IBC 2016

APPLICATIONS AND DEPLOYMENTS OF SERVER AND NETWORK ASSISTED DASH (SAND)

Emmanuel Thomas[1], M.O. van Deventer[1], Thomas Stockhammer[2],

Ali C. Begen[3], Mary-Luc Champel[4], Ozgur Oyman[5],

[1]TNO, the Netherlands, [2]Qualcomm Incorporated, Germany,

[3]MediaMelon, USA, [4]Technicolor, France, [5]Intel, USA

ABSTRACT

MPEG DASH provides formats that are suitable to stream segmented media content over HTTP. DASH clients follow a client-pull paradigm by adapting their requests based on the available bandwidth and other local resources. This has proven to be easier to deploy over CDN infrastructure than server-push technologies. However, this decentralised nature introduces new challenges such as offering a consistent and higher quality of service for premium users. MPEG is addressing this issue in the to-be-published new MPEG DASH part 5, Server and Network-assisted DASH (SAND). The key features of SAND are asynchronous network-to-client and network-to-network communication of quality-related assisting information. In addition, DASH-IF is further defining interoperable guidelines to optimise SAND deployments in a variety of environments: home network, over-the-top, etc. MPEG is expected to publish SAND by end of 2016 while DASH-IF aims for the course of 2017.

INTRODUCTION

Streaming media technologies have evolved over the past few years. HTTP-based adaptive streaming is today the technology of choice for streaming over the Internet. Many standards and industry bodies, including DVB, 3GPP, HbbTV and DASH-IF, have adopted the Dynamic Adaptive Streaming over HTTP (DASH) standard (1) published in 2012 by MPEG. The DASH formats are designed to be used in client-pull based deployments primarily with HTTP protocol for media delivery. A client first retrieves a manifest file, a Media Presentation Description (MPD), and then it selects, retrieves and renders content segments based on that metadata.

By leveraging HTTP communication, DASH offers some fundamental benefits over other streaming technologies such as: firewall pass-through, content being stored on regular HTTP servers, or high scalability. Most importantly, the DASH client selects segments of quality and size that fits within the bandwidth it has available. This way, buffer underruns are prevented and the end user can benefit from a continuous media experience.

However, the fundamental decentralised and client-driven nature of DASH also has some drawbacks. Service providers may not necessarily have control over the client behaviour. Consequently, it may be difficult to offer a consistent or a premium quality of service. There are many examples of situations where quality of experience can be affected. Resources announced in the MPD may become outdated after a network failure or reconfiguration, resulting in misdirected and unsuccessful DASH segment requests by the client. A DASH client can mistakenly switch to lower quality segments when a mobile hand-over or a cache miss is interpreted as a bandwidth reduction. Massive live DASH streaming may lead to cascades of cache misses in CDNs. A DASH client may unnecessarily start a stream with lower quality segments, and only ramp up after it has obtained bandwidth information based on a number of initial segments. Multiple DASH clients may compete for a shared bandwidth, leading to unwanted mutual interactions and possibly oscillations. As a consequence, service providers may not be able to guarantee a premium quality of service with DASH, even in managed networks where regular DASH clients may not fully take advantage of the offered quality of service features.

In 2013, MPEG started the Core Experiment on Server and Network-assisted DASH (CE-SAND), in which experts collected use cases and explored solutions. Based on the results of CE-SAND, MPEG initiated in February 2015 the edition of MPEG DASH part 5 SAND, ISO/IEC 23009-5 (2). This new part of MPEG DASH defines an architecture, data models and a protocol to solve the issues mentioned above. MPEG DASH SAND is expected to be published in Q3 2016. This paper presents the SAND architecture as defined in the MPEG DASH part 5 in terms of terminologies, data formats and protocols. In addition, concrete deployment scenarios are presented based on the ongoing SAND discussions in DASH-IF wherein experts are establishing guidelines to deploy DASH services augmented by SAND. The scenarios highlight the appropriate SAND messages to use as well as the achieved benefits for both the DASH clients and the service providers.

