

Method to stream contents in a peer-to-peer network

GENERAL TECHNICAL FIELD

This invention relates to a method to stream content in a peer-to-peer network.

STATE OF THE ART

5 Streaming refers to a technique for playing an audio or video stream "live", i.e., as it is retrieved from the Internet by a client device. It is thus opposed to downloading which requires retrieving all the data from the audio or video content before it can be played.

In the case of the streaming, content storage is temporary and partial, with data being continuously downloaded into a client buffer (typically the RAM), decoded on the fly,
10 sent to an output interface (a screen and/or speakers) and then replaced with new data.

Traditionally, the content is made available on a streaming server. The client wishing to access it sends a request to retrieve the first segments (the term, segment is understood to mean a block of content data, generally corresponding to a few seconds of playback). When there is enough data in the buffer to read the beginning of the content,
15 the playback starts. In the background, the stream continues to download in order to continuously feed the buffer with the rest of the content.

However, this approach has its limits if a large number of clients wish to read the same content simultaneously: the server becomes saturated, unable to provide the content at a rate sufficient for smooth playback, and jerking occurs.

20 Recently, an alternative strategy based on the peer-to-peer (P2P) has been proposed, in which each client acts as a server for other clients: these are called peers. A peer that has started reading the content can retransmit the segments it has already received to others, and so on, making it easy to distribute regardless of the number of interested clients. For example, this strategy is described in international application WO
25 2012/154287 and brings satisfaction.

However, most players implement a so-called Adaptive BitRate (ABR) logic and this has proved to be problematic when combined with P2P.

The general idea of the ABR is to allow the automatic variation of the quality of the retrieved segments according to the "capacities" of a peer. More precisely, each segment
30 is available at several quality levels corresponding to several bitrates, i.e., data rates. This is because it is understood that a segment of better quality has better resolution, less compression, more frames per second, etc., and is therefore larger than the same segment of lower quality, and it is therefore necessary to maintain a higher data rate.

The ABR attempts to automatically determine, for each segment, the best quality
35 that can be chosen, usually based on two criteria, which are the observed bandwidth

and/or the buffer fill rate.

In the first case, if the algorithm judges that the estimated bandwidth is sufficient to support a higher quality, then it will instruct the client to switch to this quality (or conversely to lower the quality if the bandwidth is too low). In the second case, the principle is to divide the buffer into different intervals, each interval corresponding to an increasingly higher quality as the filling of the buffer increases (increasingly lower if it decreases).

In both cases, while ABR algorithms have no fundamental incompatibility to be used in a P2P streaming context, the problem is that the ABR algorithms were designed to work in a simple streaming scenario, i.e., with all the segments retrieved on request from the content server.

However, in practice, P2P streaming performs "pre-buffering", by downloading P2P segments into a dedicated P2P cache before the player actually makes the request. Indeed, the objective of the P2P streaming is to request as little as possible (and as a last resort) the original content server: a direct request for a segment to this server is only made if there is a risk that there are no more segments in the video buffer and if the playback will be interrupted ("re-buffering"), otherwise there is a maximum reliance upon the P2P network.

This results in extremely high apparent bandwidths from the player's point of view, since segments can be loaded into the buffer from the P2P cache a fraction of a second after they have been requested. In addition, the video buffer fill rate is artificially high.

This leads to uncontrolled decisions by the ABR to increase quality if the current quality is not the maximum quality, regardless of the actual capacity of the network, quality that it may not necessarily be able to support.

At best, this results in unstable oscillations in the quality of the stream. At worst, it results in repeated interruptions of the playback and numerous and unnecessary requests from the content server.

It would be desirable to be able to control the ABR logic, but unfortunately this is often player-specific and not very accessible, let alone modifiable. So it would be desirable to have a way to control any ABR algorithm in a P2P streaming context.

This invention will improve the situation.

