**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/N16733**

**January 2017, Geneva, CH**

|  |  |
| --- | --- |
| **Source** | **Requirements** |
| **Title** | **Draft requirements for network distributed video coding** |
| **Authors** | Rickard Sjöberg, Xavier Ducloux, Kyungmo Park, Rufael Mekuria |

# Introduction

MPEG has developed various technologies for multimedia coding and transport, such as AVC/HEVC, 3D audio and MPEG-2 TS/ISOBMFF/DASH/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 MPEG media, especially given the recent increase in demand for distribution of MPEG media in heterogeneous environments, 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 draft requirements and use cases for potential NDVC standards addressing the needs of network-distributed emerging applications.

# Objectives

In this document a network-distributed video coding system is defined as follows:

* **Network-distributed video coding system:** A system for video encoding and decoding where 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, as illustrated in Figure 1.
* **Unit**: A node in a delivery network which executes functionality for video transactions in a distributed manner.

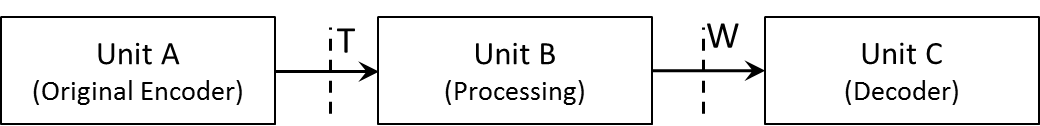


Figure 1- *Schematic view of a network-distributed encoding system.*

# Scope

The scope of this document is to present requirements and use cases for network-distributed video coding. One particular application is guided transcoding, this addresses video compression aspect with focus on links and processing units that are traditionally not covered by video coding standards. Another aspect of the scope is storage formats and delivery protocols to support network-distributed video coding such as cloud transcoding.

The scope includes stream format specifications of the four interfaces T-W, see Figure 2.

* Stream format to packaging/delivery unit (T)
* Stream format to transcoding unit (U)
* Stream format to packaging/delivery unit (V)
* Stream format to decoding unit (W)

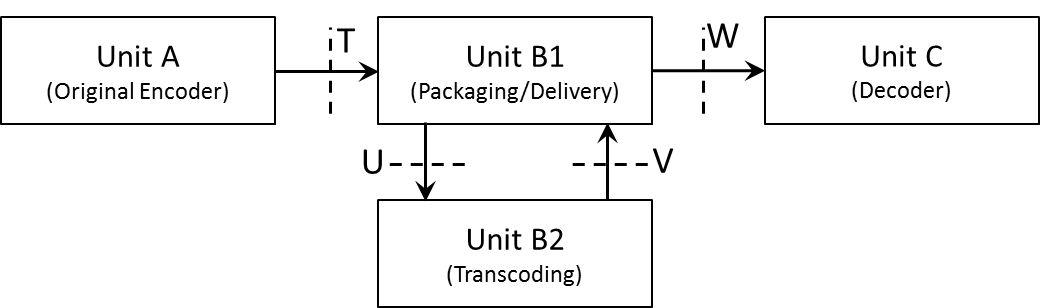


Figure 2- *Schematic view of guided transcoding system.*

Unit A is the original encoder capable of creating stream T. Unit B1 is a network entity to support the delivery of media content from Unit A to Unit C. Unit B2 is a processing unit having transcoding 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.

