ICNRG
Internet Draft
Intended status: Informational
Expires: August 31, 2015

S. Lederer
D. Posch
C. Timmerer
Alpen-Adria University Klagenfurt
C. Westphal, Ed.
A. Azgin
S. Liu
Huawei
C. Mueller
Bitmovin
A.Detti
University of Rome Tor Vergata
D. Corujo
University of Aveiro

February 23, 2015

Adaptive Video Streaming over ICN
draft-irtf-icnrg-videostreaming-03.txt

Status of this Memo

Copyright Notice

Abstract

This document considers the consequences of moving the underlying network architecture to an Information-Centric Network (ICN) architecture on video distribution. As most of the traffic in future networks is expected to be video, we consider how to modify the existing video streaming mechanisms. Several important topics related to video distribution over ICN are presented, covering a wide range of scenarios: we look at how to evolve DASH to work over ICN, and leverage the recent ISO/IEC MPEG Dynamic Adaptive Streaming over HTTP (DASH) standard; we consider layered encoding over ICN; P2P mechanisms introduce distinct requirements for video and we look at how to adapt PPSP for ICN; IPTV adds delay constraints, and this will create more stringent requirements over ICN as well. As part of the discussion on video, we discuss DRMs in ICN. Finally, in addition to consider how existing mechanisms would be impacted by

ICN, this document lists some research issues to design ICN specific video streaming mechanisms.

Table of Contents

1. Introduction

The unprecedented growth of video traffic has triggered a rethinking of how content is distributed, both in terms of the underlying Internet architecture and in terms of the streaming mechanisms to deliver video objects.

In particular, the IRTF ICN working group has been chartered to study new architectures centered upon information; the main contributor to Internet traffic (and information dissemination) is video, and this is expected to stay the same in the short- to mid-term future. If ICN is expected to become prominent, it will have to support video streaming efficiently.

As such, it is necessary to discuss along two directions:

- Can the current video streaming mechanisms be leveraged and adapted to an ICN architecture?

- Can (and should) new, ICN-specific video streaming mechanisms be designed to fully take advantage of the new abstractions exposed by the ICN architecture?

This document intends to focus on the first question, in an attempt to define the use cases for video streaming and some requirements.

This document focuses on a few scenarios, namely Netflix-like video streaming, peer-to-peer video sharing and IPTV, and identifies how the existing protocols can be adapted to an ICN architecture. In doing so, it also identifies the main issues with these protocols in this ICN context.

Some documents have started to consider the ICN-specific requirements of dynamic adaptive streaming [2][3][4][6].

In this document, we give a brief overview of the existing solutions for the selected scenarios. We then consider the interactions of such existing mechanisms with the ICN architecture and list some of the interactions any video streaming mechanism will have to consider. We then identify some areas for future research.

2. Conventions used in this document

In examples, "C:" and "S:" indicate lines sent by the client and server respectively.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC-2119 [RFC2119].

In this document, these words will appear with that interpretation only when in ALL CAPS. Lower case uses of these words are not to be interpreted as carrying RFC-2119 significance.

In this document, the characters ">>" preceding an indented line(s) indicates a compliance requirement statement using the key words listed above. This convention aids reviewers in quickly identifying or finding the explicit compliance requirements of this RFC.

3. Use case scenarios for ICN and Video Streaming

For ICN specific descriptions, we refer to the other working group documents. For our purpose, we assume here that ICN means an architecture where content is retrieved by name and with no binding of content to a specific network location.

The consumption of multimedia content comes along with timing requirements for the delivery of the content, for both, live and on-demand consumption. Additionally, real-time use cases such as audio-/video conferencing [7], game streaming, etc., come along with more strict timing requirements. Long startup delays, buffering periods or poor quality, etc., should be avoided to achieve a good Quality of Experience (QoE) to the consumer of the content. Of course, these requirements are heavily influenced by routing decisions and caching, which are central parts of ICN and which have to be considered when streaming video in such infrastructures.