PRESENTATION OF THE INVENTION

This invention thus relates to a method of streaming content, broadcast within a peer-to-peer network of client equipment to a player of a client device, said content consisting of a sequence of segments available in a plurality of quality levels and said player being adapted to choose the quality level of the segments in accordance with an

adaptive bitrate regulation logic, ABR; the client device comprising a first buffer adapted to store segments in a format suitable for transfer within the peer-to-peer network, the method being characterized in that it comprises the implementation by data processing means of the device, of steps of:

- 5 (a) Receiving a request for a segment from the player,
- (b) Response delivery of said segment from the first buffer upon expiry of a response time defined with respect to said player ABR logic.

This method cleverly provides feedback information to the ABR algorithm by adding an artificial delay to the response.

10 According to advantageous and non-limiting characteristics:

- said ABR logic is defined by a first function making it possible to calculate a quality level in which said segment is required as a function of at least one parameter representative of a segment reception rate;
- a second function in step (a) makes it possible to calculate said response
15 time as a function of said at least one parameter representative of a segment reception rate;
- the second function is constructed from the first function;
- the second function is monotonic with respect to said at least one parameter representative of a segment reception rate;
- 20 • the device further comprises a second buffer adapted to store segments in a format suitable for playback by the player, said segment being provided in step (b) in the second buffer;
- said parameter representative of a segment reception rate is a fill level of the second buffer and/or a bandwidth;
- 25 • step (a) comprises predicting a quality level in which the next segment will be required as a function of the calculated response time;
- the method comprises a step (c) to verify said quality level prediction in which the next segment will be required;
- the method comprises learning the second function based on the predicted
30 and actually required quality levels;
- the availability of the required segment is incomplete in the first buffer. Step (a) comprises the modification of the calculated response time as a function of an estimated time necessary to finish retrieving the segment, the segment being provided in complete manner in step (b) at the earliest upon expiry of the modified response time.

According to a second aspect, the invention proposes a device for streaming, on a player, content broadcast within a peer-to-peer network of client devices, said content consisting of a sequence of segments available in a plurality of quality levels and said player being adapted to choose the quality level of the segments in accordance with an adaptive bitrate regulation logic, ABR; the client device comprising a first buffer adapted to store segments in a format suitable for transfer within the peer-to-peer network and data processing means, characterized in that the data processing means are configured, following the reception of a request for a segment from the player, to provide, in response, said segment from the first buffer upon expiry of a response time defined with respect to said player's ABR logic.

According to a third and a fourth aspect, the invention proposes a computer program product comprising code instructions to execute a method, according to the first aspect of streaming content in a peer-to-peer network of client devices connected to a content server; and computer equipment readable storage means upon which a computer program product comprises code instructions to execute a method according to the first aspect, for streaming content in a peer-to-peer network of client devices connected to a content server.

PRESENTATION OF THE FIGURES

Other characteristics and advantages of this invention will appear upon reading the following description of a preferred embodiment. This description will be given with reference to the appended figures wherein:

Figure 1 represents an architecture to implement the method according to the invention;

Figure 2 illustrates a preferred embodiment of the method according to the invention.

DETAILED DESCRIPTION

Architecture

Referring to Figure 1, the invention proposes a method to stream content broadcast within a network 1, as shown. Here, network 1 is a large-scale telecommunications network and in particular, the Internet. This network 1 comprises a peer-to-peer network 10 of client devices 11,12. Each client device 11,12 is typically a personal computer device such as a PC, a smartphone, a tablet, etc. connected to the network 1, having data processing means 110 such as a processor, an interface for the playback of the content, and data storage means such as a random access memory and/or a mass memory.

The playback is implemented by a player, i.e., an application executed by the data processing means 110, which can be of a varied nature, for example a dedicated application, an internet browser particularly compatible with HTML5, a module of the operating system, etc.