MPEG DASH PART 5 SAND

Standardisation Status in MPEG and DASH-IF

The SAND activity in MPEG started in July 2013. To kick-start the work, MPEG and IETF organised a joint workshop to discuss the main issues that hinder the cooperation between HTTP servers delivering the DASH content and the DASH clients. The experts came to the conclusion that the exchange of client and server-side information through the network would alleviate some of the issues inherent in the client-driven philosophy of MPEG DASH. For instance, the service provider has very little control on the decisions made by the DASH clients. Keeping the DASH clients informed about server failovers or other sort of delivery-related events may mitigate their negative impact on the quality of experience for the end users. On the other end, the DASH client has very little knowledge of the network conditions and must continuously estimate the network conditions (e.g., available bandwidth) while a network monitoring element could provide exact first-hand information (e.g., in a managed network).

Since then, MPEG experts worked on collecting use cases and technical solutions for the identified problems (see “SAND use cases and experiments” in (3)). The standardisation of MPEG DASH SAND is expected to reach the FDIS ISO stage by June 2016 which means that the first edition of SAND should be published by the end of 2016.

While SAND provides a set of well-defined messages, the specification leaves its implementation open in terms of message workflow, message frequency, the physical network entities that are SAND capable, etc. However, these aspects may significantly vary in different environments. For instance, it may be reasonable to have a high frequency of message exchange in a home environment while the same amount of messages sent over-the-top may cause unacceptable traffic overhead for the service provider. In order to improve interoperability around SAND, the DASH-IF (4) has initiated a task force on SAND in order to address concrete deployment scenarios that may be supported by SAND. This activity is expected to deliver SAND implementation guidelines with potentially SAND profiles to accommodate different deployment environments. The publication is expected in the course of 2017, supported by conformance and test tools.

SAND Architecture and Message Flow

Motivation for SAND was to bring additional capabilities to network elements present within a DASH infrastructure while ensuring that both SAND enabled network elements and regular DASH network elements can operate together within the same DASH infrastructure. As such, SAND architecture has been defined as an augmentation of the regular DASH architecture (see Figure 1) thanks to the addition of a SAND communication channel that allows SAND enabled elements (i.e., DASH clients, DASH servers, or any other network element present in the DASH infrastructure) to exchange SAND messages between them.

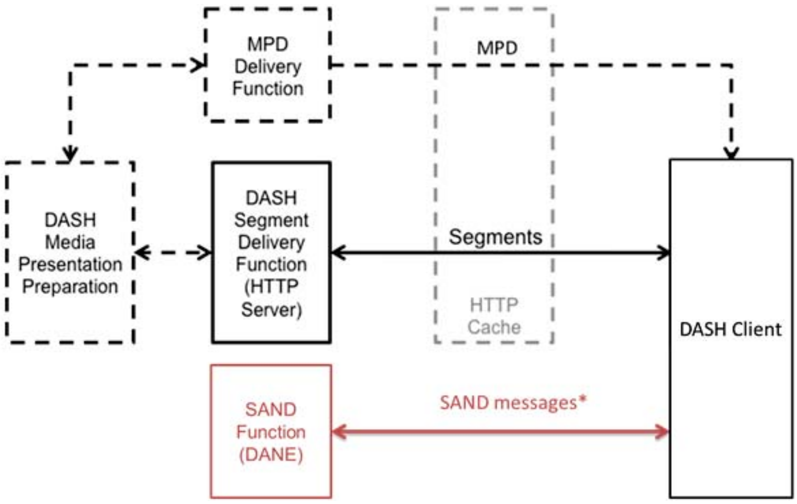

Figure 1 – SAND augmented DASH architecture

SAND reference architecture (see Figure 2) includes the following four categories of network elements:

- DASH clients with SAND capabilities,
- DASH-aware network elements (referred as DANE), which are network elements with minimum intelligence about DASH; for instance, they may be aware that delivered objects are DASH formatted objects, they may be able to parse DASH Media Presentation Description (MPD), or they may be able to have different behaviours depending on the DASH formatted objects they transport,
- DASH metrics server, being in charge of gathering DASH metrics from clients,
- and, finally, regular network elements (referred as RNE) which are present in the DASH infrastructure but have no understanding whatsoever about DASH formatted objects and treat DASH delivery objects as any other object.