# Requirements

1. General Requirements
   1. NDVC shall have advantages in terms of compression, computation, bandwidth, storage efficiency over existing MPEG standards.
   2. NDVC shall have low computational resource demands in the network.
   3. NDVC shall support video coding distributed across processing units.
   4. NDVC shall support video coding with processing units that are interconnected with each other in a network.
   5. NDVC shall support transcoding for adaptive streaming in the network.
   6. NDVC should support functionalities for filtering, modifying and mixing of media content, and be aware of context.
2. Video Coding
   1. NDVC shall provide means for network-distributed video coding without affecting the compliance to existing decoding standards in the end-device.
   2. NDVC encoder shall provide means to signal NDVC metadata (e.g. side information).
3. Container Format
   1. NDVC 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. NDVC shall support identification of conformance points of each content component (e.g. elementary stream).
   3. NDVC shall support storage of multiple components for a single application.
   4. NDVC shall support delivery for applications that use common components (e.g. ISOBMFF).
   5. NDVC shall support storage of content that uses common components.
   6. NDVC shall support content component identification.
   7. NDVC shall support clock recovery (e.g. PCR)
   8. NDVC 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. NDVC shall support a format for requesting transcoding of time segments of media data for adaptive bit-rate streaming.
   10. NDVC should support dynamic configuration (e.g. merging) of content components during delivery.
4. Delivery
   1. NDVC shall support streaming.
   2. NDVC shall support file delivery.
   3. NDVC shall support progressive download.
   4. NDVC shall support delivery over IP-based connection.
   5. NDVC shall support delivery over HTTP and RTP.
   6. NDVC shall support delivery over UDP and TCP.
   7. NDVC shall support delivery using existing MPEG protocols (e.g. ISO/IEC 23008-1).
   8. NDVC shall support delivery of media using existing standardized delivery formats.
   9. NDVC shall support push-based streaming, e.g., over unidirectional or multicast channels.
   10. NDVC shall support push-based progressive download, e.g., over unidirectional or multicast channels.
   11. NDVC shall support low latency delivery (e.g. to support conversational applications, live content, etc.).
   12. NDVC shall support use of different QoS types and levels.
   13. NDVC shall support low-complexity format conversion for delivery and storage.
   14. NDVC shall support relaying received content stored on storage devices.
   15. NDVC shall support control of content relay and retransmission.
   16. NDVC shall support dynamic media processing during delivery.
   17. NDVC should support signalling messages of media service for optimal use of network resources in distributed network entities.
5. Transcoding
   1. NDVC shall support transcoding of content in domains such as temporal, spatial, quality/fidelity or view perspective.
   2. NDVC shall provide means to signal transcoding options and constraints (such as spatial and temporal resolution, etc).
   3. NDVC shall support transcoding to a wide range of devices having different capabilities and resource limitations.
   4. NDVC shall support conveying information applicable to multiple different operating points and be significantly more effective compression-wise compared to conveying multiple independent compressed bitstreams for these operating points (e.g., simulcast).
   5. NDVC shall support conveying the nominal operating point as an independent compressed bitstream, while other operating points shall be conveyed as a dependent information from the nominal operating point.
   6. NDVC shall use a transcoding process of significantly lower computational complexity than conventional transcoding (i.e. decoding followed by conventional re-encoding) without introducing degradation compared to conventional transcoding.
   7. Additional delay introduced in NDVC by the transcoding function shall be minimal and have negligible impact on overall performance.
   8. NDVC shall support delivery of any of the multiple different operating points to the end device with minimal or no bitrate overhead compared to single layer coding for that operating point.
   9. NDVC shall support just-in-time transcoding requests.
   10. NDVC should support operating points to differ in terms of image resolution, bitrate, frame rate, codec, profile/tier/level, dynamic range, colour space and chroma subsampling format.
6. Content Protection
   1. NDVC should support signalling, delivery and utilization of content using multiple protection and rights management tools.
   2. NDVC should support seamless change between content rights management schemes.
7. Delivery Environments
   1. NDVC shall support hybrid delivery environments, such as multiple transmission channels, possibly of different types.
   2. NDVC shall support multipath delivery.
   3. NDVC shall support both unidirectional and bidirectional communication environments.
   4. NDVC shall support seamless use of heterogeneous network environments including broadcast, unicast, multicast, storage, peer-to-peer, and mobile.

# Use cases

The following section contains three example transcoding applications that may benefit from network-distributed video encoding. They all involve delivery of ABR (adaptive bitrate) video, where video content is provided in several different representations. Here, the representations may differ in terms of image resolution, frame rate, bitrate, as well as codec profile/level. They may also differ in terms of colour space, chroma subsampling format, dynamic range, codec and streaming formats.