Due to this range of requirements, we find it useful to narrow the focus on four scenarios (more can be included later):

- a video delivery architecture similar to that if itune, where the whole file is being downloaded to the client and can be replayed there multiple times;
- a video streaming architecture for playing back movies; this is relevant for the naming and caching aspects of ICN, as well as the interaction with the rate adaptation mechanism necessary to deliver the best QoE to the end-user;
- a peer-to-peer architecture for sharing videos; this introduces more stringent routing requirements in terms of locating copies of the content, as the location of the peers evolves and peers join and leave the swarm they use to exchange video chunks;
- IPTV; this introduces requirements for multicasting and adds stronger delay constraints.

Other scenarios, such as video-conferencing and real-time video communications are not explicitly discussed in this document, while they are in scope. Also, events of mass-media distribution, such as a large crowd in a live event, are also adding new requirements to be included in later version.

We discuss how the current state-of-the-art protocols in an IP context can be modified for the ICN architecture. The remainder of this document is organized as follows. In the next section, we consider video download. Then in Section 5, we briefly describe DASH [1], and Layered Encoding (MDC, SVC). P2P is the focus of Section 6, where we describe PPSP. Section 7 highlights the requirements of IPTV, while Section 8 describes the issues of DRM. Section 9 lists some research issues to be solved for ICN-specific video delivery mechanisms.

This research items include videoconferencing and real-time video communications, which will be detailed more in future versions of this document; as well as the mass distribution of content at live large-scale events (stadium, concert hall, etc) for which there is no clearly adopted existing protocol.

4. Video download

Video download, namely the fetching of a video file from a server or a cache down to the user’s local storage, is a natural application of ICN. It should be supported natively without requiring any specific considerations.

This is supported now by a host of protocols (say, scp, ftp, or over http), which would need to be replaced by the protocols to retrieve content in ICNs.

However, current mechanisms are built atop existing transport protocol. Some ICN proposals (say, CCN or NDN for instance) attempt to leverage the work down upon these transport protocol and it has been proposed to use mechanisms such as the TCP congestion window (and the associated Adaptive Increase, Multiplicative Decrease - AIMD) to decide how many object requests ("interests" in CCN/NDN terminology) should be in flight at any point in time.

It should be noted that ICN intrinsically supports different transport mechanisms, which could achieve better performance than TCP, as they subsume TCP into a special case. For instance, one could imagine a link-by-link transport coupled with caching. This is enabled by the ICN architecture, and would facilitate the point-to-point download of video files.

5. Video streaming and ICN

5.1. Introduction to client-driven streaming and DASH

Media streaming over the hypertext transfer protocol (HTTP) and in a further consequence streaming over the transmission control protocol (TCP) has become omnipresent in today’s Internet. Content providers such as Netflix, Hulu, and Vudu do not deploy their own streaming equipment but use the existing Internet infrastructure as it is and they simply deploy their own services over the top (OTT). This streaming approach works surprisingly well without any particular support from the underlying network due to the use of efficient video compression, content delivery networks (CDNs), and adaptive video players. Earlier video streaming research mostly recommended to use the user datagram protocol (UDP) combined with the real time transport protocol (RTP). It assumed it would not be possible to transfer multimedia data smoothly with TCP, because of its throughput variations and large retransmission delays. This point of view has significantly evolved today. HTTP streaming, and especially its most simple form known as progressive download, has become very popular over the past few years because it has some major benefits compared to RTP streaming. As a consequence of the consistent use of HTTP for this streaming method, the existing Internet infrastructure, consisting of proxies, caches and CDNs, could be used. Originally, this architecture was designed to support best effort delivery of files and not real time transport of multimedia data. Nevertheless, real time streaming based on HTTP could also take advantage of this architecture, in comparison to RTP, which could not leverage any of the aforementioned components. Another benefit that results from the use of HTTP is that the media stream could easily pass firewalls or network address translation (NAT) gateways, which was definitely a key for the success of HTTP

streaming. However, HTTP streaming is not the holy grail of streaming as it also introduces some drawbacks compared to RTP. Nevertheless, in an ICN-based video streaming architecture these aspects also have to be considered.

The basic concept of DASH [1] is to use segments of media content, which can be encoded at different resolutions, bitrates, etc., as so-called representations. These segments are served by conventional HTTP Web servers and can be addressed via HTTP GET requests from the client. As a consequence, the streaming system is pull-based and the entire streaming logic is located on the client, which makes it scalable, and allows to adapt the media stream to the client's capabilities.