Examples of DANEs are DASH servers or caches with SAND capabilities whereas examples of RNE are regular HTTP servers (that serve DASH objects as any other object) or transparent caches. Note also that, as seen on Figure 2, DANE and DASH metrics server are not necessarily present on the DASH media delivery path (network path along which DASH segments are flowing).

Between those network elements, SAND architecture defines the following four categories of SAND messages:

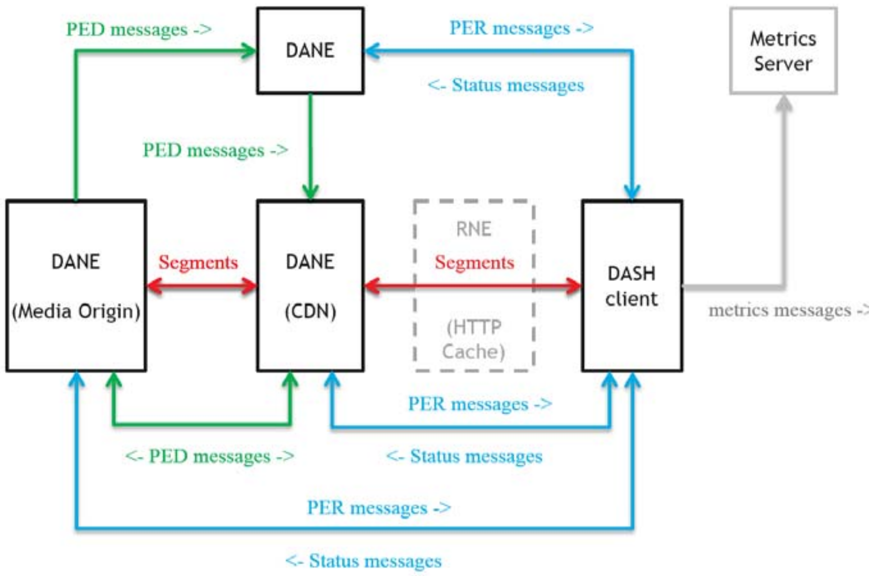

Figure 2 – SAND reference architecture.

- Parameters Enhancing Delivery (PED) messages that are exchanged between DANEs,
- Parameters Enhancing Reception (PER) messages that are sent from DANEs to DASH clients,
- Status messages that are sent from DASH clients to DANEs,
- Metrics messages that are sent from DASH clients to Metrics servers.

SAND specification defines the semantics of a total of 22 messages. For instance, the `QoSInformation` PER message carries up to 4 parameters (`gbr`, `mbr`, `delay` and `pl`) that each gives QoS related information to the DASH client (namely, guaranteed bit rate, maximum bit rate, maximum packet delay and packet loss rate).

For interoperability purposes, SAND specification also defines minimum message formats (XML and text) and a SAND transport protocol (HTTP) but does not preclude the use of other formats or transport protocols. Indeed, SAND specification also defines a signalling mechanism for the SAND transport protocol and gives an example of such signalling for the use of WebSockets (8) rather than HTTP.

Depending on the nature of the SAND message (PER, PED, status or metrics), the minimum transport protocol is HTTP GET, HTTP POST or HTTP header extensions (see **Error! Reference source not found.**). When HTTP header extensions are used, SAND messages are text formatted and carry (parameter, value) pairs. For other transports, XML format (as defined in an XML schema in SAND specification) is used.

Table 1 – Mandatory usages of HTTP for carrying SAND messages

Metrics messages	HTTP POST (HTTP headers may be used for small metrics messages.
Status messages	HTTP headers
PER messages	HTTP GET
PED message	HTTP headers (HTTP POST may be used)

Within a DASH infrastructure, SAND capability can be initiated thanks to two mechanisms: explicit signalling of SAND communication channel in the MPD through the `sand:channel` element, or the use of a specific SAND HTTP header extension that gives a hint to recipients about the presence of SAND elements in the infrastructure. Such hint would usually be inserted by a SAND enabled DASH server in the HTTP response to the first DASH segment request from a DASH client.

More information about which SAND messages are actually supported (note that it is not mandatory to implement all SAND messages) can then be exchanged between DASH clients and DANEs thanks to the `DaneCapabilities` and `ClientCapabilities` messages.

