Distributed Load Balancing for the Adaptive Video Streaming Using CDN with Ring System of Server Consolidation by Circulant Topology

H A Nekrasov¹,², A Yu Romanov¹,³
¹National Research University Higher School of Economics, 20 Myasnitskaya Ulitsa, 101000, Moscow, Russian Federation
²E-mail: alax-27@mail.ru
³E-mail: a.romanov@hse.ru

Abstract. This article is devoted to development of an automated service for media content delivery network when adaptive online streaming on the Internet. This allows automatically adjusting of the amount of transmitted data per unit of time (bitrate) with changing in quality of requested media content accordingly. Consequently, the quality will be underestimated to those using low-bandwidth data channels, and vice versa, the content will be provided with highest possible quality for those connected to high-bandwidth communication channels. In this process, a user does not take any part: client video player itself determines which bitrate will optimally display the content to a particular user in order to avoid video buffering with the highest possible quality ensured. Switching between bitrates can occur continuously with a decrease/increase according to changes in quality of currently available bandwidth. In addition to adaptive video broadcast, the developed service allows transcoding of video streams on the fly. After receiving of input RTSP/RTMP stream, the network converts it to RTMP/HLS multibitrate, resulting in the generation of video streams of different quality which can be supported by both mobile devices and personal computers.

1. Introduction
In today’s context, no Internet resource, such as a blog, social network, or news resource, can be imagined without the use of media content. Photos, graphics, animations, audio, and videos are most common “digital assets”. Over the past decade, the use of mobile gadgets to view such “assets” has increased worldwide to a significant extent. More and more users, instead of usual textual exchange of information, give their preference to exchanging of media content, as well as to viewing of stream videos (news or social events) on the mobile devices and prefer to do it not only at home, but also on the road or on vacation.

According to a recent study, conducted by Cisco [5], by 2021 nearly 70% of all mobile traffic will have belonged to video content. The company notes that according to their forecasts, the annual growth of video traffic will be about 55% up to 2021. Social networks and websites, actively building video recordings on their pages, make a significant contribution to such a rapid development of media content provision.

As a result, requirements to quality delivery of multimedia content have substantially increased which entails, in its turn, emergence and spread of new broadcasting technologies.

Since mobile gadgets often use a wireless connection to the Internet, a number of important issues arise, and they cannot be neglected. These issues directly depend on the following factors: technology
of the radio signal propagation, traffic prioritization, and other processes affecting the speed and quality of video content delivery. Taking into account all the above, we can draw a simple conclusion that guaranteed delivery of streaming videos over mobile networks can only be adaptive.

The basis for the implementation of automated service for video streaming is the technology of adaptive broadcasting. This approach allows delivering of media content to a user with a sufficiently high reliability and quality with adaptation to constantly changing connection speed. This solution also focuses on the type of the end user’s device, its screen resolution, and also bandwidth of Internet connection during playback which makes it possible to get rid of the picture deadlock, long buffering, and other distortions.

When adaptive broadcasting, the original source of high quality video is converted into a wide range of video files of the different quality (bitrates) which are saved for the later distribution in CDN content delivery network [11, 12].

Whenever a user decides to reproduce the adaptive video, broadcast server sends him a manifest file containing the information about all the video files of different quality available on the server (figure 1).

![Figure 1. Scheme of Adaptive Streaming.](image)

If there are several stream quality options available on the broadcast server, the video player will constantly monitor current bandwidth and select next high-quality segment it can download, thereby choosing the optimal bitrate to minimize buffering of media stream.

To distribute and deliver the video, received from the broadcast server to the end user, so-called Content Delivery Networks (CDN) is used [1, 3]. The main concept of CDN is that “asset” loading, carried out by multiple users, should not interfere with the server, where the main content is hosted; moreover, it should not even participate in it. The advantage of this approach is that these networks serve users’ “assets” without using any main server's computing resources [4, 6].

The proposed media content delivery solution is based on the use of distributed servers each of which performs its content delivering function with the highest speed and quality for any device in real time mode (figure 2).

