**INTERNATIONAL ORGANISATION FOR STANDARDISATION**

**ORGANISATION INTERNATIONALE DE NORMALISATION**

**ISO/IEC JTC1/SC29/WG11**

**CODING OF MOVING PICTURES AND AUDIO**

**ISO/IEC JTC1/SC29/WG11 MPEG2017/N**17062

**July 2017, Torino, Italy**

|  |  |
| --- | --- |
| **Source** | **Requirements** |
| **Status** | **Approved** |
| **Title** | **Use cases and draft requirements for Network Based Media Processing (v1)** |
| **Authors** | Kyungmo Park, Imed Bouazizi, Yiling Xu |

# Introduction

MPEG has developed various technologies for multimedia coding and transport, such as AVC/HEVC, 3D audio, MPEG-2 TS, ISOBMFF, DASH and MMT. These technologies have been widely adopted and are heavily used by various industries in various applications, such as digital broadcasting, audio and video streaming over the Internet, in mobile terminals, etc.

In order to develop standardized and efficient solutions for network-distributed video coding of media processing, especially given the recent increase in demand for distribution of MPEG media in next generation network environments such as 5G, MPEG evaluates and addresses the current limitations of available standards in the MPEG media distribution area including taking considerations of processing units in networks and challenges in emerging network environments into account.

This document contains use cases and draft requirements for potential NBMP standards addressing the needs of network-distributed emerging applications.

# Objectives

In this document a network-distributed media processing system is defined as follows:

* **Network-based media processing system:** A system for video encoding and decoding where media processing is distributed across three or more processing units. The processing units are interconnected through links with individual bandwidth constraints, and each unit has an individual processing capability. One of the units is the “original” encoder and one of the processing units is the “final” decoder.
* **Unit**: A node in a delivery network which executes functionality for video transactions in a distributed manner.

# Scope

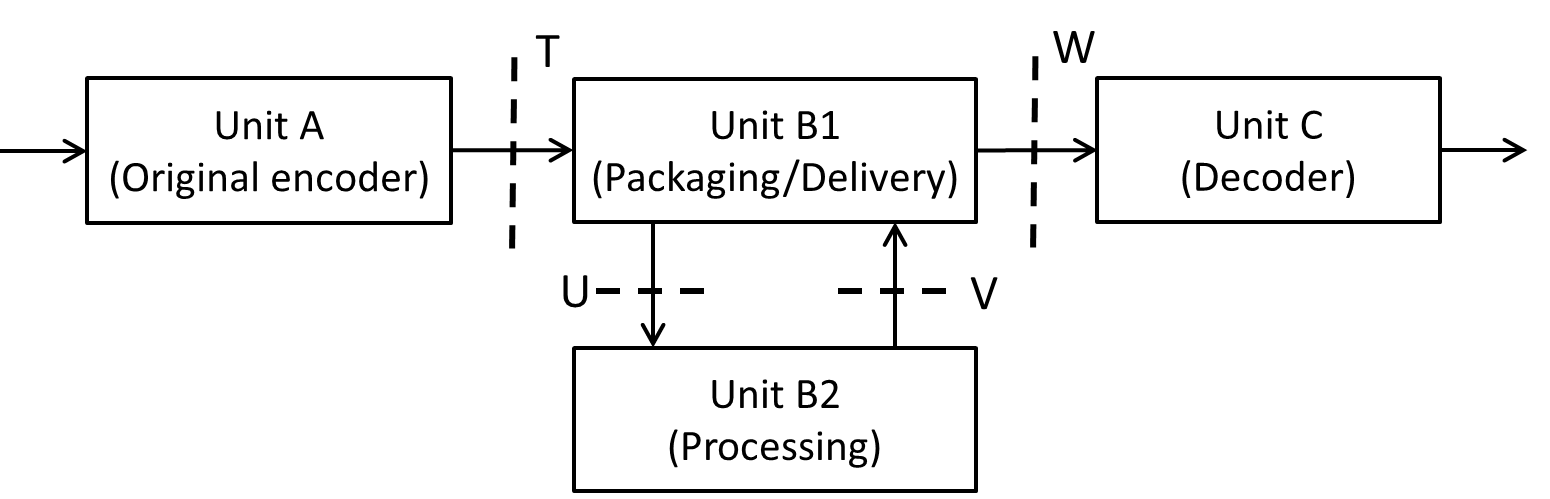


Figure 1 Potential framework for Network-Distributed Media Processing system.

