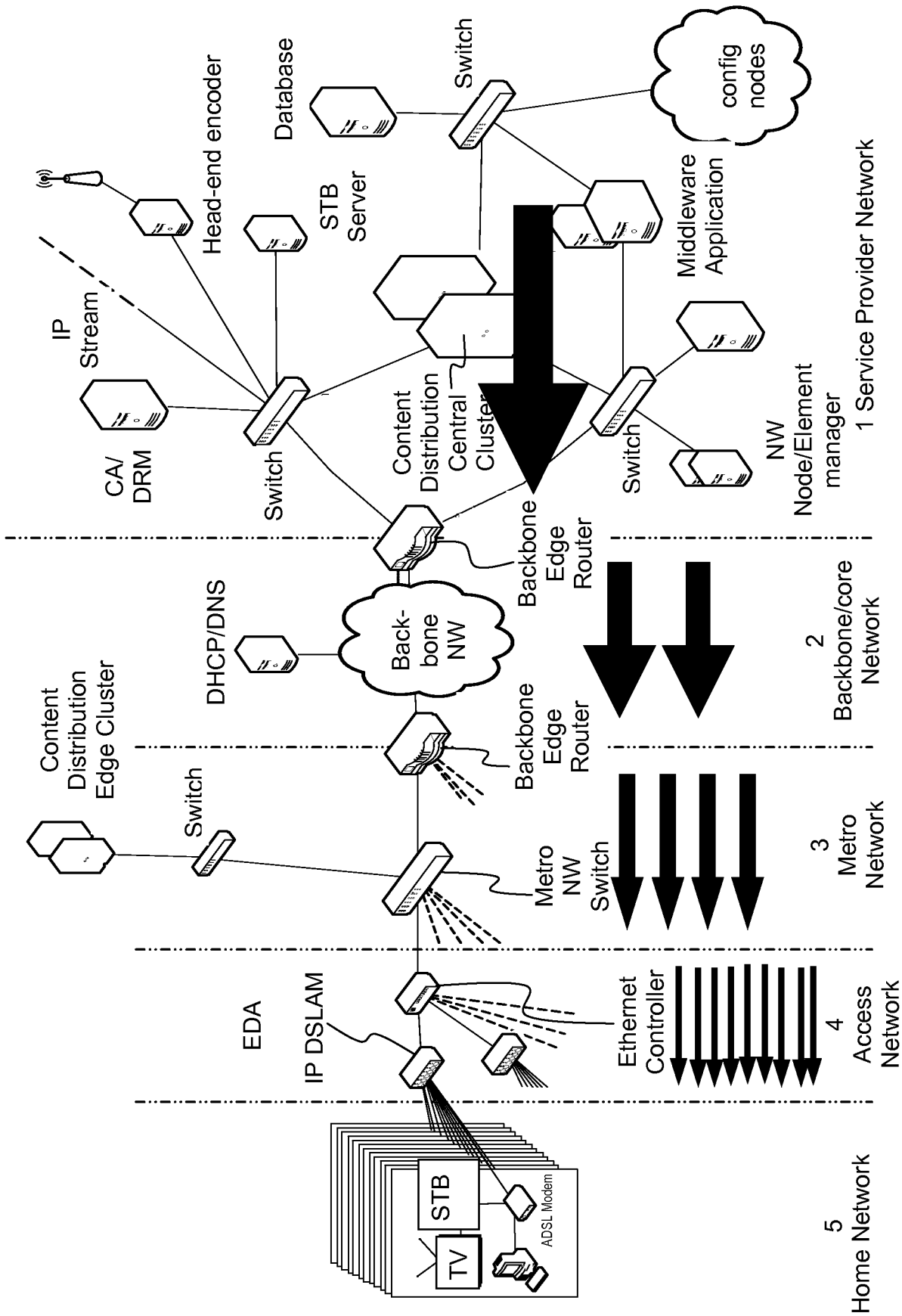


Figure 1

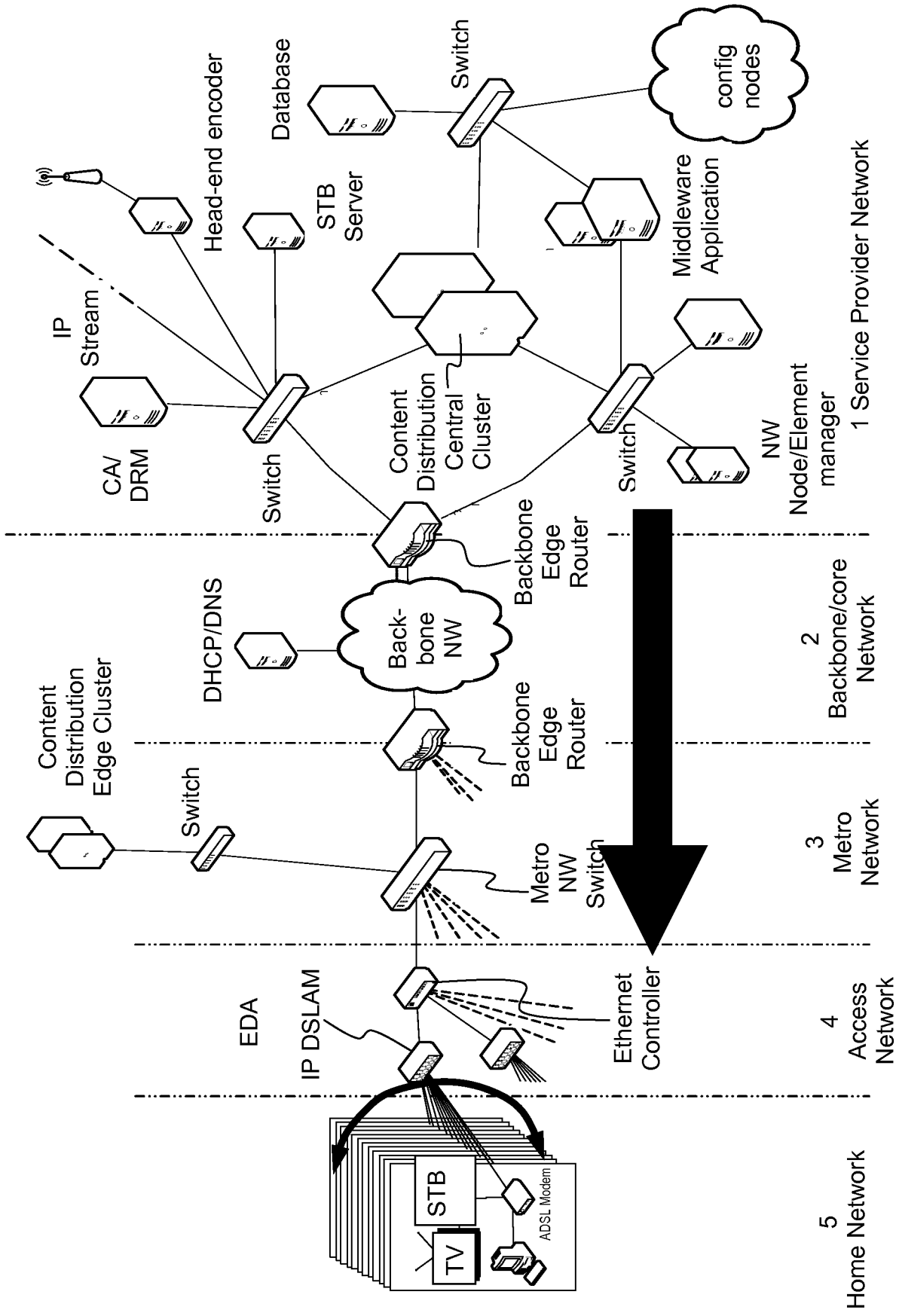


Figure 2

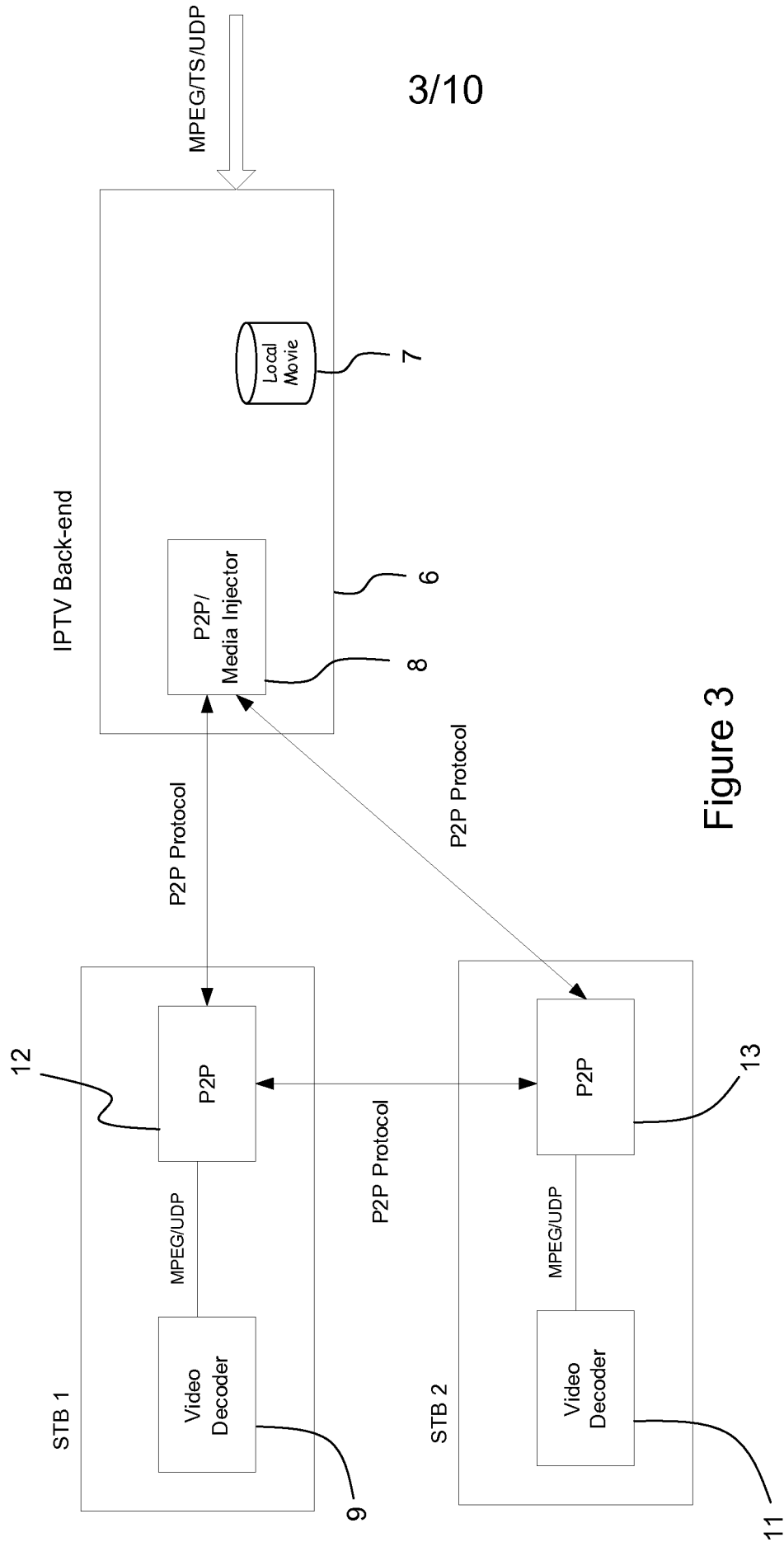


Figure 3

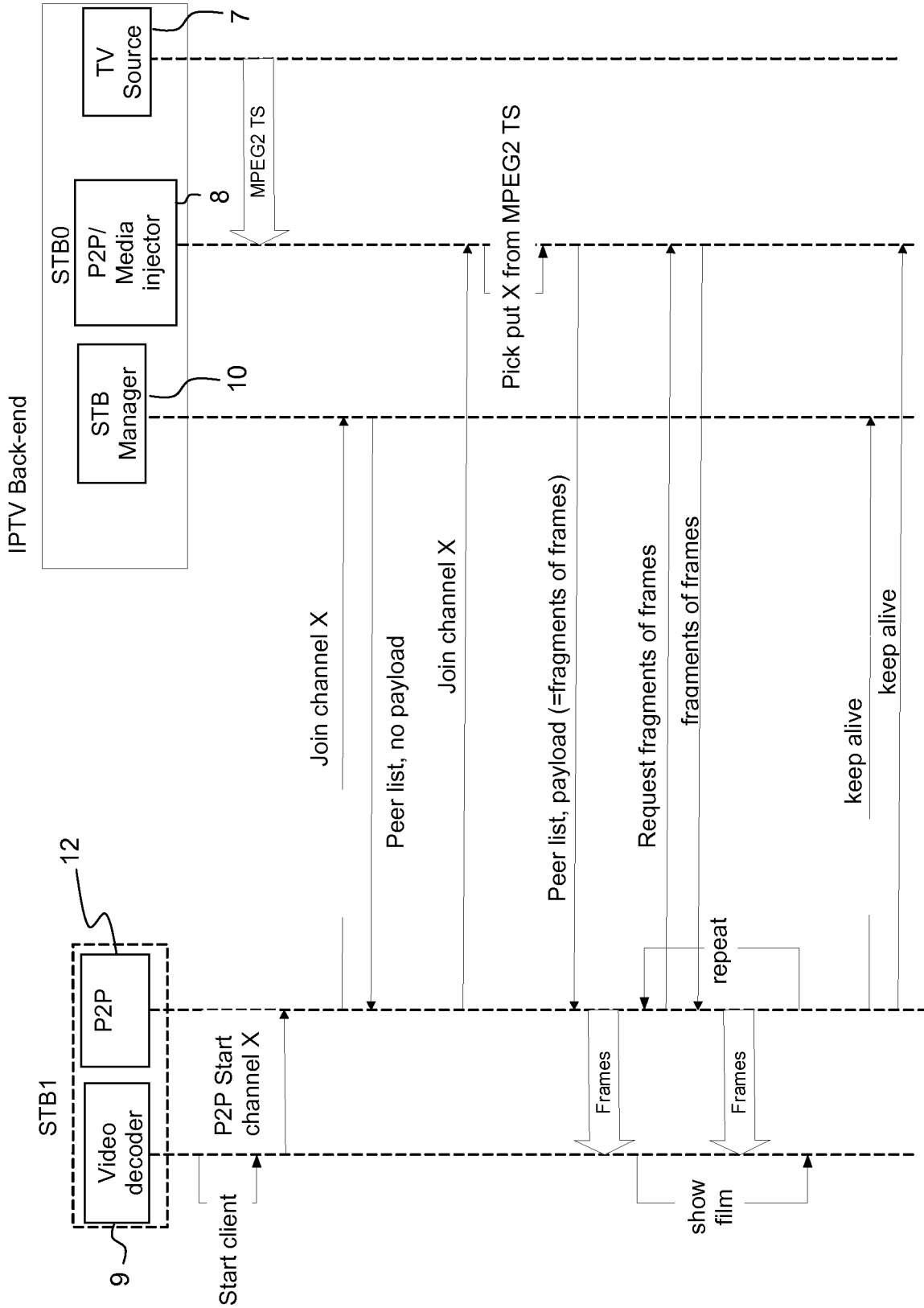


Figure 4

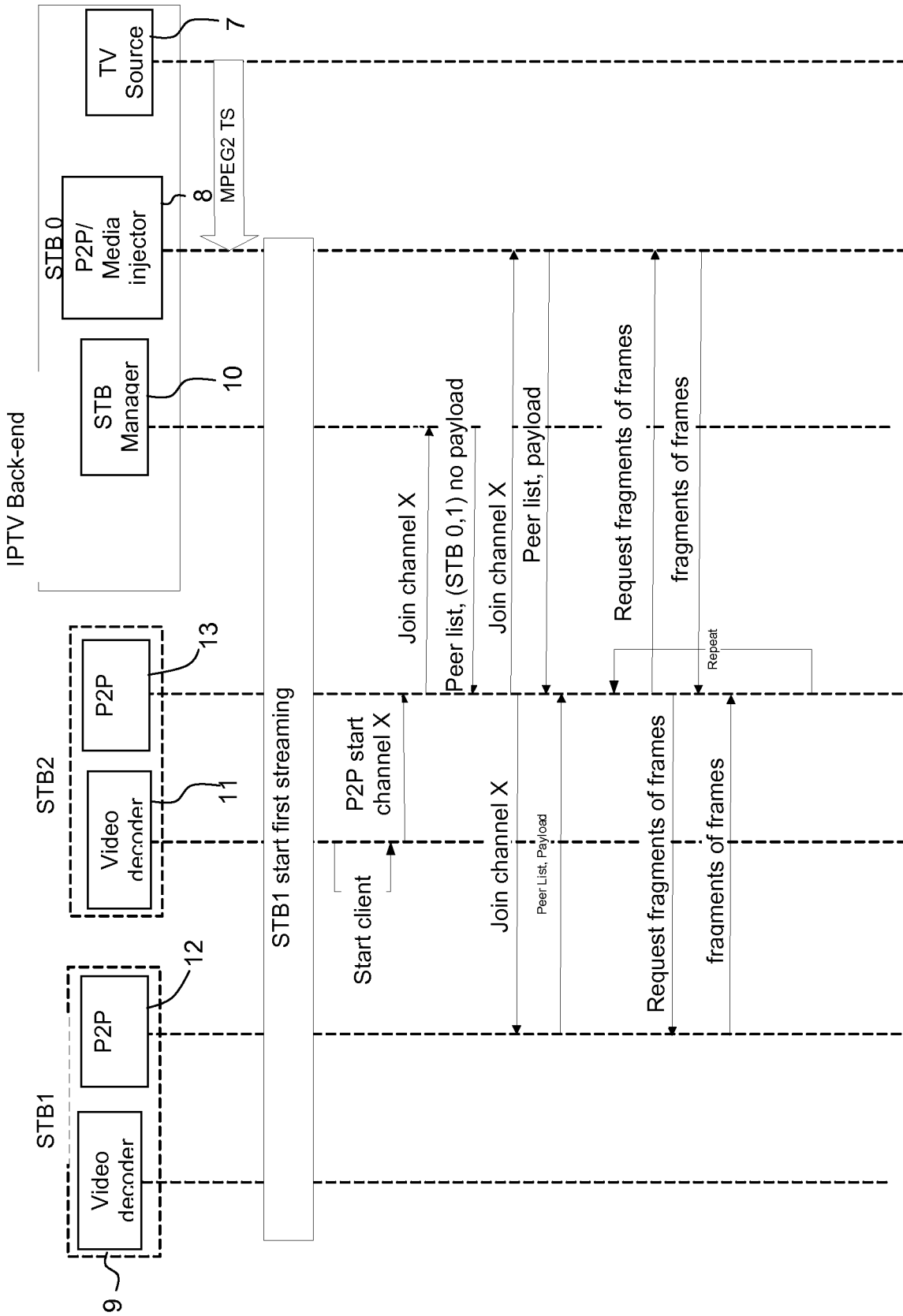


Figure 5

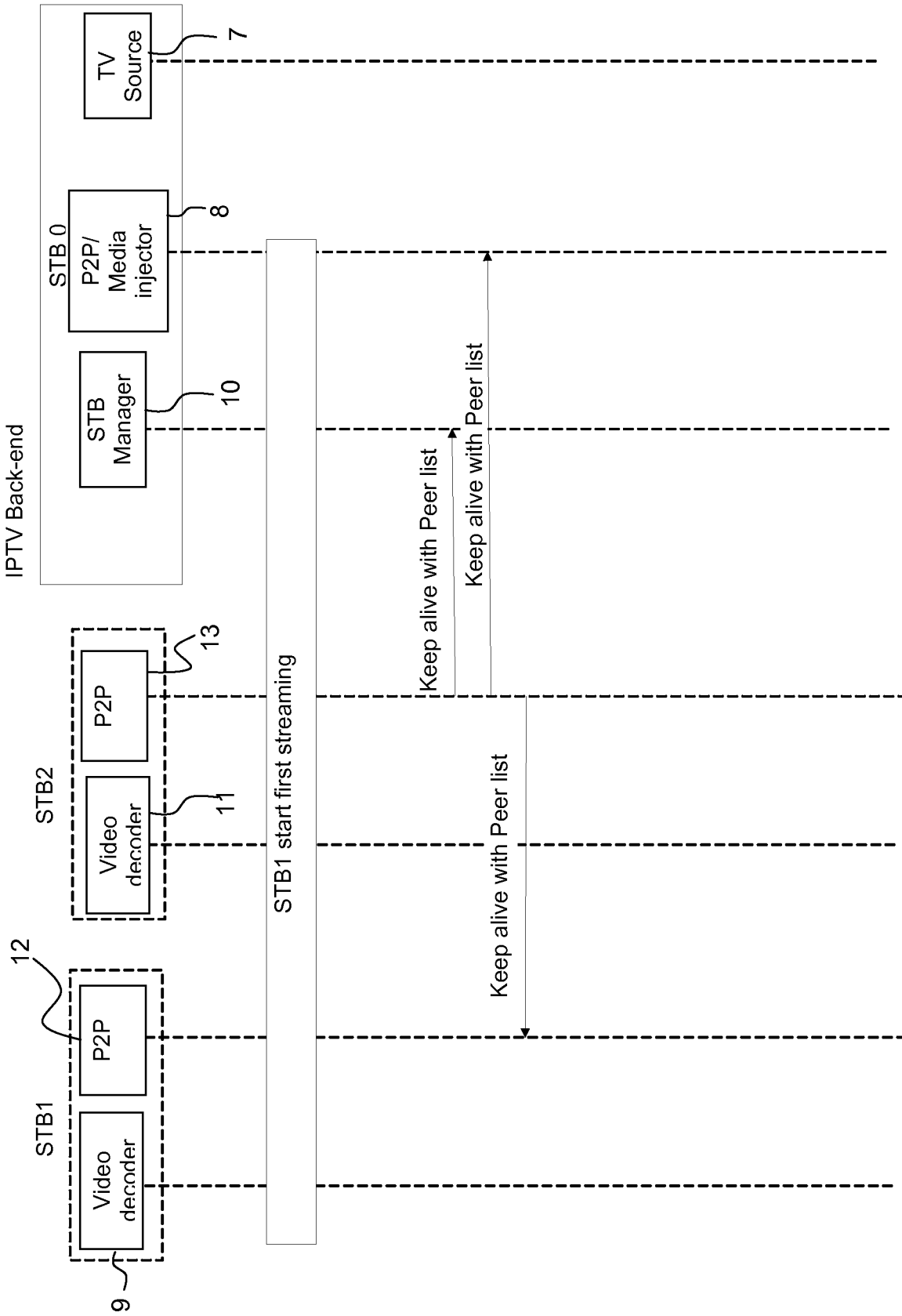


Figure 6