Use of a multilevel distributed service contributes to horizontal scaling which is especially important, since it is now impossible to imagine a data service infrastructure without a load sharing on the server. In order to more evenly share the load and optimize network traffic, the developed platform contains the means to provide geographically distributed delivery infrastructure for the client content. The peculiarity of this architecture (in the context of its application to content management systems) is allocation of “common client content” servers being a single center for all the systems installed. Placement of the shared content between several servers (when CDN using) reduces network data path and accelerates service work from user’s point of view (figure 3).
Advantage of this technology is not only significant traffic savings for geographically remote offices with centrally installed system, but also considerable architectural improvements available to IT professionals, such as: ability to allocate common client content transfer with caching setting, improvement of system operational characteristics, optimization of server infrastructure, etc.

By placing data closer to a user, one can expect it to be delivered faster, let alone reducing the number of calls to the main server because of caching on distributed nodes. Also, it is always possible to transfer compressed data from the central server to CDN node in keep-alive connections and to configure CDN node so that competing requests being cached would not run in parallel.

2. Concept and Problem Solutions

The following principles were based on the concept of implementation of the proposed automated service:

**Video Stream Player.** The end user is provided with a player page that plays the video stream and allows (in addition to the adaptive quality) selecting the required quality of the video stream to play independently.

**Live Video Streaming.** This approach provides an opportunity to conduct online broadcasts of the media content with support for various qualities. In this case, conversion of the content to another quality is performed by the system itself. Broadcast is initiated upon the request from user’s player to distribution server. There are no limitations on the type of user player. It is possible to broadcast one event in several bitrates simultaneously. In this case, a separate stream is published for each bitrate. Each stream is published separately at simultaneous carrying out of several translations from different places.

**Screenshots Generating.** This service allows quickly accessing the “snapshots” (screenshots) of the live stream. Screenshot is a static image in JPEG format and allows users to preview the content offered for viewing. Screenshots are generated with a specified periodicity (the default is 1 minute) and size. Several screenshots of different sizes can be created simultaneously. Screenshots of the stream are available for download via HTTP and have a permanent URL for each given size. When the thread stops, the last captured screenshot will be displayed.

**HTTP Content Caching.** Content is cached on the distribution servers as needed and is returned from the optimal server at the user’s request. When caching, not only the data object itself, but also associated HTTP headers are saved, which allows caching both statically and dynamically generated content. Content is downloaded via HTTP protocol to distribution server cache from source server on the first request of the user and is used for servicing the subsequent requests. The object, received by distribution server, is stored in the cache for a certain time. In case of cache filling, the objects are
replaced by the rotation principle, that is, the objects with the longest storage period are replaced. Reloading from the source occurs only if there is no requested object in the server’s cache.

**Redirection Requests.** A user requests redirection to optimal distribution servers at DNS level. Distributed content must be available for domain names that are in CDN responsibility zone. For this, the domain name in CDN zone is allocated; it should be used when generating links to the content.

**Video Editing.** It is possible to mount a video on the fly; that is, to play it not entirely, but only some of its parts. For example, one can require CDN to play a video from the 5th to the 70th second, then from the 100th to the 106th, and so on. This service significantly reduces the time of publication of a video clip on the site.

**Video Conference.** It is possible for end users to communicate with each other in a chat room with live video.

Mobile devices focus; support for broadcasting to the mobile devices (using HLS protocol) is provided.

**Traffic Balancing.** It is possible to evenly distribute the users between servers.

The work of implemented automated service is schematically shown in figure 4.

---

**Figure 4.** Schematic Display of Service Operation.

---

It should be emphasized that the project has chosen adaptive broadcasting technology, since one of its advantages is support for both live broadcasts and recorded content which is important with current distribution of agreement between Internet providers on the traffic exchange. Adaptive broadcasting provides the data encryption and user authentication through HTTPS protocol, which allows protecting the rights of the owners of published material.

Coding server receives RTMP/RTSP stream, recodes it to different image quality, and converts it into RTMP/HLS multi-bitrate stream.