The scope of this document is to present use cases and draft requirements for network-based media processing. The potential framework includes stream format specifications of the four interfaces T-W, illustrated Figure 1.

* Stream format to packaging/delivery unit (T)
* Stream format to media processing unit (U)
* Stream format to packaging/delivery unit (V)
* Stream format to decoding unit (W) (Note: Feedback interface needs to be considered)

Unit A is the original encoder capable of creating stream T includes system aspect. Unit B1 is a network entity to support the delivery of media content (e.g., DASH and MMT) from Unit A to Unit C. Unit B2 is a processing unit having media processing capabilities that may be needed if the format of media content requested by Unit C is not directly provided by Unit A. Units B1 and B2 could be co-located in single Unit or distributed in the network. Unit C is a decoder that conforms to an existing video compression standard such as AVC or HEVC and support the capability of system aspect as well.

# Use cases

The following section contains use cases of media processing applications that may benefit from network-distributed media processing.

## VR Processing in cloud

High quality 360° video content is uploaded to the Cloud to provide high quality Virtual Reality (VR) streaming service to the users. The service provider wants to save mobile data usage for users that are experiencing virtual reality through streaming. Hence, the service provider wants to take advantage of the high processing capability of the Cloud. The high processing capability of the Cloud facilitates the machine learning and media processing in the Cloud to provide advanced VR streaming services. Here the advanced VR streaming service aims at offering high quality VR experiences with low data consumption.

To save mobile data usage, a tiling and frame-packing based Field-of-View (FoV) VR streaming (or alternatively called view-dependent VR streaming) is used. Here, the tiling and frame-packing based FoV VR streaming allows the server to only send video data that correspond to the user’s current view port in high quality. A lower quality encoding of the whole panorama is streamed as fallback as described in the following:

1. Tiling approach: A high quality partition of the 360° scene and a low quality whole panorama are sent to the users as separate video streams.
2. Frame packing approach: A high quality partition of the 360° scene and a low quality version of the whole 360° scene are frame packed together into a single frame packed video stream and sent to the user. During the packing operation, rotation may be applied.

In both the tiling and frame packing based FoV VR streaming, the equi-rectangular projection (ERP) is typically used as input format. In addition, multiple encodings of each tile may be required to address different bandwidth and device capabilities and needs. This results in a significant amount of processing. Therefore, the service provider prefers to perform this VR processing in distributed networks. Compared to preparing the FoV VR content in a centralized manner, the usage of distributed networks would significantly reduce processing time and save network bandwidth.

The service provider may want to apply advanced machine learning techniques to optimize its VR encoding configurations. This processing functionality can be provided to content providers as a configurable “Processing as a Service” offering.

## Intelligent video up-scaling

A mobile user captures a video scene with her mobile phone. The video sequences captured by a mobile phone typically have an odd picture aspect ratio (PAR) due to a narrow width and/or height. When videos with such a PAR are played on a wider screen such as Desktop PC or Smart TV, the content will usually not cover the whole screen, which degrades the user experience. If traditional up-scaling is applied to match the video content to the presentation screen, the user experience might be even worse because it distorts the objects on the scene by applying a larger scaling factor on one dimension of the video.

A service provider wants to offer high processing capacity of the Cloud to provide intelligent video up-scaling feature to improve the use experience. The service provider may want to provide content adaptive up-scaling. The content adaptive up-scaling may be carried out either horizontally or vertically or on both dimensions. In intelligent up-scaling, the video scene is up-scaled so that the central objects of the video scene become more harmonious in terms of the ratio of central object(s) and background, position of central object(s) in the video background etc. An example of such an up-scaling is shown in Figure 2, wherein the top, mid, and bottom row figures show the original captured video frames, traditional up-scaled video frames and intelligently up-scaled video frames, respectively.