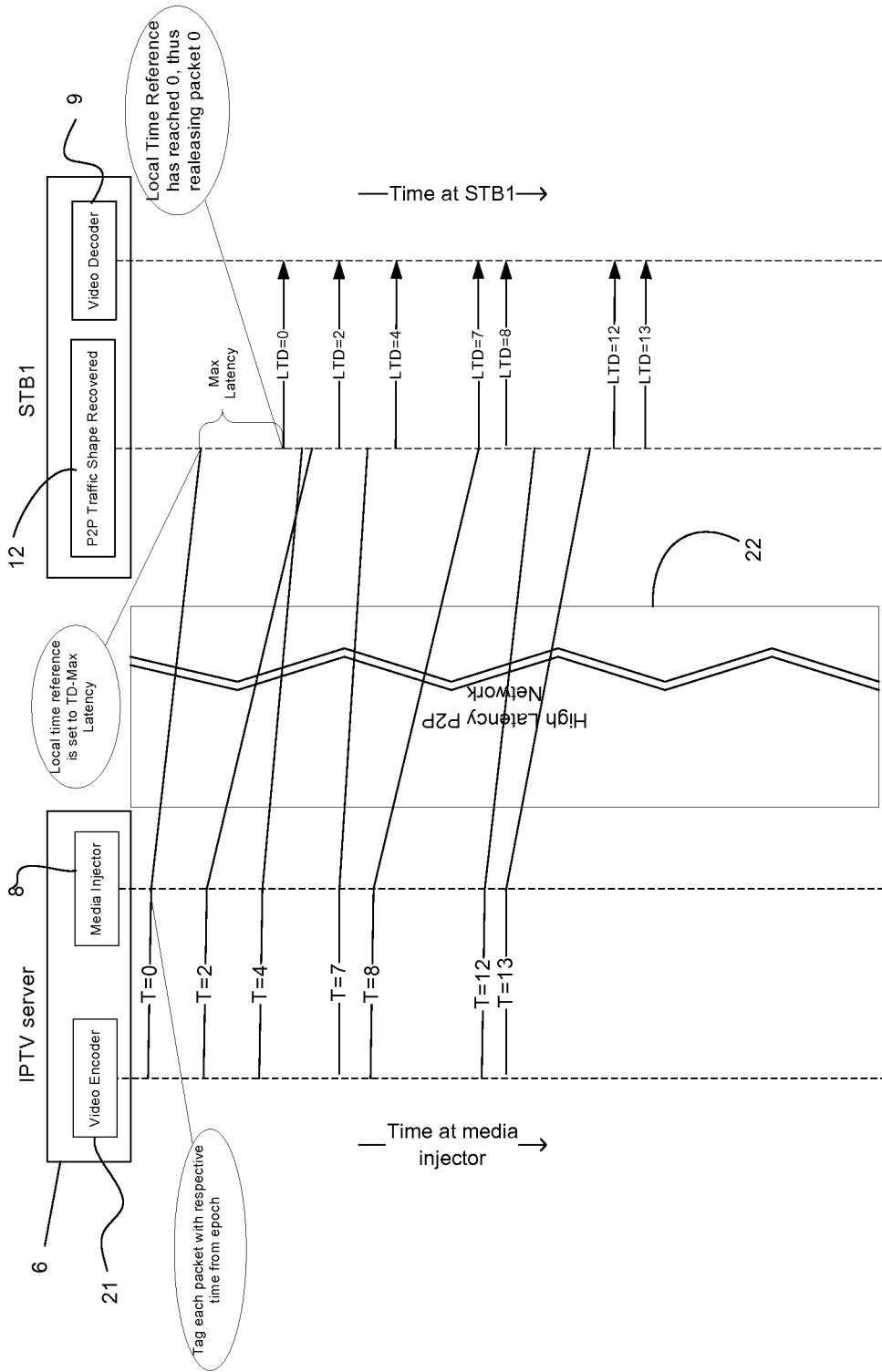


Figure 7

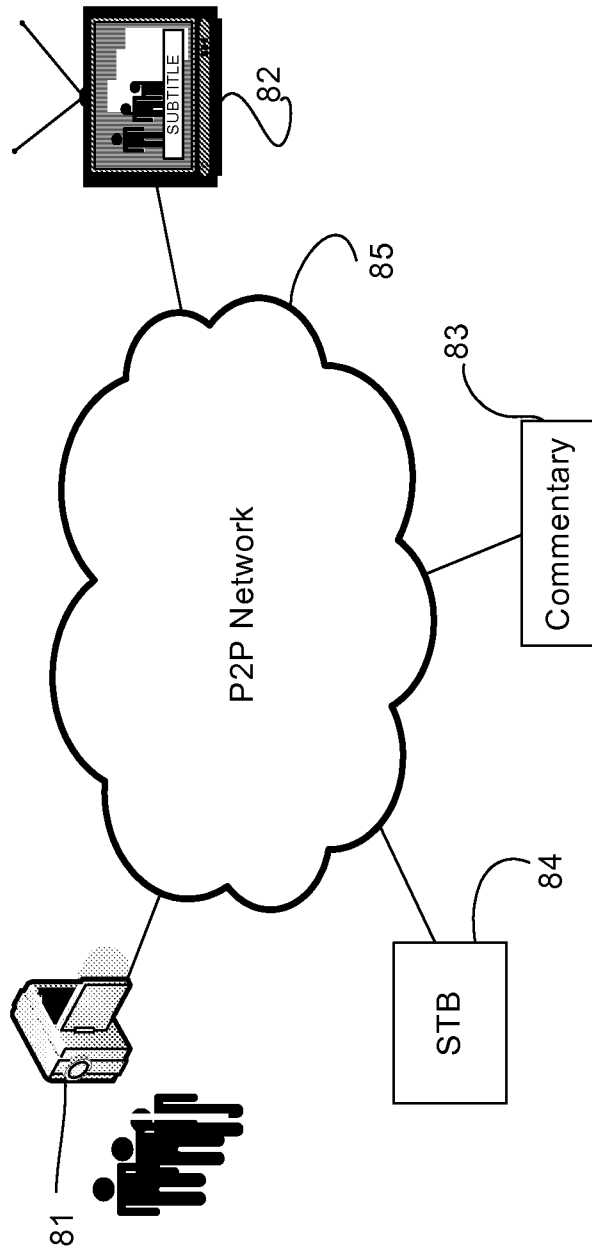


Figure 8

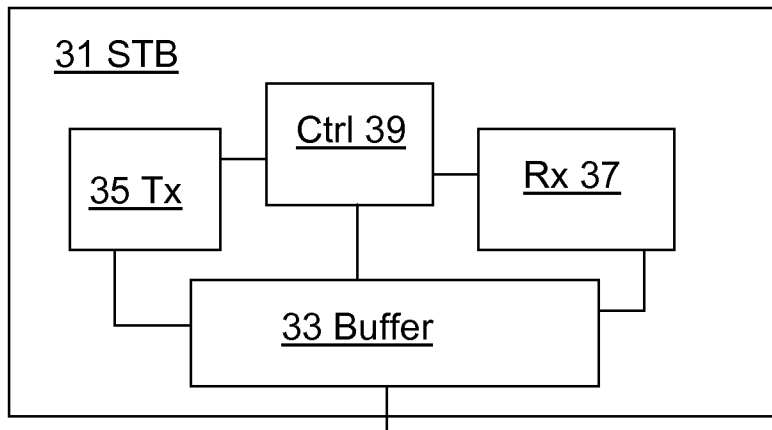


Figure 9

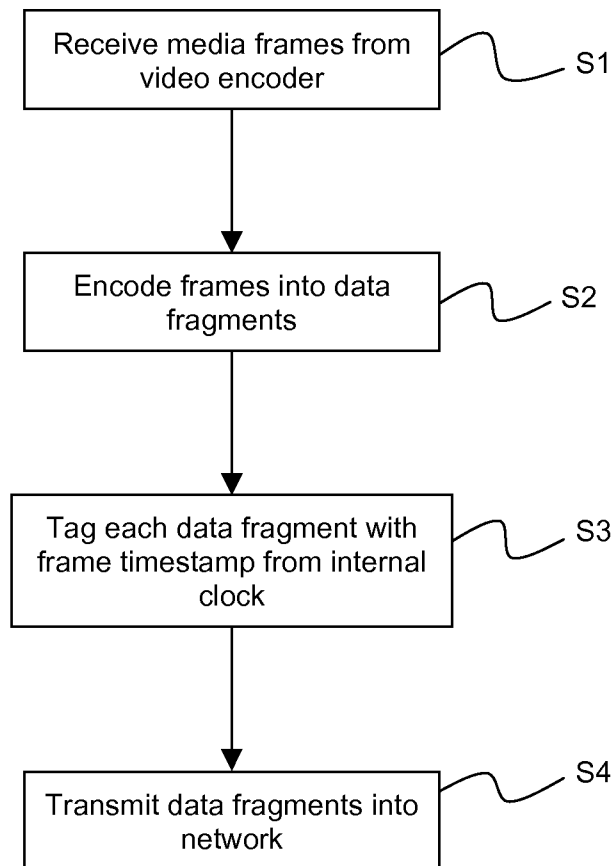


Figure 10

10/10

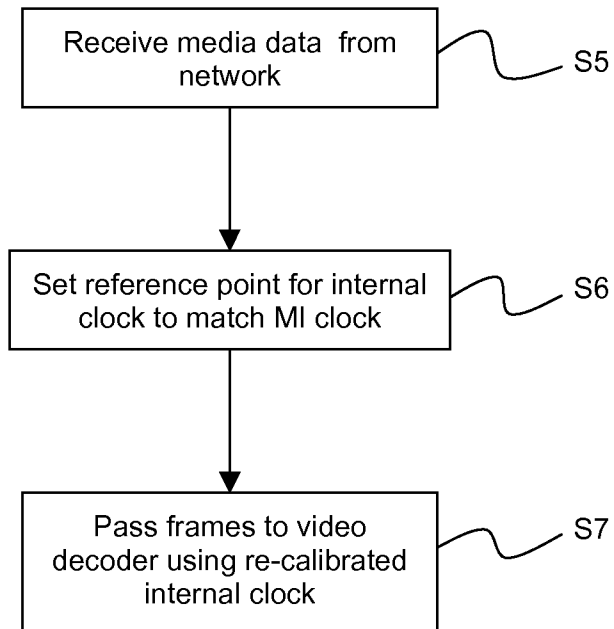


Figure 11

**INTERNATIONAL SEARCH REPORT**

International application No  