5 In the following description, it will be assumed that the player is "as is", i.e., it is not modified for the implementation of this method, or even for P2P streaming. In particular, the player implements adaptive bitrate regulation logic, ABR, i.e., said content to be played consists of a sequence of segments available in a plurality of qualities and the player is able to decide autonomously which quality to require, in accordance with this
10 ABR logic. The various qualities correspond to different bitrates, i.e., a variable volume of data per time unit (and thus per segment). Naturally, it is understandable that better quality content requires a higher bitrate.

More will be said in more detail about the concept of ABR logic below, it will only be understood that in the context of This method it is not necessary for the ABR logic to be
15 controlled or even known: This method is completely universal and can be adapted to any player implementing any ABR logic on the basis of any criteria. It will be assumed that the ABR logic is predefined and that the client software (see below) only undergoes it. Furthermore, the device 11 (and more precisely its data storage means) has two buffers M1 and M2 (also called "buffer"), typically two areas of a random access memory, each of
20 which can store (in a different way as will be seen) all or part of the content temporarily (the term, temporary is understood to mean that the segments are deleted from this memory shortly after they have been played: they are not stored in the long term as it is the case for a direct download). For example, in the case of playback via a browser, all the segments are typically deleted (i.e., the buffers are reset) when the browser or the tab on
25 which the video is played is closed at the latest.

The first buffer M1 is called "peer-to-peer cache". It stores the segments in a format called "raw" format. The term, raw segments is understood to mean in a format suitable for transfer within the peer- to-peer network 10, but unsuitable for playback on the device
11.

30 The second buffer M2 is called "video buffer". It is controlled by the player and stores the segments in a format called "converted" format. The term, converted segments is understood to mean converted from raw segments in a format suitable for playback (and particularly for the player) on the device 11, but unsuitable for transfer within the peer-to-peer network 10, see below.

35 As explained in the introduction, the devices 11,12 are "peers" (also called

"nodes") of the peer-to-peer network 10.

"Client devices 11, 12 of a peer-to-peer network 10" is understood to mean devices connected in the network 1 by a peer-to-peer network protocol. In other words, the data processing means of each peer implement a particular program (client software, called "Peer-Agent", PA), which can be for example integrated in the player (for example an extension of a web browser), be the subject of a dedicated application, or even be incorporated into any other on-board software (for example a player from an Internet access box or a multimedia box, i.e., a "set-top box"), for peer-to-peer use. This method is mainly implemented via this client software. In the remainder of this description, it will be assumed that the client software communicates with the player in order to provide it with segments, but that it operates independently. More precisely, it is understood that the role of the player is the playback itself, i.e., the rendering of the segments, while the role of the client software is to simply obtain segments for the player, and the client software operates the player and in particular its ABR logic.

As explained, a peer-to-peer or P2P network is a decentralized sub-network within the network 1, in which data can be transferred directly between two client devices 11, 12 of the network 10, without passing through a central server. It thus allows all client devices 11, 12 to act both as client and server. The peers 11, 12 are thus defined as "seeders" (i.e., data providers) and/or "leechers" (i.e., data recipients).

Said content, which is specifically audio or video content, i.e., a medium of certain duration, consists of a sequence of segments (called playlist) stored on a means for storing the data on a server 2 connected to the peer-to-peer network 10, and as explained available in several quality levels. The segments have a predetermined length, typically one or two seconds of the content, but it can range from a fraction of a second to about ten seconds. All the segments of a given content have generally the same length, which will be noted here ρ .

Server 2 is a content server (called CDN), advantageously present in network 1 and connected to the peer-to-peer network 10. In other words, this is one (or more) server(s) of internet network 1 that provides the segments of various contents in accordance with a given streaming protocol. HLS (HTTP Live Streaming) can be mentioned as an example, in which the segments are "ts" files, listed in a "m3u8" playlist file. HLS implies the MPEG2 format for the content, the DASH, Smooth streaming or HDS streaming protocols may also be mentioned. Raw segments can be exchanged between peers via a WebRTC-type protocol.