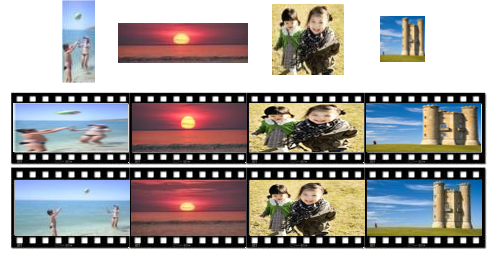


Figure 2 - Example of video (top), conventional up-scaling (middle), contents preserved up-scaling (bottom)

To provide advanced intelligent up-scaling, the service provider typically deploys Artificial Intelligence (AI) to improve the video quality and the resulting user experience. The service provider is being aware of the importance of learning the large data in machine learning. The service provider wants to deploy its unique advantages of the distributed networks or Cloud in the big database, high and parallel processing capabilities, and complete set of APIs to improve the AI-based intelligent up scaling. Being aware of the unique advantages of the Cloud or distributed networks in machine learning, the service provider expects that the Cloud-based and distributed networks based intelligent up scaling significantly improves the user’s quality of experience compared to the mobile device or personal computer based approaches; hence further improves the business of the service provider.

Different users typically have different perceptions of the quality of the scaled videos, which is another important motivation of locating the scaling processing in distributed networks as close as possible to the end users. By processing the up-scaling in the distributed networks, the service provider not only facilitates the intelligent up-scaling to meet users’ requirements but also saves network bandwidth as the up-scaling processes are carried out at a proximity from the user.

The distributed network ecosystem based solution has its unique advantage in terms of saving resources compared to the traditional broadcasting systems. In network based media processing, the complete media processing functionalities could be provided as an ecosystem and used by various service providers in a shared manner, which will significantly save resources compared to the traditional broadcasting system where each of the service provider needs to have its own media processing functionality.

## Intelligent user centric broadcasting

In traditional broadcasting of sports games, e.g. soccer and American football, multiple cameras are placed at various locations of the field. Video editing tools are typically used to edit multiple video scenes to produce a uniform video scene to broadcast to all viewers. In a user centric broadcasting scenario, the service provider wants to upload multiple scenes to the Cloud and broadcast user centric broadcasting service by utilizing the end user’s preferences. Many times a football fan might love a certain team and he/she may be a fan of a certain football star. In user centric broadcasting, the service provider wants to provide a service so that a user who is fan of a star could continuously follow the player within a sub-scene window, which is overlapping with the main scene window, and switch between the sub-scene to the main scene upon request. In user-centric broadcasting, the service provider wants to deploy AI to analyze the user’s preference and edit the video streams based on the user’s preference automatically.

To provide an intelligent user centric broadcast experience, the service provider typically deploys Artificial Intelligence (AI) to customize the user experience. The service provider is aware of the complexity of processing big data and fine tuning the machine learning models accordingly and wants to deploy distributed networks or Cloud processing of big data to improve the user centric broadcasting expeirence. Being aware of the unique advantages of the Cloud or distributed networks for machine learning, the service provider expects that the Cloud-based and distributed networks based intelligent user centric broadcasting significantly improves the user’s quality of experience compared to the central computer based approaches; hence further improves the business of the service provider.

In user centric broadcasting, the service provider wants to provide user aware broadcasting service by taking into account that different users or different groups of users have their own preferences. This is another important motivation for residing the processing and editing of the video streams in the distributed networks close to the users.

The distributed network ecosystem based solution has its unique advantage in terms of saving resources compared to a traditional broadcasting system. When the media processing functionality resides in distributed networks or in the Cloud, the complete media processing functionalities could be provided as an ecosystem and used by various service providers in a shared manner, which will significantly save resources compared to the traditional broadcasting system where each of the service provider needs to has its own media processing functionality.



Figure 3 - User centric broadcasting

## Video and AR

A service provider wants to provide mixed video and AR experience to its users. In mixed video and AR, the source video sequence is overlaid with graphics images. The service provider may want to produce multiple versions of such mixed video and AR content by overlaying different graphics images on top of the video taking into consideration the user’s preferences. An example of such scenario is shown in Figure 4.

Source Video

NDVC

(w/ User preference/ object tracking)

+ Graphic overlaying

Video 1

Video 2

Video n

Figure 4 Video and AR overlaying use case

Figure 4 shows an example of a volleyball sport video with overlapping graphics. In the example, the volleyball trajectory is augmented with moving graphic content that provides an augmented graphic trajectory of the moving ball. Multiple versions of such video and AR contents are prepared. The service provider then provides an appropriate version of the content to each user depending on that user’s preference.