## Storage of ABR video for on-demand delivery

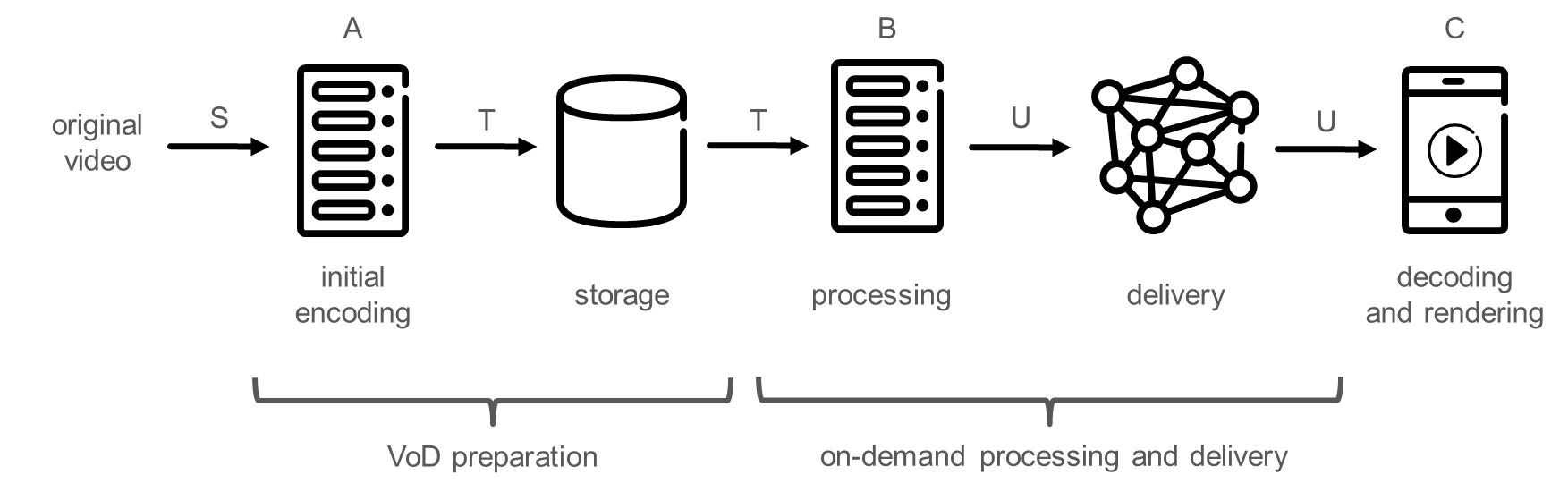


Figure 3 -*Application example: Storage of ABR video for on-demand delivery*

Figure 3 shows an example of a system for preparation and delivery of ABR video. Original video (S) is ingested into the VoD or n-PVR preparation system. The purpose of the VoD or n-PVR preparation system is to prepare the video in a way such that any specific ABR video representation can be delivered to the end user upon the user’s request, so that the end user’s equipment can decode (C) and render the video. In order to do so, the VoD or n-PVR preparation system applies an initial encoding, transforming the video into an initial stream (T), which is stored. At the time when a user requests a specific ABR representation, relevant parts of the stored stream (T) are retrieved and processed (B), generating the requested ABR video representation (U), which is then delivered to the end user’s equipment.

In order to maximize the end user quality experience, the final delivered ABR video representation (U) needs to be compressed very efficiently, i.e. with minimal or no compression performance loss compared to direct encoding. At the same time, from an economical perspective, the cost for storage and processing needs to be minimized. One commonly employed solution today (1) performs initial encoding and storage of independent ABR video representations. While that solution has very low processing requirements at B, it can be very cost-intensive in terms of storage. Due to that, an alternative just-in-time transcoding approach (2) has emerged, in which the initial encoding at A provides only a single high quality ABR representation, and lower quality ABR representations are generated by the processing step at B on-the-fly as the content is requested. While that solution is efficient in terms of storage, it poses very high processing requirements in B. Also, the transcoding operation introduces compression performance loss and thus degraded user experience.