In addition to this, the content can be distributed using conventional CDNs and their HTTP infrastructure, which also scales very well. In order to specify the relationship between the contents' media segments and the associated bitrate, resolution, and timeline, the Media Presentation Description (MPD) is used, which is a XML document. The MPD refers to the available media segments using HTTP URLs, which can be used by the client for retrieving them.

5.2. Layered Encoding

Another approach for video streaming consist in using layered encoding. Namely, scalable video coding formats the video stream into different layers: a base layer which can be decoded to provide the lowest bit rate for the specific stream, and enhancement layers which can be transmitted separately if network conditions allow. The higher layers offer higher resolutions and enhancement of the video quality, while the layered approach allows to adapt to the network conditions. This is used in MPEG-4 scalable profile or H.263+. H264SVC is available, but not much deployed. JPEG2000 has a wavelet transform approach for layered encoding, but has not been deployed much either.

It is not clear if the layered approach is fine-grained enough for rate control.

5.3. Interactions of Video Streaming with ICN

5.3.1. Interaction of DASH and ICN

Video streaming, and DASH in particular, have been designed with goals that are aligned with that of most ICN proposals. Namely, it is a client-based mechanism, which requests items (in this case, chunks of a video stream) by name.

ICN and MPEG-DASH [1] have several elements in common:

- the client-initiated pull approach;
- the content being dealt with in pieces (or chunks);
- the support of efficient replication and distribution of content pieces within the network;
- the scalable, session-free nature of the exchange between the client and the server at the streaming layer: the client is free to request any chunk from any location;
- the support for potentially multiple sources.

As ICN is a promising candidate for the Future Internet (FI) architecture, it is useful to investigate its suitability in combination with multimedia streaming standards like MPEG-DASH. In this context, the purpose of this section is to present the usage of ICN instead of HTTP in MPEG-DASH

However, there are some issues that arise from using a dynamic rate adaptation mechanism in an ICN architecture:

o Naming of the data in DASH does not necessarily follow the ICN convention of any of the ICN proposals. Several chunks of the same video stream might currently go by different names that for instance do not share a common prefix. There is a need to harmonize the naming of the chunks in DASH with the naming conventions of the ICN. The naming convention of using a filename/time/encoding format could for instance be made compatible with the convention of CCN.

o While chunks can be retrieved from any server, the rate adaptation mechanism attempts to estimate the available network bandwidth so as to select the proper playback rate and keep its playback buffer at the proper level. Therefore, there is a need to either include some location semantics in the data chunks so as to properly assess the throughput to a specific location; or to design a different mechanism to evaluate the available network bandwidth.

o The typical issue of access control and accounting happens in this context, where chunks can be cached in the network outside of the administrative control of the content publisher. It might be a requirement from the owner of the video stream that access to these data chunks needs to be accounted/billed/monitored.

- Dynamic streaming multiplies the representations of a given video stream, therefore diminishing the effectiveness of caching: namely, to get a hit for a chunk in the cache, it has to be for the same format and encoding values. Alternatively, to get the same hit rate as for a stream using a single encoding, the cache size must be scaled up to include all the possible representations.

- Caching introduces oscillatory dynamics as it may modify the estimation of the available bandwidth between the end user and the repository where it is getting the chunks from. For instance, if an edge cache holds a low resolution representation near the user, the user getting this low resolution chunks will observe a good performance, and will then request higher resolution chunks. If those are hosted on a server with poor performance, then the client would have to switch back to the low representation. This oscillation may be detrimental to the perceived QoE of the user.

- The ICN transport mechanism needs to be compatible to some extent with DASH. To take a CCN example, the rate at which interests are issued should be such that the chunks received in return arrive fast enough and with the proper encoding to keep the playback buffer above some threshold.

- The usage of multiple network interfaces is possible in ICN, enabling a seamless handover between them. For the combination with DASH, an intelligent strategy which should focus on traffic load balancing between the available links may be necessary. This would increase the effective media throughput of DASH by leveraging the combined available bandwidth of all links, however, it could potentially lead to high variations of the media throughput.