Server 2 is the primary source of the segments, insofar as initially no peer has

content (before a first transfer from server 2 to this peer 11, 12). Content is either from the outset stored in their entirety on the server 2 (as in the case of the VOD previously mentioned), or generated in real time (as in the case of the live streaming), and in the latter case the list of segments which constitute them evolves dynamically.

5 Live Streaming proposes broadcasting content associated with "live" events in real time, for example concerts, conferences, sports events, video game plays, etc., which are taking place simultaneously. Compared to streaming of an already fully existing content such as a movie, a broadcast content in live streaming is effectively generated as the associated event takes place. Technically, as it is the case of a live broadcast on
10 television, such content can only be broadcasted with a slight delay. The user will want this to be as low as possible. This delay is typically on the order of one minute, but can go down to around twenty seconds. This makes a playlist of only a handful of segments (at most a few dozen) available at any time, the segments of this list being renewed dynamically according to a rotation: as the event unfolds, new segments are created,
15 "aged", are received and played by the clients (at the end of the expected delay), and they finally leave the list.

In the latter case (live streaming), the content should preferably be viewed as a continuous stream. The sequence of segments is thus dynamic, i.e., it is updated on a regular basis. Each time a new segment is generated, it is added at the end of the
20 sequence, and the first segment of the sequence (the oldest one) is deleted. All the others shift according to a shift mechanism that can be compared to a FIFO list. The first segment of the list (the oldest one) can be the one at the play point, in other words the "live" segment (and therefore the segments are removed from the playlist as soon as they are played), or a "past" segment if the content server accepts that the content is played
25 with a delay (some platforms propose live streaming with a delay of up to 2 hours), this is called DVR (Digital video recorder).

This method can be implemented in any context.

A peer management server 3 called "tracker" is also advantageously connected to the peer-to-peer network 10. The tracker 3 has data processing means and storage
30 means. It coordinates the exchanges between peers 11, 12 (by controlling the client software implemented by each of the client devices 11, 12), but it is not directly involved in the data transfer and does not have a copy of the file.

ABR

As mentioned, there are two main types of ABR logics, based respectively on the
35 bandwidth (i.e., the observed data rate) and on the fill level (either in value, i.e., in seconds

or in number of segments, or in rate) of the second buffer M2 (possibly in combination). In other words, the player monitors the bandwidth and/or the fill rate/volume of the second buffer M2, and consequently makes decisions as to whether the quality level of the required segments should be modified or not.

5 Generally, in either case, the ABR logic can be defined by means of a first function making it possible to calculate the quality level to be chosen (the bitrate) as a function of at least one parameter representative of a segment reception rate, see for example the curve as represented in Figure 2 (left part, which represents the player side, the right part illustrating the client software side).

10 It is understood that said parameter representative of a segment reception rate is a monitored parameter, which can be any parameter illustrating the ability of the device 11 and/or of the network 10 to receive the segments "fast enough". As explained previously, known ABR logics generally use a fill level as a parameter of the second buffer M2 and/or a bandwidth.

15 Regardless of the type of ABR logic, said first function is monotonic, principally increasing, between a minimum quality value and a maximum quality value. A preferred model is that of a continuous piecewise function, with three pieces (connected at two singular points), including a first and a third constant piece.

20 Thus, in the example of Figure 2 (left part), the example of an ABR logic based on the fill level of the second buffer M2 (expressed in seconds) is represented, and we see that:

- for a fill level lower than or equal to a minimum threshold $B_{\min}$, the bitrate is set equal to the minimum bitrate $R_{\min}$ (in other words the buffer is dangerously empty, which is representative of difficulties in obtaining segments, and then the quality is minimal to promote re-filling);
- for a fill level higher than or equal to a maximum threshold $B_{\max}$, the bitrate is set equal to the maximum bitrate $R_{\max}$ (the buffer is full therefore there is no risk of affording the maximum quality);
- between the two, there is an increasing, preferably linear, curve - in the example of Figure 2 it has the equation $R=R_{\min}+(B-B_{\min})*(R_{\max}-R_{\min})/(B_{\max}-B_{\min})$ - which means that the fuller the buffer, the higher the quality.