The goals of network-distributed video encoding in this application may include:

* preserve the ABR video representation quality of independent ABR video representations storage solution (1) with an increase of processing at B to reduce the storage space,
* or to keep the advantage of storage efficiency (slight increase of storage) of the just-in-time transcoding solution (2) with a much lower processing demands at B and no impact on the perceived quality.

## Delivery of live ABR video to multiple receivers in the home

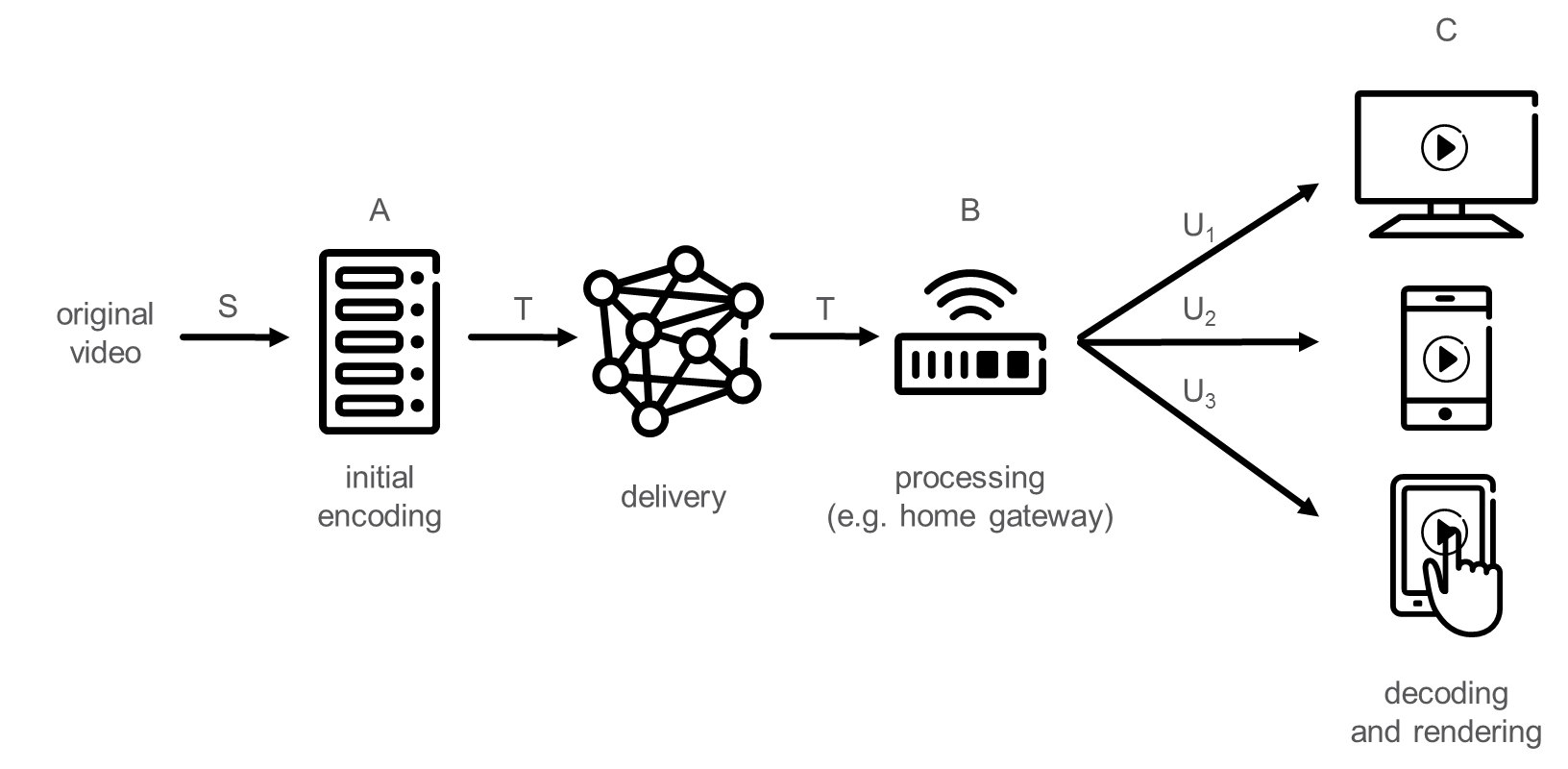


Figure 4- *Application example: Delivery of live ABR video to multiple receivers in the home*

Figure 4 shows an example of a system that delivers live ABR video to a user’s home. Within the home, several different ABR video representations are consumed by different devices. Original video (S) is ingested into the system and initially encoded (A) into a compressed stream (T), which is then delivered to the user’s home. Within the home, e.g. in the home gateway, processing (B) is applied to produce specific ABR video representations (U1, U2, U3) as requested by the different user devices. Note that the different ABR video representations (U1, U2, U3) may need to meet different device capabilities in terms of for example video resolution, codec support, profile/level support, and/or dynamic range support.

There are several potential bottlenecks within such a system. Firstly, the delivery to the home is bandwidth limited (it may be broadcast or multicast delivery). Therefore, in order to maximize the user experience, the stream that is delivered into the home (T) needs to be efficiently compressed. Secondly, the delivery within the home is bandwidth limited as well. Therefore, the streams (U) delivered within the home need to be efficiently compressed. Similar to the example in Section 5.1, (1) simulcast delivery to the home and (2) single stream delivery to the home with transcoding in the home gateway are two potential solutions, having drawbacks in terms of compression efficiency for the delivery into the home (1), and in terms of processing cost and compression efficiency for delivery within the home (2), respectively.

In this application, network-distributed video encoding tries to limit the bitrate overhead of the stream (T) delivered to the home compared to conventional transcoding (2) while operating with significantly lower processing in the home gateway (B) and provide equivalent or improved compression efficiency for delivery within the home.