- DASH does not define how the MPD is retrieved; hence, this is compatible with CCN. However, the current profiles defined within MPEG-DASH require the MPD to contain HTTP-URLs (incl. http and https URI schemes) to identify segments. To enable a more integrated approach as described in this document, an additional profile for DASH over CCN has to be defined, enabling ICN/CCN-based URIs to identify and request the media segments.

We describe in Section 5.4 a potential implementation of a dynamic adaptive video stream over ICN, based upon DASH and CCN [5].

5.3.2. Interaction of ICN with Layered Encoding

Issues of interest to an Information-Centric network architecture in the context of layered video streaming include:

- Caching of the multiple layers. The caching priority should go to the base layer, and defining caching policy to decide when to cache enhancement layers;
- Synchronization of multiple content streams, as the multiple layers may come from different sources in the network (for instance, the base layer might be cached locally while the enhancement layers may be stored in the origin server);
- Naming of the different layers: when the client requests an object, the request can be satisfied with the base layer alone, aggregated with enhancement layers. Should one request be sufficient to provide different streams? In a CCN architecture for instance, this would violate a one interest-one data packet principle and the client would need to specify each layer it would like to receive. In a Pub/Sub architecture, the rendezvous point would have to make a decision as to which layers (or which pointer to which layer's location) to return.

5.4. Possible Integration of Video streaming and ICN architecture

5.4.1. DASH over CCN

DASH is intended to enable adaptive streaming, i.e., each content piece can be provided in different qualities, formats, languages, etc., to cope with the diversity of todays' networks and devices. As this is an important requirement for Future Internet proposals like CCN, the combination of those two technologies seems to be obvious. Since those two proposals are located at different protocol layers - DASH at the application and CCN at the network layer - they can be combined very efficiently to leverage the advantages of both and potentially eliminate existing disadvantages. As CCN is not based on classical host-to-host connections, it is possible to consume content from different origin nodes as well as over different network links in parallel, which can be seen as an intrinsic error resilience feature w.r.t. the network. This is a useful feature of CCN for adaptive multimedia streaming within mobile environments since most mobile devices are equipped with multiple network links like 3G and WiFi. CCN offers this functionality out of the box which is beneficial when used for DASH-based services. In particular, it is possible to enable adaptive video streaming handling both bandwidth and network link changes. That is, CCN handles the network link decision and DASH is implemented on top of CCN to adapt the video stream to the available bandwidth.

In principle, there are two options to integrate DASH and CCN: a proxy service acting as a broker between HTTP and CCN as proposed in [6], and the DASH client implementing a native CCN interface. The former transforms an HTTP request to a corresponding interest packet as well as a data packet back to an HTTP response, including reliable transport as offered by TCP. This may be a good compromise to implement CCN in a managed network and to support legacy devices. As such a proxy is already described in [6] this draft focuses on a more integrated approach, aiming at fully exploiting the potential of a CCN DASH Client. That is, we describe a native CCN interface within the DASH client, which adopts a CCN naming scheme (CCN URIs) to denote segments in the Media Presentation Description (MPD). In this architecture, only the network access component on the client has to be modified and the segment URIs within MPD have to be updated according to the CCN naming scheme.

Initially, the DASH client retrieves the MPD containing the CCN URIs of the content representations including the media segments. The naming scheme of the segments may reflect intrinsic features of CCN like versioning and segmentation support. Such segmentation support is already compulsory for multimedia streaming in CCN and, thus, can also be leveraged for DASH-based streaming over CCN. The CCN versioning can be adopted in a further step to signal different representations of the DASH-based content, which enables an implicit adaptation of the requested content to the clients' bandwidth conditions. That is, the interest packet already provides the desired characteristics of a segment (such as bit rate, resolution, etc.) within the content name (or potentially within parameters defined as extra types in the packet formats). Additionally, if bandwidth conditions of the corresponding interfaces or routing paths allow so, DASH media segments could be aggregated automatically by the CCN nodes, which reduces the amount of interest packets needed to request the content. However, such approaches need further research, specifically in terms of additional intelligence and processing power needed at the CCN nodes.