RTMP (Real Time Messaging Protocol) is convenient because AS3 programming language provides its support; it does not require a lot of resources on client workstation and has lots of advantages (broadcast with variable bitrates and switching clients to a higher quality if there is a free channel).

HLS (HTTP Live Streaming) is a protocol that is currently supported by most of mobile devices [2, 9]. This technology, developed by Apple, provides coding on a source server into a multitude of segments of different quality. Video data is encoded in H.264 format, audio data – in MP3, HE-AAC, or AC 3, and then it is packed into MPEG 2 TS container for later distribution. The generated content of different quality is divided into a series of 10-second fragments (chunks) information about which is entered the manifest/playlist file with extension *.m3u8 [10, 13].

Each user receives an individual playlist, which includes links to main manifest file containing information about all levels of quality and location of content on the server, which is a prerequisite for correct choice of the adaptive quality/bitrate by user's media player. When the playback process starts, the player initiates a sequential download of 10-second ‘chunks’. Initially, it chooses the parts of the video with a low quality, and then, based on the speed of user's Internet connection, it starts downloading video clips of the higher quality.

HLS is now de facto the standard in the world of streaming video for the majority of the consumer devices. At the same time, RTMP remains indispensable if data transmission with a speed close to
real-time broadcasting is required [15]. HLS gives some connection delay to download the first block with data, and in case of RTMP, displaying starts almost immediately. In general, it is currently the most advanced real-time media transfer protocol, despite all the ambiguities in the implementation and more time-consuming deployment and scaling with respect to HLS. RTSP protocol is still used as a fallback for Android devices of previous generations.

A set of free libraries with FFmpeg open source code is used as utilities that perform video conversion on the encoding server. They allow users to record, convert, and transfer digital audio and video recordings in various formats. The kit also includes libavcodec encoding library, audio and video decoding, libavformat library for multiplexing, demultiplexing audios and videos into a media container.

FFmpeg consists of the components most of which are involved in this project when implementing coding servers:

- ffmpeg – command line utility for converting video from one format to another; with its help, it is also possible to capture video in real time from TV-card;
- ffserver – HTTP streaming server (RTSP is currently being developed) for video or radio broadcasting;
- ffplay – a simple media player based on SDL and FFmpeg libraries;
- libavcodec – library with audio/video codecs;
- libavformat – library with multiplexers and demultiplexers for various audio and video formats;
- libavutil – auxiliary library with standard general routines for various ffmpeg components which includes Adler-32, CRC, MD5, SHA1, LZO-decompressor, Base64/DES/RC4/AES encoder/decoder;
- libpostproc – library of standard video processing subprograms;
- libswscale – library for video zooming;
- libavfilter – replacement of vhook which allows changing the video stream between the decoder and encoder in real time.

The choice of FFmpeg for implementation of the service being developed is conditioned by the fact that it is cross-platform and has all necessary functionality for working with video for broadcasting on the Internet. FFmpeg also provides APIs for convenient work with various programming languages supplied as extensions.

Distribution servers directly distribute the media stream to users. The principle of the system is to announce the network address which refers to a site located in a geographically distributed network (CDN), and to redirect the requests to a conditional single server; that is, they provide copies of content on the different servers around the world and send the client to the nearest client server. The result of this redirection is reduction in the delay between the request and response.

The web server with Nginx open source code was chosen as distribution and load balancing which falls on the distribution server [7]. It helps to avoid such a load when packets, transmitted over the network, are flooded to one server while other servers are idle. The balancer analyzes workload of servers and, if necessary, redirects the requests to a less busy server.

Nginx feature is the ability to process multiple concurrent connections that request digital assets either from primary storage server, or from caching server. Nginx work is initially built around a modular, event-driven, asynchronous architecture. It creates the work processes each of which can serve thousands of connections. Such a result is achieved thanks to a mechanism based on a fast cycle in which events are checked and processed. Separation of basic work and processing of connections allows each process to do its job and handle the connections only when a new event has occurred.