SAND DEPLOYMENT SCENARIOS

Application to Home Network

DASH clients can now be found on many devices such as PCs, smart TVs, tablets, smartphones, STBs and it is not rare to have several DASH clients competing for the same Internet bandwidth within a home. Even if the home gateway has caching capabilities, it is highly unlikely that two DASH clients requesting the same content will actually request the same DASH content representation within a short time window so they can benefit from caching. Moreover, as those two DASH clients would actually compete over the same bandwidth, it is even likely that each of them would get a lower representation of the DASH content that what they could get if they could get the same representation together. Unfortunately, without SAND, DASH clients have no means to benefit from such collaboration between them.

Figure 3 shows an example of a home network with total bandwidth limited to 15Mbit/s within which two DASH clients are accessing the same DASH content with 2, 4, 6, 8, 12 and 20 Mbit/s representations.

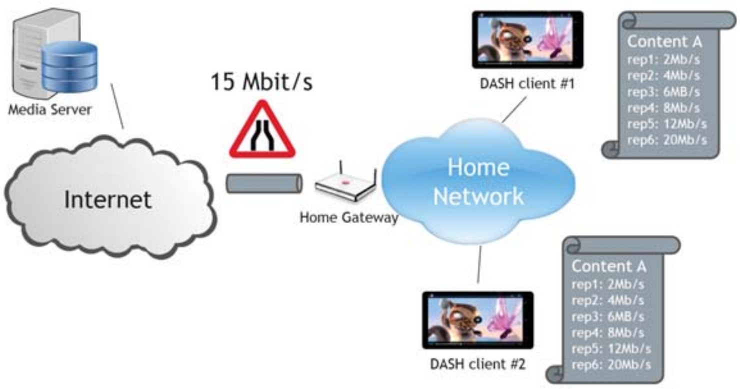

Figure 3 – Smart caching in home network

Without SAND capabilities, it is likely clients will be limited to 6Mbit/s representations. But if the home gateway and the DASH clients are SAND enabled, it is possible for the clients and the home gateway to work on a much more collaborative way. The `AnticipatedRequests` message allows clients to tell in advance the home gateway about which future DASH segments should be cached. Using the `AcceptedAlternatives` or `NextAlternatives` messages, clients can inform the home gateway that they are ready to accept other representations than the one they initially requested so that the home gateway may be able to serve them with already cached representations rather than retrieving new ones from the DASH server. Finally, using the `DaneResourceStatus` message, the home gateway can inform the clients about which segments/representations are already in cache. With all those capabilities, with the configuration shown in Figure 3, both DASH clients would be able to play the highest possible representation at 12Mbit/s.

Another issue home users face as well when there are several DASH clients within the same home network is that the sharing of bandwidth can be unpredictable and the quality of premium content may be severly impacted by the streaming of lower quality contents.

Another example could also be that some DASH clients would not be able to access their requested content because a very aggressive DASH client is already consuming most of the bandwidth. If the home gateway and DASH client are SAND enabled, it becomes possible to set-up a collaboration between DASH clients. Using the `SharedResourceAllocation` message, clients can tell the home gateway about the bandwidth requirements of the various representations they are willing to accept and they can also optionally carry user preferences about content priority. After receiving bandwidth requirements from all DASH clients, the home gateway can compute the maximum bandwidth each DASH client should be limited to in order to ensure user preferences are met (such as premium quality first, fair sharing or guaranteed or service for everybody) and send such information to each client by using the `SharedResourceAssignment` message. If DASH clients follow the bandwidth recommendations given by the home gateway, then successful collaboration between DASH clients within the same home network becomes possible.

Outdated MPDs in a CDN

Although MPEG DASH was designed for delivery over Internet, it is not necessarily friendly with regards to delivery over Content Delivery Networks (CDN). Section 4.6 of RFC 7336 (5) lists a set of HTTP adaptive streaming concerns. A first concern is the large number of segments, which makes content acquisition and file management much more complex than for single-file content. A second concern is the sheer volume of logging information for the delivered segments and the complexity of logging aggregation, especially when multiple CDN delivery nodes are involved in a specific DASH delivery session. A third concern is managing the purging of DASH content from a CDN, making sure that the correct set of segments is purged.