After requesting the MPD, the DASH client will start to request particular segments. Therefore, CCN interest packets are generated by the CCN access component and forwarded to the available interfaces. Within the CCN, these interest packets leverage the efficient interest aggregation for, e.g., popular content, as well as the implicit multicast support. Finally, the interest packets are satisfied by the corresponding data packets containing the video segment data, which are stored on the origin server or any CCN node, respectively. With an increasing popularity of the content, it will be distributed across the network resulting in lower transmission

delays and reduced bandwidth requirements for origin servers and content providers respectively.

With the extensive usage of in-network caching, new drawbacks are introduced since the streaming logic is located at the client, i.e., clients are not aware of each other and the network infrastructure and cache states. Furthermore, negative effects are introduced when multiple clients are competing for a bottleneck and when caching is influencing this bandwidth competition. As mentioned above, the clients request individual portions of the content based on available bandwidth which is calculated using throughput estimations. This uncontrolled distribution of the content influences the adaptation process of adaptive streaming clients. The impact of this falsified throughput estimation could be tremendous and leads to a wrong adaptation decision which may impact the Quality of Experience (QoE) at the client, as shown in [8]. In ICN, the client does not have the knowledge from which source the requested content is actually served or how many origin servers of the content are available, as this is transparent and depends on the name-based routing. This introduces the challenge that the adaptation logic of the adaptive streaming client is not aware of the event when the ICN routing decides to switch to a different origin server or content is coming through a different link/interface. As most algorithms implementing the adaption logic are using bandwidth measurements and related heuristics, the adaptation decisions are no longer valid when changing origin servers (or links) and potentially cause playback interruptions and, consequently, stalling. Additionally, ICN supports the usage of multiple interfaces and a seamless handover between them, which again comes together with bandwidth changes, e.g., switching between fixed and wireless, 3G/4G and WiFi networks, etc. Considering these characteristics of ICN, adaptation algorithms merely based on bandwidth measurements are not appropriate anymore, as potentially each segment can be transferred from another ICN node or interface, all with different bandwidth condition. Thus, adaptation algorithms taking into account these intrinsic characteristics of ICN are preferred over algorithms based on mere bandwidth measurements.

5.4.2. Testbed, Open Source Tools, and Dataset

For the evaluations of DASH over CCN, a testbed with open source tools and datasets is provided in [9]. In particular, it provides two client player implementations, (i) a libdash extension for DASH over CCN and (ii) a VLC plugin implementing DASH over CCN. For both implementations the CCNx implementation has been used as a basis.

The general architecture of libdash is organized in modules, so that the library implements a MPD parser and an extensible connection manager. The library provides object-oriented interfaces for these modules to access the MPD and the downloadable segments. These components are extended to support DASH over CCN and available in a separate development branch of the github project available at http://www.github.com/bitmovin/libdash. libdash comes together with a fully featured DASH player with a QT-based frontend, demonstrating the usage of libdash and providing a scientific evaluation platform. As an alternative, patches for the DASH plugin of the VLC player are provided. These patches can be applied to the latest source code checkout of VLC resulting in a DASH over CCN-enabled VLC player.

Finally, a DASH over CCN dataset is provided in form of a CCNx repository. It includes 15 different quality representation of the well-known Big Buck Bunny Movie, ranging from 100 kbps up to 4500 kbps. The content is split into segments of two seconds, and described by an associated MPD using the presented naming scheme in Section 4.1. This repository can be downloaded from [9], and is also provided by a public accessible CCNx node. Associated routing commands for the CCNx namespaces of the content are provided via scripts coming together with the dataset and can be used as a public testbed.

6. P2P video distribution and ICN

Another form of distributing content - and video in particular- which ICNs need to support is Peer-to-Peer distribution (P2P). We see now how an existing protocol such as PPSP can be modified to work in an ICN environment.

6.1. Introduction to PPSP

P2P video Streaming (PPS) is a popular approach to redistribute live media over Internet. The proposed P2PVS solutions can be roughly classified in two classes:

- Push/Tree based

- Pull/Mesh based

The Push/Tree based solution creates an overlay network among peers that has a tree shape. Using a progressive encoding (e.g. Multiple Description Coding or H.264 Scalable Video Coding), multiple trees could be set up to support video rate adaptation. On each tree an enhancement stream is sent. The more the number of stream received,

the higher the video quality. A peer control video rate by fetching or not the streams delivered on the distribution trees.