Each connection processed is placed in an event loop along with other connections. In this loop, events are processed asynchronously which ensures that tasks are handled in non-blocking mode. When a connection is closed, it is removed from the loop.
This approach to the connection processing provides Nginx zooming with limited resources. Because the server is single-threaded and it does not create the processes for each connection, the memory and CPU usage is relatively equal, even at high loads. Thus, the chosen architecture effectively uses the server resources ensuring the growth of CDN and ability to serve up to 10,000 requests per second.

Nginx does not have the ability to independently process the requests for dynamic content. To handle the requests, Nginx must pass the request to external processor for execution and wait for response to be generated, and upon receipt, send it to the client. This means that it is necessary for administrators to configure Nginx to communicate with such a processor by using one of the protocols known by Nginx (http, FastCGI, SCGI, uWSGI, or memcache).

The advantage of this approach is as follows. Since the interpreter is not built into each process, the overrun of resources, associated with such an event, will occur only when the requests for dynamic content are made, and static content will be returned to the client in a simple way which will cause queries to the interpreter to be executed only when it is required.

The choice of Nginx is also implemented because it can work not only as a web server, but also as a proxy server. Therefore, it works primarily with URI, translating them, if necessary, into the requests to the file system. This feature also manifests itself in the way Nginx constructs and interprets the configuration files. In the case of requests to static files, all requests must be mapped to the path in the file system, since Nginx first selects the server and location blocks that will be used to process the request, and then merges the document root with URI, according to the specified configuration.

3. Dynamic Site Acceleration and Use of BGP Protocol

Dynamic site acceleration (DSA) is used [14] as an approach to optimize the proposed solution and for more efficient delivery of the dynamic content in the project, resulting as follows:

- Connection management is improved through multiplexing the client connections and maintaining HTTP efficiency;
- Pre-selection of answers is carried out (uncatchable-web);
- Dynamic cache management is added;
- Additional real-time compression is provided;
- Full caching of pages is provided;
- SSL termination is disabled;
- Transmission through TCP protocol is optimized.

In addition, it is possible to optimize routes using BGP (Border Gateway Protocol) [8]. The use of routers using the BGP protocol allows obtaining information about the availability of networks and their attributes through which one can choose the best route and configure the routing policy.

4. CDN Network Topology Optimization by Applying Circulant Topologies

The idea of usage of a multilevel distributed service contributes to horizontal scaling of servers (Figure 2), but the tree-like topology of connecting the lower-level servers to the central server imposes certain restrictions, since the same traffic can be distributed to neighboring geographically located CDN servers. It seems to us that regional servers can be combined by horizontal connections, forming a ring topology, which will reduce the access to central server, in case the necessary traffic is already available on the neighboring nodes. This also increases the reliability of the system, in case of a failure of radial connections, since it is possible to deliver traffic to neighboring connections.

The distance between distant nodes in the ring can be also reduced by applying circulant topologies of compounds, for example, two-dimensional circulants with a single generatrix (figure 5) which have good parameters for conducting translational and complete exchanges [16].
It should be noted that in spite of the fact that circulant topologies are primarily used for constructing multiprocessor systems and networks on a chip, their topological properties are so attractive that there is a number of studies where they are proposed with the certain improvements for global networks [17, 18]. In case of content delivery networks, this topological solution also seems relevant to us.

5. Conclusion
Against a backdrop of the rapid development of social networks, e-commerce, and growing personalization of the Internet, an increasing portion of the content, provided to users, is created in real time. The users need fast, reliable, and customizable solutions that do not depend on the browser, location, device, or network. However, adding new features often slows down loading of the pages and creates inconvenience to users. Often, users access the Internet through shared networks which can be unpredictable in their work. Adaptive streaming technology dynamically adjusts the quality of video stream based on available bandwidth.

Standard support for CDN allows caching files closer to the users, speeding up the delivery of static files. However, with the development of dynamic web-applications, content caching in the boundary locations is not possible, because the server creates content in response to actions of users. Speeding up the delivery of such content is more difficult than in the traditional border caching. This requires a complete solution that accurately adjusts each element throughout the entire data path - from the time it was created to the delivery.