The service provider wants to provide user aware AR experience. In a volleyball game, John is a fan of team A. In such a scenario, the service provider wants to add overlaying graphics by analyzing to cheer team A’s attack on team B. For the same volleyball game, Susan is a fan of team B. In such a scenario, the service provider wants to provide different graphics overlaying in support of team B.

To provide an advanced user aware video and AR experience, the service provider typically deploys Artificial Intelligence (AI) to achieve the required effects. The service provider deploys its unique assets of high processing power in the cloud to analyze the video content and the user’s profile and offer a tailored user experience.



Figure 5 An example of sport video and graphic overlapping

The distributed network processing solution has its unique advantages in terms of resource efficiency compared to the traditional broadcasting systems. When the media processing functionality is located in distributed networks or the Cloud, the complete media processing functionalities could be provided as Processing as a Service to various service providers in a shared manner, which will significantly save resources compared to traditional media processing systems, where each of the service provider needs to develop its own media processing functionality.

## Interactive media services

Multiple clients can interact through the processing unit B in NBMP. The client can upload the media content to the processing unit B, which can process and integrate the media content from the server and the multiple clients, and send the processed media content to different clients.

Figure 6 shows an example of the system for multi-client interacting based on processing unit B. Original video (S) is ingested and initially encoded into an initial stream (T) at an origin (A). The initial stream (T) is delivered to a node (B) within the media delivery network, up to the edge server. Processing unit B can simultaneously receive the media resources T from the server and the media content W1, W2, W3, W4 generated at clients. The processing unit B can process and integrate the media content, and then the processed media content U1, U2, U3, U4 can be delivered to the corresponding clients. Note that different versions of media content (U1, U2, U3, U4) need to meet the requirements of different device capabilities and media content.



Figure 6 Application example: the multi-client interacting through processing unit B

The following use cases appear to be feasible in a multi-client interaction scenario.

### Use case 1: media processing for Vehicles

In the Internet of Vehicles scenario, the existing technology is the car to get around the traffic information and some of the surrounding vehicles simple information, in the local information processing, to guide the safety of the car driving. However, these information is often limited, single car's processing capacity is limited. In the Internet of Vehicles, many vehicles can be connected by Wi-Fi, LTE-U or other ways for free information exchange. A processing unit B may be added to these devices to improve the processing capability of vehicle status information, traffic information, etc., to obtain more reliable and effective results to guarantee safe driving.

As shown in Figure 7, in the Internet of Vehicles, the unit B (monitoring & control infrastructure) may be established as a roadside unit, and the vehicle in the area covered by the roadside unit transmits its own position, speed, acceleration, steering information and so on. At the same time，the roadside unit side itself can also collects traffic information through the sensor, such as the front may be pedestrians crossing the road. Roadside units also have stored some local information, such as the map information. Number of adjacent roadside units can communicate with each other and exchange traffic information. In addition, the upper side of the roadside unit may have an upper layer information processing and control unit, and the roadside unit may upload information to the upper unit, and the upper unit may also send information to the roadside unit. Road side unit can process information from all above (vehicles, adjacent roadside units, upper layer unit, local storage, and calculate the control information used to guide each car's safe driving, such as to remind the driver in front of the car may break, slow down, or broadcast traffic information to all cars.