A similar curve is obtained for an ABR logic based on the bandwidth.

35 As will be seen below, this method proposes to control the ABR not by altering its logic, but by deceiving it by means of an added response time before satisfying a request for the segment. Indeed, simply delaying the provision of a segment makes it possible to

artificially limit the bandwidth and the fill level of the second buffer M2, and thus to act on the expected quality of the segments.

For this a second function is defined, allowing said response time as a function of said at least one parameter representative of a segment reception rate in the represented examples, this is therefore again the fill level of the second buffer M2).

The second function is preferably constructed (or even learned, see below) from the first function, i.e., the ABR logic can be mapped by means of the second function. Thus, according to a preferred embodiment, the second function is monotonic compared to said at least one parameter representative of a segment reception rate, and generally (in particular in the case of an ABR logic based on the fill level of the second buffer M2 which will be developed below) the first and second functions have the same monotony with respect to said at least one parameter representative of a segment reception rate (in particular, an increasing function). Indeed the second function is most often similar to the first function of the ABR logic (again a continuous piecewise function with three pieces including two extreme segments), see for example the corresponding curve which is represented on the right side in Figure 2.

For example, mapping can consist of taking two points from the first function, in particular the singular points connecting the pieces of coordinates $\{B_{\min}, R_{\min}=f_1(B_{\min})\}$ and $\{B_{\max}, R_{\max}=f_1(B_{\max})\}$, and constructing the second function on the basis of the two corresponding points $\{B_{\min}, \rho_{\min}=f_2(B_{\min})\}$ and $\{B_{\max}, \rho_{\max}=f_2(B_{\max})\}$. Preferably, the second function is constructed with the same monotony as the first function, particularly a continuous piecewise function with three pieces connected via said two points.

In the example of Figure 2 on the right, the second function is thus constructed as follows:

- for a fill level lower than or equal to the minimum threshold $B_{\min}$, the response time is set to a minimum value $\rho_{\min}$, preferably zero, i.e., no time is added (in other words the buffer is dangerously empty, and there is no risk in directly providing the received segments);
- for a fill level higher than or equal to a maximum threshold $B_{\max}$, the response time is set equal to a maximum value $\rho_{\max}$, preferably the length p of a segment (this makes it possible not to further increase the fill rate of the video buffer, the P2P cache keeps margin by providing a new segment only when a previous segment has been fully played);
- between the two, there is an increasing, preferably linear, curve
- in the example of Figure 2, it has the equation $\rho=\rho_{\min}+ (B-B_{\min}) \cdot \frac{\rho_{\max}-\rho_{\min}}{B_{\max}-B_{\min}}$

$\min) * (\rho_{\max} - \rho_{\min}) / (B_{\max} - B_{\min})$ - which means that the fuller the buffer, the longer the response time, so as to prevent a further increase in this fill rate.

Note that it is also possible that the first and second functions have reverse monotony compared to said at least one parameter representative of a segment reception rate (i.e., the second function is increasing when the first function is decreasing, and vice-versa), this is especially the case if the ABR logic is based on the bandwidth (then the first function increases and the second function decreases).

This method will not be limited to a given response time application strategy, and those skilled in the art will be able to adapt the idea to the ABR logic, to the type of parameter monitored by the player, to the behavior of the P2P network, etc. In general, it will be understood that it suffices to construct a second function defining the response time from the first function defining the ABR logic.

Playback

In the remainder of this description, the focus will be on a client device that may be in the process of retrieving the content from other devices and/or the server 2, i.e., the first buffer M1 has already stored at least one raw segment, in at least one quality level, if possible a sub-sequence of the sequence constituting the content.

The method then begins with the implementation to receive a request for a segment from the equipment of a step (a), by the processing means 110. In this case, the request is for the next segment to be put into the second buffer M2 (not necessarily the next segment to play, there are normally buffered advance segments). Said request is received from the player, and preferably defines the quality level in which the segment is required, i.e., the bitrate (by application of the ABR logic).