The fourth and perhaps the most important concern is the interaction between CDN-based request routing and MPD-based DASH delivery. Request routing is the process of resolving and redirecting to the most appropriate CDN delivery node to deliver a specific piece of content to a specific DASH client. However, if each DASH segment has been individually request routed, then the request routing would pose a huge signalling load to the CDN. Moreover, the request-routing delay would significantly impact the CDN throughput (6). Alternatively, each DASH client could receive a dedicated MPD that identifies specific CDN delivery node(s). Section 3.3.2.2 of RFC 6983 (7) specifies an information-exchange model that enables this "personalised rewriting" of MPDs.

Whereas the MPD-rewriting solution is recognised as being the most efficient, it also has some brittleness. CDN delivery nodes may go in overload, go offline or have content purged, while DASH clients still have MPDs pointing to those nodes. When a DASH client attempts to retrieve a segment, the result will be a "404 Not Found" error, resulting in the video playout to halt and an unhappy user experience.

SAND provides a set of solutions to the problem of out-of-date MPDs. Clause 9 and 10 of SAND (2) specify the SAND communication channel. When the CDN detects that a CDN Delivery Node becomes unavailable, it may use the SAND communication channel to push an updated MPD to the DASH client, or to trigger the DASH client to retrieve a new MPD. The latter option is specified in clause 6.5.4 of SAND, using a `MPDValidityEndTime` PER message which carries the wall-clock time at which the MPD will no longer be valid. A DASH client may be triggered to immediately retrieve a new MPD by setting the value of the `validityEndTime` attribute to the current time, see Figure 4. Alternatively, this attribute may also be used to announce scheduled outages in the (near) future.

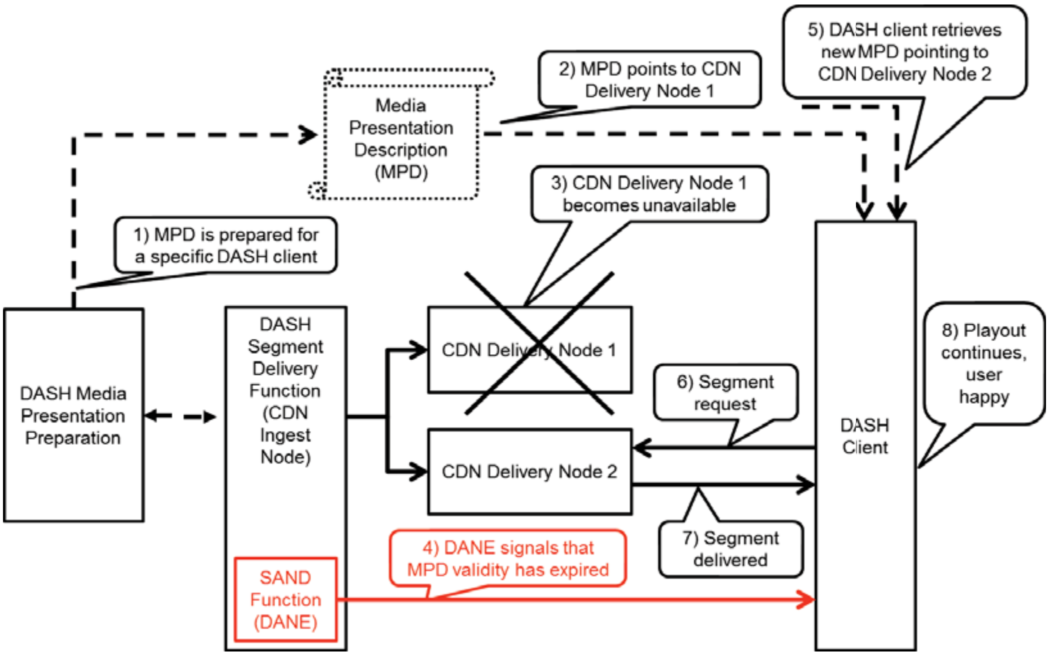

Figure 4 – CDN-integrated DANE triggers DASH client to get new MPD

Intelligent Edge Caches for Mobile Users

Video streaming over mobile broadband networks today is challenging due to limitations in bandwidth and difficulties in maintaining high reliability, quality, and latency demands imposed by rich multimedia applications. Yet, the consumer demand for mobile video services continues to grow and hence new wireless multimedia solutions are required to optimize future wireless networks for video services and deliver enhanced Quality of Experience (QoE).