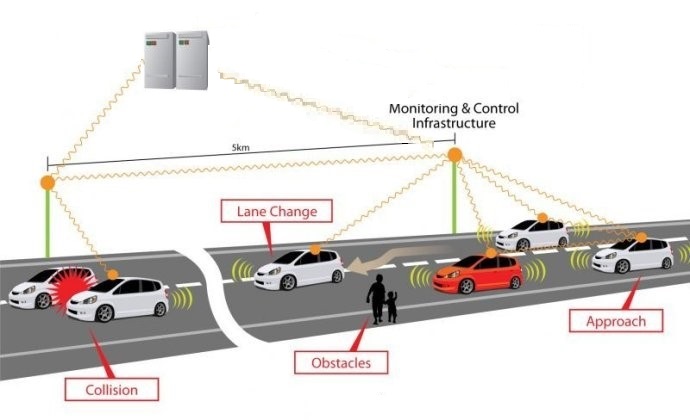


Figure 7 Unit B as monitoring & control infrastructure in Internet of Vehicles

### Use case 2: media processing for interactivity

In an interactive game scenario, you can build a processing unit, unit B in NBMP, that is closer to the client which performs server-like functionality. This can reduce server workload, speed up processing, and reduce server-side bandwidth consumption. The game user uploads the operation information to the unit B, the unit B may give the game control information according to the operation information of the plurality of users. If the two users use the mouse to move the character, the unit B calculates the position according to the user's mouse movement displacement and the user's previous spatial position Whether the two users meet, and based on the role of two users to determine whether to give a match.

As shown in Figure 8, the figure, in city A, city B, city C, city D have a game intermediate processing unit B, the server in city E. When a number of game users (user 1, user 2, user 3) are in city A, the operation information of each user only need to upload to city A's unit B for processing. When the game users from different regions, such as game users from city A (user 4), city B (user 5) and city C (user 6), the user's operation information can be uploaded to the local processor, after the initial processing the information will be uploaded to processor in city D, and then processor in city D to do further processing.



Figure 8 The distribution of the unit B and the server

## Media aware caching

### Online transcoding

There have been three types of video transcoding strategies: In the first type，the content provider to provide different versions（bit rates） of the video content stored in the video distribution device, when the user requests a video content from the server, the video distribution device send the corresponding version to the client.

In the second type， the video provider sends only one video content with high bitrate to the video distribution device, and the video distribution device transcodes the high bitrate video content into different versions of lower bitrate. Before the users reveal theirs requests.

In the last type, the video provider sends the video content of the high bitrate to the video distribution device, and the video distribution device transcode the stored high bit rate video to a lower bitrate version according to the user request. The first two types require large storage at the distribution device, while due to the transcoding latency, the third type which facilitates instantaneous transcoding may suffer from higher probability of video freeze，leading to degraded user experience.

Caching is a technology that pre-fetches video content from original content server to the distribution devices that are close to edge users. Cache device will send the video content to users immediately if that content has already been cached. The original content server will send the remaining video content to the cache device. So, the client feels the transmission distance is the distance from the client to the cache device. Because the cache is closer to the client, user experience can be improved in regard to directly pulling of that content from original server.

In accordance with the idea of caching, in unit B in NBMP, we can pre-transcode only part of the video content and store them in the memory. When the client requests reveal, the transcoded video content is first sent to the client, while the remaining video content will be served in the way of real-time transcoding. So, with probably not so high transcoding speed, and the requirement for storage will reduce dramatically while user experience will degrade slightly, comparing with the first way, if the pre-transcoded video content is enough and the transcoding speed is fast enough.

Then what part of the video content to be pre-transcoded and how to schedule the real-time transcoding process is critical to user experience maintenance.

### Coded caching implementation

Code caching method is a promising caching technology which was proposed by Ali & Niesen recently [1]. By careful design of content placement across end user cache during off-peak hours, simultaneous multicasting opportunities are created to serve several different content requests. The coded caching method shows great potential in shifting peak-hour traffic to off-peak hours, which leads to considerable bandwidth saving in content delivery. We explain the method of coded caching by follow example.

Suppose there are two media contents A for an interview of one basketball player1, and B for another interview of player2 which provided by the same provider. Two consumers are watch these interviews simultaneously at home. Conventionally, the end user facility may prefetching parts of the media content. But once the cache misses, or one of the user change the player interview, the network needs to transport the whole new content to the users through the common delivery network.

The coded caching implementation means that caching facility only need to cache parts of the media content, let’s say A1(half of A) B1(half of B) for the first user and A2 B2 for the second user, in the placement phase. When in the delivery phase, the source only need to deliver some coded XOR information compute from A&B base on the actual demands of the users. In this case, no matter what the choice of the users are, the server only need to transport a small size of coded media content. The caching process is shown in Figure 9.

In order to utilize such promising caching technology in NBMP, we propose to facilitate coded caching implementation by the Caching node in NBMP structure. As we can see, this coded cache method needs to implement near the end users and it also needs some computational resources. Thus, firstly NBMP unit may serve as the user cache to store the bits and compute the raw information after receiving delivered information. Secondly NBMP may also serve as the source which can generate coded information about the media content and compute the computing deliver messages base on the demands of the users.