It is assumed that said segment is at this stage at least partially available (i.e., at least one fragment) in the first buffer M1, in the quality required by the player. If this segment/segment fragment is of another quality, it should be retrieved again, generally directly from the content server 2 because there is not enough time.

As explained, step (a) then advantageously comprises the calculation of a response time defined relative to said ABR logic of the player, in particular as a function of a parameter monitored by the ABR logic (said at least one parameter representative of a segment reception rate), preferably the fill level of the second buffer M2 and/or the bandwidth. Step (a) comprises, if necessary, the "measurement" of said parameter representative of a segment reception rate.

As explained, the second function defining the response time is preferably increasing with said parameter representative of a segment reception rate between a

minimum response time value and a maximum value.

In the case where it is only a fragment of the next segment that has been retrieved (the segment is said to be available in incomplete form), preferably the calculated response time is modified according to the length of the fragment so as to reflect the fact that only one fragment of the response time should be actually applied. Indeed, the second buffer M2 can only be supplied with complete segments and not fragments, and the idea is to provide the segment in a complete way at the end of a shortened response time reflecting the fact that there will already be a waiting period implicitly corresponding to the time for completing (finish retrieving) this segment in the first buffer M1. Thus, step (a) includes modifying the calculated response time based on the estimated time necessary to finish retrieving the segment.

For example, the formula $\rho' = \rho - \text{tdw}$ could be applied where ρ' is the modified response time and tdw is the estimated duration needed to finish retrieving the segment. Thus, waiting for the time tdw plus application of ρ' before providing the complete segment is equivalent to the application ρ , so that the overall time remains the same.

In a step (b), said required segment is provided in response to the request, from the first buffer M1, upon expiry of said response time. "Provided upon expiry of said response time" is understood to mean that the player does not have it before the expiry of the response time (at best at the time of expiry, or even only after in some cases, see below). Most often, the segment is transmitted at once upon expiry of the response time, but it will be understood that it is quite possible to stream it within the device 1, i.e., to transmit it from the first buffer M1 gradually (bit by bit) so that the last bit is transmitted (at the earliest) upon expiry of the response time (the response time is then a "transmission time of the last bit of the segment"). Indeed, although only complete segments are playable, some players can accept sub-segments of the segment. Note that such a progressive transmission does not change anything since, as long as the segment is not fully received, it is not available by the player and therefore not considered to be provided but makes it possible to facilitate bandwidth measurements.

In the case where only a fragment of the segment was available in the first buffer M1 and where the response time has been modified according to an estimated duration necessary to finish retrieving the segment, normally the segment is also provided in step (b) upon expiry of the modified response time. As explained, although the provision can be fragmented, sub-segments of a complete segment (which correspond to consecutive pieces of segment obtained from a completely downloaded segment) should not be confused with an incomplete segment (in which only some data portions, most often

corresponding to disparate pieces, have been downloaded). Only a segment available in complete form in the first buffer M1 can be provided (if necessary gradually) in response to the request (and not a fragment), so that if the download took longer than expected, the segment may be available in complete form only after expiry of the modified response time. Thus, the complete segment is provided at the earliest upon expiry of the modified response time (i.e., not before), but possibly after. In practice, the complete segment is provided when the two following conditions are satisfied: the segment is available in complete form (its download is complete), and the modified response time has expired.

Preferably, in all cases the segment is provided to the second buffer M2, and as such, step (b) may comprise the conversion into a format suitable for playback on device 11 of said segment of the first buffer M1. This step consists in transforming the raw segment into a converted segment, which can be played by the player of the device 11 unlike the first one.

For example, if the player is the built-in player of a browser compatible with HTML5, the conversion consists of injecting the video data of the segment using the API Media Source Extension of the browser.

Naturally, step (b) advantageously comprises the simultaneous playback of a previous segment stored in the second buffer M2, so that the segments "rotate". The segment retrieved in step (b) will be soon played in turn.