## Network delivery of ABR video

### Delivery of live or on-demand ABR video inside a media delivery network

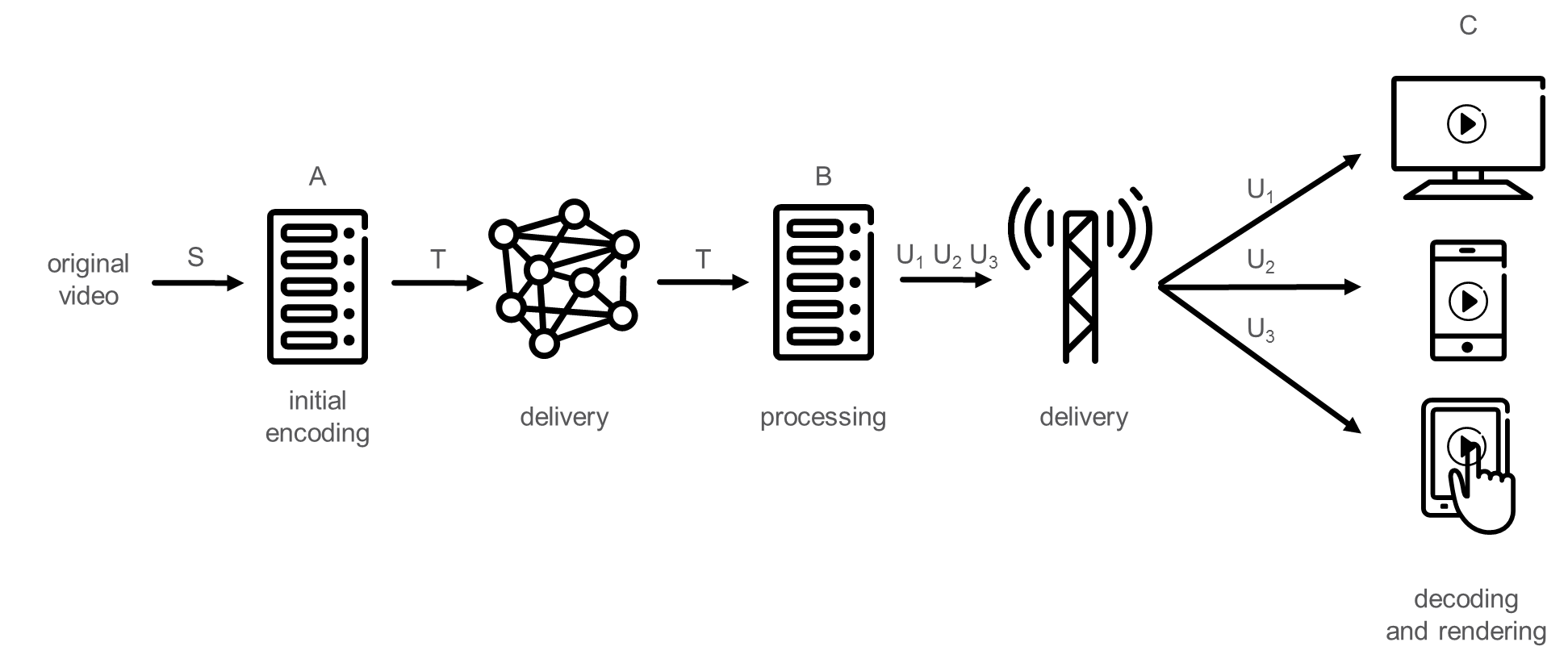


Figure 5- *Application example: Delivery of live or on-demand ABR video inside a media delivery network*

Figure 5 shows an example of a media delivery network in which video content is prepared for live or on-demand ABR video delivery. 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, at which processing is applied in order to transform the initial stream (T) into ABR video representations (U1, U2, U3) that can be delivered to the end user devices (C). Note that the different ABR video representations (U1, U2, U3) may need to meet different device capabilities in terms of e.g. resolution, codec support, profile/level support, and/or dynamic range support.

While simulcast of ABR representations is an obvious and commonly used solution when preparing the initial stream (T) in this application example, there may be cases in which the bandwidth within the media delivery network is constrained and therefore more efficient compression of the initial stream (T) is desirable. A possible solution is to deliver only a high quality video representation within the media delivery network and generate the lower quality representations closer to the media network edge through transcoding (B). However, that solution may be economically inefficient due to the high processing cost associated with the transcoding. Moreover, the transcoding introduces compression performance loss, which affects the last-mile delivery to the end user and thus leads to degraded user experience.

In this application, network-distributed video coding can offer a much lower processing cost than conventional transcoding for a comparable compression performance for the last-mile delivery.