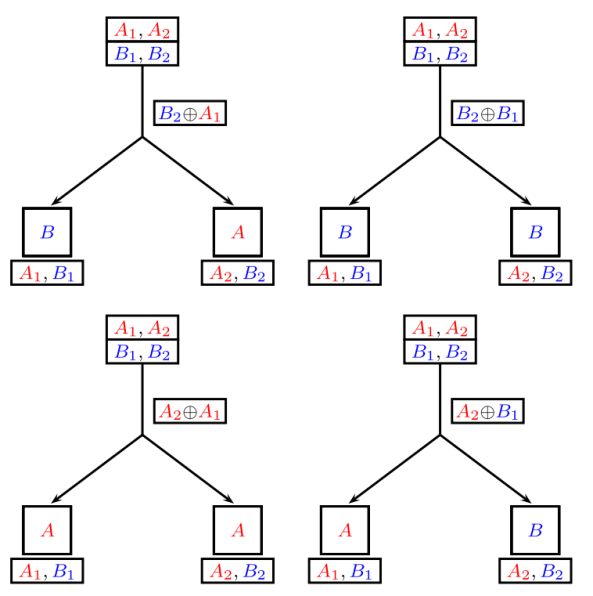


Figure 9 Coded caching process

### Targeting caching

We propose that NBMP node serve as family cache or dormitory cache rather than neighborhood cache. The promise range of NBMP cache serves several end users but not tens of hundreds of end users. Since that NBMP can pull precise popular contents or fragments from server or CDN nodes base on the local users’ interest, view history and other user features.

For example, if a family like sports more than news, and they like viewing TV shows on the mobile screens, the NBMP node may pre-fetch 4K sports fragment and HD TV shows in the off-peak hours. In the peak hours, their content request may be served partially or totally by the local caching node (family cache), thus reducing content delay by avoiding access to the original content server, from which the content transport rate is often limited by the network bottleneck.

In other words, NBMP may serve as CDN end nodes. The difference is that NBMP can target precisely to the little number of users so that improve the bit rate and increase the QoS.

### Large size caching

Since future media content like immersive media such as VR or point cloud always have large scale of data volume, the end user cache cannot afford enough storage for such services. NBMP can provide a large cache near the consumer because the implementation of NBMP includes the utilization of NBMP nodes with computational and memory resource. This NBMP large size caching may serve dozens of the users with higher definition, longer fragments, lower time jitters media service.