Steps (a) and (b) can now be repeated as long as the playback lasts: the application of the time stabilizes the ABR, and at worst, in case of change anyway in the quality level required by the player due to a new ABR decision, the segments will now be loaded from the P2P network in accordance with the new required quality level, so that the user won't be bothered.

Prediction

In a particularly preferred manner, step (a) further comprises the prediction of a possible next ABR decision, i.e., the prediction of the quality level in which the next segment will be required (i.e., at the next occurrence of step (a)).

Indeed, it is possible by application of the first function, to determine before applying the calculated response time, whether the quality level required by the ABR will change.

It will be assumed that a complete segment, of length p , has been provided so as to facilitate the calculations. Thus, at the end of step (b), the fill level of the second memory M2 will have increased by p , i.e., the length of the segment. On the other hand, the response time ρ has elapsed, and an identical duration of the content of the second

memory M2 has been played and deleted, so that the new fill level B_n can be expressed relative to the previous one B_{n-1} (at the time of the last provision of a segment) by the formula $B_n = B_{n-1} + \rho - \rho_n$ (it is noted that a similar calculation can be made if the parameter representative of the segment reception rate is the bandwidth). Therefore it is sufficient to calculate $f_1(B_n)$ to predict the next ABR decision, and if necessary anticipate a change in the quality level and require the future segments in the right quality level (this is "pre-fetch"). Thus, the probability that the user will be bothered is even lower.

Note that the method may comprise a step (c), at the end of step (b), of verifying the prediction (if necessary confused with a next occurrence of step (a)).

More precisely, the ABR logic is relaunched to calculate a "new" required quality level, i.e., the required quality level for the next segment, and a corresponding request for segment will in this regard be sent to the client software. The verification simply consists in comparing the predicted quality level with the required quality level in practice, for the next segment.

Learning

According to a first embodiment, the first function is assumed to be known, and the second function can be pre-constructed from this first function.

However in other cases, the first function may be poorly known or not known at all if the player is completely closed.

Then, it is possible to gradually learn the first function and/or the second function, by using in particular the prediction of the ABR decision as a learning means: for each segment, the prediction and the decision actually made, i.e., the predicted quality level and the quality level actually required, which constitutes a learning loop, are compared in step (c).

It is possible to estimate the first or the second function, in particular starting from a default function, and by changing its parameters over time. For example, the first function can be parameterized via the values of the four parameters B_{min} , R_{min} , B_{max} and R_{max} by assuming a piecewise function with three segments, and similarly the second function can be parameterized via the values of the four parameters B_{min} , ρ_{min} , B_{max} and ρ_{max} by assuming a piecewise function with three segments (therefore, a total of six variables).

Thus, on the first segments of the content, the client software will probably calculate the response time in an inaccurate manner, which will lead to unexpected ABR decisions and uncontrolled quality level changes, but the algorithm will quickly improve and the ABR logic will be completely under control.

In particular, learning algorithms can be used, for example, neural networks.

Note that the data upon which the learning is done can be kept between sessions to eliminate or limit the problems related to the learning phase during subsequent sessions.

5 Some factors, such as a change in the version number of the player, can be used to invalidate the saved parameters and start over blank parameters for a new learning phase.

Alternatively/additionally, the learned functions can be transmitted to other peers 12, for example via P2P messages, or to the peer management server 3.

Device and computer program product

10 According to a second aspect, the invention relates to the client device 11 for the implementation of This method for playing content, on a player adapted to choose the quality level of the segments in accordance with adaptive bitrate regulation logic, ABR.

This device 11 comprises, as explained:

- A first buffer M1 temporarily storing at least one segment of content
15 consisting of a sequence of segments, each raw segment being in a format suitable for transfer within peer-to-peer network 10;
- A second buffer M2 temporarily storing at least one converted segment of said content, in a format suitable for playback by the player; and
- A data processing means 110.