### Delivery of live or on-demand ABR video from a production facility to a distribution facility

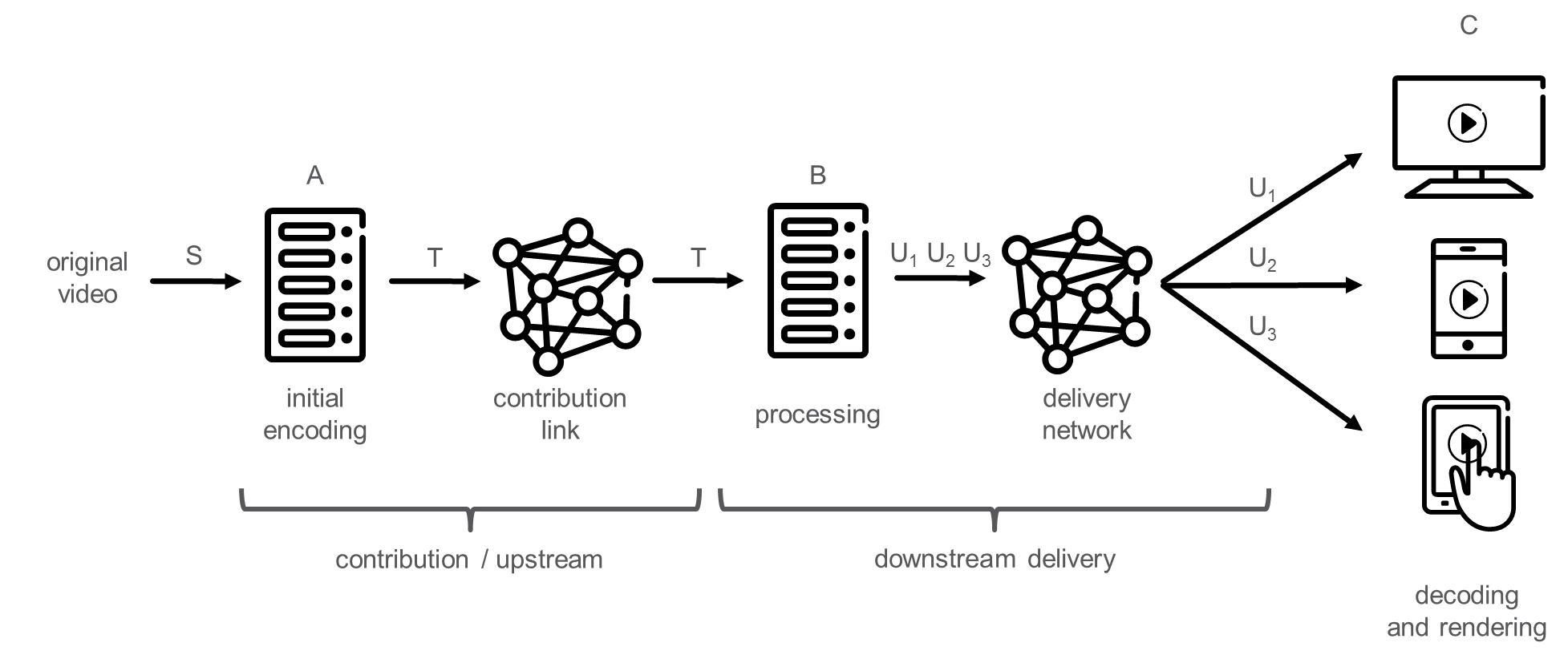


Figure 6- *Application example: Delivery of live or on-demand ABR video from a production facility to a distribution facility*

Figure 6 shows an example of a media production and distribution system in which video content is prepared for live or on-demand ABR video delivery. Original video (S) is ingested at a production facility and initially encoded (A) into an initial stream (T). The initial stream (T) is uploaded over a contribution link to a cloud-based distribution facility, at which processing is applied in order to transform the initial stream (T) into ABR video representations (U1, U2, U3) that can be delivered over a delivery network to the end user devices (C). Note that the different ABR video representations (U1, U2, U3) may need to meet different device capabilities in terms of e.g. codec support, profile/level support, and/or dynamic range support.

A first existing technical solution for this application example is to deliver a high quality video over the contribution link, and to generate the lower quality representations at the distribution facility through transcoding (B). In order to minimize transcoding-induced compression efficiency losses, the bit rate for coding the high quality video that is sent over the contribution link may need to be very high, which may be conflicting with technical or economical bandwidth constraints on the contribution link. A second existing technical solution for the application example is that the different ABR video representations (U1, U2, U3) are generated already in the initial encoding step, and uploaded as simulcast over the contribution link. However, similar as in the first technical solution, since the bandwidth of the contribution link may be constrained, more efficient compression of the initial stream (T) is desirable.

# Timeline

A preferred timeline is to finalize specification(s) for supporting network-distributed video coding no later than finalization of next generation video codec. It should be considered to address specification(s) targeting existing video coding standards such as HEVC and AVC with a shorter timeline.

# Background information

This section provides informative overview of some of the developments in the distributed network infrastructure, particularly cloud based applications which may be considered in NDVC.