In this context, SAND can play a central role in mobile networks by enabling edge optimizations. Caching, analytics and resource allocation, can be improved by the high proximity, ultra-low latency and high-bandwidth of edge connections. In addition, the mobile broadband experience can be differentiated to deliver better streaming performance. For instance, the local edge server inside the mobile network can cache DASH segments that DASH clients are expected to request. Similar to the use case “Application to Home Network”, the DASH clients can inform the local edge server about the next segments to be requested by sending the SAND status messages `AnticipatedRequests`. This way, the local edge server may decide to prefetch the content for faster delivery. The client may also provide a deadline using `AbsoluteDeadline` or `MaxRTT` or for the delivery of the Segment such that the edge server can schedule and pace the delivery to receive the request just-in-time without overloading the network.

Another benefit of SAND for mobile networks is that it can enable intelligent resource allocation to deliver consistent QoE across the DASH users. The edge server may for example determine and fix the throughput certain users during a period of time to avoid over or underload using the `Throughput` or `QoSInformation` messages. Such throttling technologies have proven providing benefits for example in OTT streaming services such as T-Mobile's Binge-On (9). SAND may be a candidate to provide standardized interfaces across content provider and edge caches, as well as towards the client.

At the moment, 3GPP's SA4 Working Group is conducting a Release 14 feasibility study on SAND in order to identify enhancements offered by MPEG DASH SAND (ISO/IEC 23009-5) in the 3GPP environment, and recommend necessary modifications to the 3GPP specifications including DASH to enable these enhancements.

CONCLUSION

Server and Network-assisted DASH (SAND) augments traditional DASH services by compensating the limitations of client-controlled HTTP streaming. The technology is expected to assist and enhance the operation of client-centric DASH streaming by a close cooperation between DASH clients and service provider servers. The technology addresses messages as well a communication channel in order to fulfil the different requirements and use cases that were collected in the MPEG standardization process. In addition, DASH-IF is working on interoperable guidelines to make SAND interoperable and optimised for a broad set of applications and use cases.

REFERENCES

1. MPEG, Dynamic Adaptive Streaming over HTTP (DASH), ISO/IEC 23009, 2012.

2. MPEG, Dynamic Adaptive Streaming over HTTP (DASH) -- Part 5: Server and network assisted DASH (SAND), ISO/IEC 23009-5, 2016.

3. Thomas, E., van Deventer, M.O., Stockhammer, T., Begen, Ali C., Famaey, J., 2015, Enhancing MPEG DASH performance via server and network assistance, The best of IET and IBC 2015-2016, September, 2015, pp. 48 to 53.

4. DASH Industry Forum, source: http://dashif.org

5. Peterson, L., Davie, B., van Brandenburg, R., 2014, Framework for Content Distribution Network Interconnection (CDNI), RFC 7336, August, 2014, source: https://tools.ietf.org/html/rfc7336

6. Famaey, J., Latre, S., van Brandenburg, R., van Deventer, M.O., De Turck, F., 2013, On the Impact of Redirection on HTTP Adaptive Streaming Services in Federated CDNs, 7th International Conference on Autonomous Infrastructure, Management and Security, AIMS 2013, June 25-28, 2013, UPC Barcelona, Spain

7. van Brandenburg, R., Le Faucheur, F., Leung, K., 2013, Models for HTTP-Adaptive-Streaming-Aware Content Distribution Network Interconnection (CDNI), RFC 6983, July, 2013, source: https://tools.ietf.org/html/rfc6983

8. Fette, I., Melnikov, A., 2011, The WebSocket Protocol, RFC 6455, December, 2011, source: https://tools.ietf.org/html/rfc6455

9. T-Mobile, 2016, Content Provider Technical Requirements for Binge On, source: https://www.t-mobile.com/content/dam/tmo/en-g/pdf/BingeOn-Video-Technical-Criteria-March-2016.pdf, April, 2016