20 The data processing means 110, typically a processor, is configured to implement, following receipt of a request for a segment, the provision (to the player, typically by injection into the second buffer M2) of said segment into the first buffer M1 upon expiry of a response time defined with respect to said player's ABR logic.

25 According to other aspects, the invention relates to a computer program product comprising code instructions to execute (on data processing means, particularly those of the client device 11) a method according to the first aspect of the invention for streaming, on a client device 11, a broadcast content within a peer-to-peer network 10 of client devices 11, 12, as well as storage means playable by a computer device (for example a memory of this client device 11) on which there is this computer program product.

CLAIMS

1. A method for streaming, on a player of a client device (11), a broadcast content within a peer-to-peer network (10) of client devices (11,12), said content consisting of a sequence of segments available in a plurality of quality levels and said player being adapted to choose the quality level of the segments in accordance with an adaptive bitrate regulation logic, ABR; the client device (11) comprising a first buffer (M1) adapted to store segments in a format suitable for transfer within the peer-to-peer network (10), the method being characterized in that it comprises the implementation, by data processing means (110) of the device (11), of steps of:
- a) Receiving a request for a segment from the player
 - b) Providing, in response, said segment from the first buffer (M1) upon expiry of a response time defined with respect to said player's ABR logic.
2. The method according to claim 1, wherein said ABR logic is defined by a first function making it possible to calculate a quality level in which said segment is required as a function of at least one parameter representative of a segment reception rate.
3. The method according to claim 2, wherein a second function makes it possible to calculate in step (a) said response time as a function of said at least one parameter representative of a segment reception rate.
4. The method according to claim 3, wherein the second function is constructed from the first function.
5. The method according to any of claims 3 and 4, wherein the second function is monotonic compared to said at least one parameter representative of a segment reception rate.
6. The method according to any of claims 1 to 5, wherein the device (11) further comprises a second buffer (M2) adapted to store segments in a format suitable for playback by the player, said segment being provided in step (b) in the second buffer (M2).
7. The method according to any of claims 2 to 5 and claim 6 in combination, wherein said parameter representative of a segment reception rate is a fill level of the second buffer (M2) and/or a bandwidth.
8. The method according to any of claims 1 to 7, wherein step (a) comprises the prediction of a quality level in which the next segment will be required as a function of the calculated response time.
9. The method according to claim 8, comprising a step (c) of verifying said

prediction of a quality level in which the next segment will be required.

10. The method according to claim 9, comprising the learning of the second function on the basis of the predicted and the quality levels actually required.

5 **11.** The method according to any of claims 1 to 10, wherein the required segment is available in incomplete form in the first buffer (M1), step (a) comprising the modification of the calculated response time based on an estimated time necessary to finish retrieving the segment, the segment being provided in complete form in step (b) at the earliest upon expiry of the modified response time.

10 **12.** A device (11) for streaming, on a player, a broadcast content within a peer-to-peer network (10) of client devices (11, 12), said content consisting of a sequence of segments available in a plurality of quality levels and said player being adapted to choose the quality level of the segments in accordance with an adaptive bitrate regulation logic, ABR; the client device (11) comprising a first buffer (M1) adapted to store segments in a format suitable for transfer within the peer-to-peer network (10) and data processing
15 means (110), the device (11) being characterized in that the data processing means (110) are configured, following the receipt of a request for a segment from the player, to provide, in response, said segment from the first buffer (M1) upon expiry of a response time defined with respect to said player's ABR logic.

20 **13.** A computer program product comprising code instructions to execute a method, according to any of claims 1 to 11, for broadcasting a streaming content in a peer-to-peer network (10) of client devices (11, 12, 13) connected to a content server (2), when said program is executed by a computer.

25 **14.** A computer-readable storage means upon which a computer program product comprises code instructions to execute a method, according to any of claims 1 to 11, of streaming content into a peer-to-peer network (10) of client devices (11, 12, 13) connected to a content server (2).