PCT/EP2007/064451

**A. CLASSIFICATION OF SUBJECT MATTER**  
INV. 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)  
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	AUDIO-VIDEO TRANSPORT WORKING GROUP H SCHULZRINNE GMD FOKUS S CASNER PRECEPT SOFTWARE ET AL: "RTP: A Transport Protocol for Real-Time Applications; rfc1889.txt" IETF STANDARD, INTERNET ENGINEERING TASK FORCE, IETF, CH, 1 January 1996 (1996-01-01), XP015007673 ISSN: 0000-0003 paragraphs [0001], [02.1], [02.2], [02.3], [0003], [05.1], [6.3.1], [6.3.4], [07.1]	1-31
X	TANENBAUM: "Computer networks" 31 August 2002 (2002-08-31), PRENTICE HALL, XP002492564 paragraph [6.4.3]	1-31

Further documents are listed in the continuation of Box C.

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.
- \*Z\* document member of the same patent family

Date of the actual completion of the international search

19 August 2008

Date of mailing of the international search report

29/08/2008

Name and mailing address of the ISA/

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

Authorized officer

Goller, Wolfgang

## INTERNATIONAL SEARCH REPORT

International application No  
PCT/EP2007/064451

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	<p>REHA CIVANLAR M: "Protocols for real-time multimedia data transmission over the Internet" ACOUSTICS, SPEECH AND SIGNAL PROCESSING, 1998. PROCEEDINGS OF THE 1998 IEEE INTERNATIONAL CONFERENCE ON SEATTLE, WA, USA 12-15 MAY 1998, NEW YORK, NY, USA, IEEE, US, vol. 6, 12 May 1998 (1998-05-12); pages 3809-3812, XP010279676 ISBN: 978-0-7803-4428-0 paragraphs [04.1], [04.2] -----</p>	1-31
A	<p>MONTGOMERY W A: "TECHNIQUES FOR PACKET VOICE SYNCHRONIZATION" IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, IEEE SERVICE CENTER, PISCATAWAY, US, vol. SAC-01, no. 6, 1 December 1983 (1983-12-01), pages 1022-1028, XP000563228 ISSN: 0733-8716 paragraphs [0001], [0003] -----</p>	10