***Cloud Computing*:** Cloud service providers such as Amazon, Google Cloud and Microsoft Azure have grown and created large revenues. But perhaps more importantly, they have enabled significant innovation in many start-ups and industries that evolve their services on top of this cloud infrastructure. The video industry is no exception and OTT platforms such as Netflix, BBC iPlayer etc. have evolved on cloud based infrastructure exploiting the Infrastructure as a Service (IaaS) paradigm. One of the main advantages of this approach is that it allows services to scale with the number of active users and resource demands. Another important development is the emergence of an open source cloud operating system: OpenStack. This OS enables private clouds to be developed with functionalities similar to commercial public cloud services. This enables broadcasters, Telco’s and hosting service providers to soon be able to provide elastic cloud services such as Amazon EC2 based on OpenStack. This could potentially increase the use of cloud computing over the entire IT and Telecom industry dramatically.

***Virtualization and Containerization:***Many applications in the cloud run on virtual machines that provide operating systems on top of the host hardware. A trend in virtualization is the move towards smaller virtual machines (possibly using stripped operating system with redundant systems functions and libraries removed) and smaller containers (that do not add a full OS layer but still provide isolation using name space separation). This enable applications to be developed at a lower cost and finer granularity (as containers/VM’s can be added or increased in size).

***Network Function Virtualization*** is using virtualization techniques to implement network functions such as routing, DNS, switching etc. Compared to most virtualized cloud applications, in Network Function Virtualization hardware acceleration and/or high performance processing is rather critical. Transcoding is a critical function that benefits from high performance processing and hardware acceleration. Also, transcoding could be considered as a critical network function for many multimedia services, i.e. as a VSNF (Video Streaming Network Function).

***Micro services.***Many of the current applications running in the cloud uses many VM’s and containers running in a distributed fashion that are communicating with each other providing a higher level service. This architectural approach where applications are split into functional blocks or services that run on separate virtualized instances/hardwares and communicate via HTTP or other protocols is sometimes called *micro services*. This architectural paradigm is becoming more important and dominant to enable more robust, resource efficient and massively distributed applications in the cloud. Figure 7 shows how such applications could be composed in practice. Applications based on this architecture can be deployed using NFV, virtualization and cloud computing.

***Edge Computing*** is another evolution in cloud computing and networking. It enables services to be run at different locations and move closer to the users, for example to reduce latency and increase bandwidth (e.g. by running services near the Radio Access Network). In such scenarios, the interface between the edge and core becomes another topic of study. An example considered for network-distributed video coding is the interface between the main encoder and the edge transcoder.

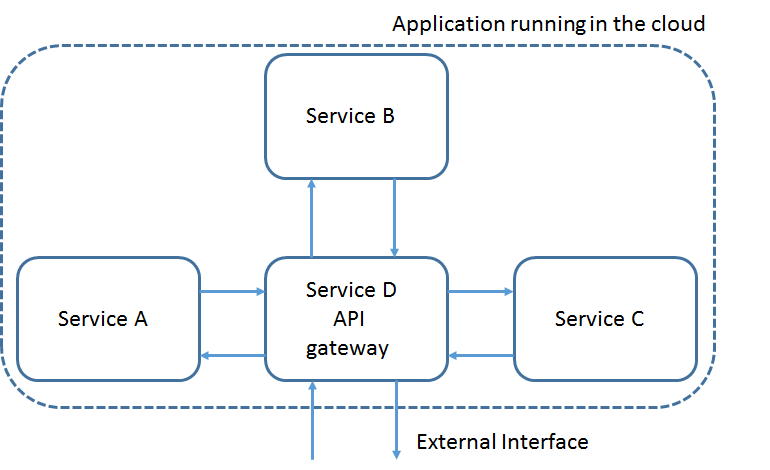


Figure 7 - *Cloud application composed of different services (micro services)*

To summarize, distributed video coding components will be embedded in these kinds of cloud environments and should take their specific requirements into account. For MPEG DASH, transcoding of media segments will be an important NDVC application.