# Draft requirements for network based media processing

1. General Requirements
   1. NBMP shall have advantages in terms of compression, computation, bandwidth, storage efficiency over existing usage of MPEG standards for transcoding.
   2. NBMP shall support video coding distributed across processing units.
   3. NBMP shall support video coding with processing units that are interconnected with each other in a network.
   4. NBMP shall support transcoding for adaptive streaming in the network.
   5. NBMP should support functionalities for filtering, modifying and mixing of media content, and be aware of context.
   6. NBMP should support signaling of video quality metrics for media processing
2. Video Coding
   1. NBMP shall provide means for network-distributed video coding without affecting the compliance to existing decoding standards in the end-device.
   2. NBMP encoder shall provide means to signal NBMP metadata (e.g. side information).
3. Container Format
   1. NBMP shall support any type of media content (be media agnostic), including the existing MPEG codecs (which may produce very low to very high data rates) and MPEG formats such as ISO/IEC 13818-1, ISO/IEC 14496-12, ISO/IEC 23008-1 and ISO/IEC 23009-1.
   2. NBMP shall support identification of conformance points of each content component (e.g. elementary stream).
   3. NBMP shall support storage of multiple components for a single application.
   4. NBMP shall support delivery for applications that use common components (e.g. ISOBMFF).
   5. NBMP shall support storage of content that uses common components.
   6. NBMP shall support content component identification.
   7. NBMP shall support clock recovery (e.g. PCR)
   8. NBMP shall support in-band or out-of-band carriage of format information such as bit-rate, resolution, codec type, frame rate, dynamic range, colour space, sub-sampling type, segment duration.
   9. NBMP shall support a format for requesting transcoding of time segments of media data for adaptive bit-rate streaming.
   10. NBMP should support dynamic configuration (e.g. merging) of content components during delivery.
4. Delivery
   1. NBMP shall support streaming.
   2. NBMP shall support file delivery.
   3. NBMP shall support progressive download.
   4. NBMP shall support delivery over IP-based connection.
   5. NBMP shall support delivery over HTTP and RTP.
   6. NBMP shall support delivery over UDP and TCP.
   7. NBMP shall support delivery using existing MPEG protocols (e.g. ISO/IEC 23008-1).
   8. NBMP shall support delivery of media using existing standardized delivery formats.
   9. NBMP shall support push-based streaming, e.g., over unidirectional or multicast channels.
   10. NBMP shall support push-based progressive download, e.g., over unidirectional or multicast channels.
   11. NBMP shall support low latency delivery (e.g. to support conversational applications, live content, etc.).
   12. NBMP shall support use of different QoS types and levels.
   13. NBMP shall support low-complexity format conversion for delivery and storage.
   14. NBMP shall support relaying received content stored on storage devices.
   15. NBMP shall support control of content relay and retransmission.
   16. NBMP shall support dynamic media processing during delivery.
   17. NBMP should support signalling messages of media service for optimal use of network resources in distributed network entities.
5. Content Protection
   1. NBMP should support signalling, delivery and utilization of content using multiple protection and rights management tools.
   2. NBMP should support seamless change between content rights management schemes.
6. Delivery Environments
   1. NBMP shall support hybrid delivery environments, such as multiple transmission channels, possibly of different types.
   2. NBMP shall support multipath delivery.
   3. NBMP shall support both unidirectional and bidirectional communication environments.
   4. NBMP shall support seamless use of heterogeneous network environments including broadcast, unicast, multicast, storage, peer-to-peer, and mobile.
7. Media processing for VR
8. NBMP shall support to configure and execute processing of VR content to change encodings and frame packing variants
9. NBMP shall provide the APIs to enable the configuration and execution of VR processing in distributed networks
10. NBMP shall support to collect network traces and user consumption logs and to apply machine learning techniques to determine or fine tune the VR processing procedures
11. NBMP shall suppot the media processing of video streams with overlay graphics
12. NBMP shall provide the APIs to allow content providers to configure and execute graphics overlaying on their own video content in a secure way.
13. NBMP shall support to collect user preferences to configure the AR creation procedures
14. NBMP should support video analysis techniques on the video stream, e.g. to perform object recognition and tracking
15. Media processing for Intelligent video up-scaling
    1. NBMP shall support the compliance to the existing standards without affecting, e.g. container format and decoding in end devices etc.
    2. NBMP shall provide the APIs to configure and run up-scaling in the cloud
    3. NBMP shall provide the APIs to collect feedback and analyze user preferences
16. Media processing for User centric broadcasting
    1. NBMP shall support customize broadcasted streams in a user-centric way without affecting the compliance to the existing standards, e.g. container format and decoding in end devices etc.
    2. NBMP shall provide the APIs to enable the configuration and execution of customization processing of the broadcasted streams.
    3. NBMP shall support to create time and space overlapping sub-video streams from the main video stream.
    4. NBMP shall provide the APIs for the collection and analyzing user feedback and preferences, e.g. by deploying machine learning, shall be provided.
17. Media processing for interactive media
    1. NBMP shall support to upload locally generated multimedia content to the processing unit in the network
    2. NBMP should support controlling multi-client when multi clients interact;
    3. NBMP should support processing the multimedia content from clients;
    4. NBMP should support processing multimedia content form multi clients and server into different versions;
    5. NBMP should support sending different versions of multimedia content to corresponding clients.
    6. NBMP should support uploading information from client to database or create local database.
18. Media processing for media aware caching
    1. NBMP shall support pre-transcoding part of video content for efficient media caching in the network
    2. NBMP shall support media aware cache with existing coded cache technology.
    3. NBMP shall support targeting caching which related with user preference.
    4. NBMP shall support large size caching for considering the large scale of media distribution.

# References

[1] Maddah-Ali, Mohammad Ali, and Urs Niesen. "Fundamental limits of caching." IEEE Transactions on Information Theory 60.5 (2014): 2856-2867