Thus, simply implemented, fast-acting, and budgetary solution in the form of an automated service has been obtained. It is based on free distributed software. The service allows making content more accessible to users which ultimately leads to an increase in the audience of the site and increases the loyalty of existing users. It reduces the response time of the web-site, increases the download speed of content and reliability of access to the site. It suggests the ways of improving the developed solution by applying the approach with the dynamic site acceleration and use of BGP protocol, as well as a topological approach based on the use of circulant topologies for horizontal integration of CDN servers.
The advantages of the proposed solution to the competitors are as follows: reduction of storage costs; acceleration of site by at least 70%; increase in the download speed of "heavy" content; reduction in coding costs; reduction of load on computing resources of web-server; reducing the load on communication channels; stability of the site to the outbreaks of attendance; increase in availability of the site for regional users; increase in fault tolerance of the site.

Acknowledgments

The publication was prepared within the framework of the Academic Fund Program at the National Research University Higher School of Economics (HSE) in 2018-2019 (grant № 18-01-0074) and by the Russian Academic Excellence Project "5-100".

References

[1] Buyya R and Pathan M and Vakali A 2008 Content Delivery Networks (Berlin: Springer Science & Business Media) p 417
[2] Austerberry D 2013 The Technology of Video and Audio Streaming (London: Taylor & Francis) p 360
[3] Vakali A and Pallis G 2003 Content Delivery Networks: Status and Trends IEEE Internet Computing, IEEE Computer Society pp 68–74
[4] Chen J B and Liao S J 2010 A Fuzzy-based Decision Approach for Supporting Multimedia Content Request Routing in CDN Parallel and Distributed Processing with Applications (ISPA) International Symposium pp 46–51
[5] Cisco: 70% of Humanity will Use the Mobile Communications by 2020 Access mode: https://www.cisco.com/c/ru_en/about/press/press-releases/2016/02-04a.html, free (date of circulation: 02.02.2018).
[6] Held G 2010 A Practical Guide to Content Delivery Networks Second Edition (USA: CRC Press) p 304
[7] Butler T 2017 NGINX Cookbook (UK: Packt Publishing Ltd) p 278
[8] Beijnum I 2002 BGP: Building Reliable Networks with the Border Gateway Protocol (USA: O'Reilly Media) p 290
[9] Russell J and Cohn R 2012 Http Live Streaming (Berlin: Book on Demand) p 94
[10] Cai Y and Magedanz Th and Li M and Xia J and Giannelli C 2010 Mobile Wireless Middleware, Operating Systems, and Applications (Chicago: Springer) p 508
[11] Miyauchi Y and Matsumoto N and Yoshida N and Kamiya Y and Shimokawa T 2012 Adaptive content distribution network for live and on-demand streaming (Santiago: Springer) p 37
[12] Pathan M and Sitaraman R K and Robinson D 2014 Advanced Content Delivery, Streaming, and Cloud Services (USA: John Wiley & Sons) p 504
[13] Kon F and Kermarrec A M 2011 Middleware 2011 (Lisbon: Springer) p 514
[14] Mur D 2012 The Second Space. The Art of Management and the Strategy of the Future. (M: Mann, Ivanov and Ferber) p 240
[15] Wankel C 2011 Streaming Media Delivery in Higher Education: Methods and Outcomes: Methods and Outcomes (Hershey: IGI Global) p 492
[16] Monakhova E A 2011 The Structural and Communicative Properties of the Circulant Networks PMM vol 3 13 pp 92-115
[17] Wang D 2015 Tradeoff study among traffic egression schemes and member allocation optimization for link aggregation groups in integrated switching systems Computer Networks vol 77 pp 56–72
[18] Ajwani D and Hackett A and Ali S and Morrison J P and Kirkland S 2016 Co-optimizing application partitioning and network topology for a reconfigurable interconnect Journal of Parallel and Distributed Computing vol 96 pp 12–26