The Pull/Mesh based solution is inspired by the BitTorrent file sharing mechanism. A Tracker collects information about the state of the swarm (i.e. set of participating peers). A peer forms a mesh overlay network with a subset of peers, and exchange data with them. A peer announces what data items it disposes and requests missing data items that are announced by connected peers. In case of live streaming, the involved data set includes only a recent window of data items published by the source. Also in this case, the use of a progressive encoding can be exploited for video rate adaptation.

Pull/Mesh based P2PVS solutions are the more promising candidate for the ICN deployment, since most of ICN approach provides a pull-based API [5][10][11][12]. In addition, Pull/Mesh based P2PVS are more robust than Push/Tree based one [13] and the Peer to Peer Streaming Protocol (PPSP) working group [14] is also proposing a Pull/Mesh based solution.

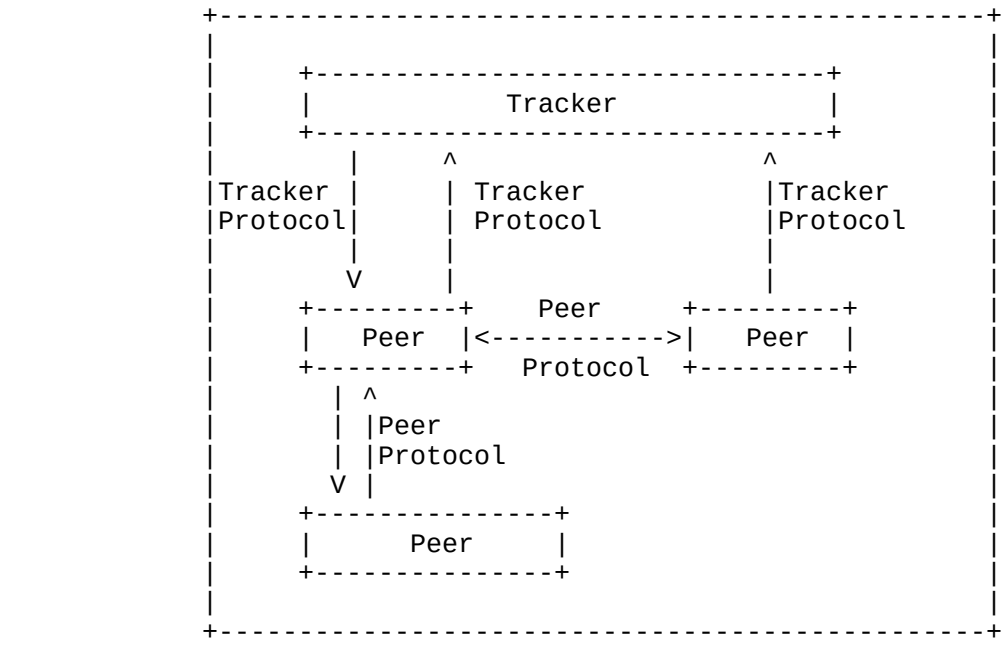

Figure 1: PPSP System Architecture (source [RFC6972])

Figure 1 reports the PPSP architecture presented in [RFC6972]. PEERs announce and share video chunks and a TRACKER maintains a list of PEERs participating in a specific audio/video channel or in the distribution of a streaming file. The tracker functionality may be centralized in a server or distributed over the PEERs. PPSP standardize the Peer and Tracker Protocols, which can run directly over UDP or TCP.

This document discusses some preliminary concepts about the deployment of PPSP on top of an ICN that exposes a pull-based API, meanwhile considering the impact of MPEG DASH streaming format.

6.2. PPSP over ICN: deployment concepts

6.2.1. PPSP short background

PPSP specifies peer protocol (PPSPP) [15] and tracker protocol (PPSP-TP)[16].

Some of the operations carried out by the tracker protocol are the followings. When a peer wishes to join the streaming session it contacts the Tracker (CONNECT message), obtains a PEER_ID and a list of PEER_IDs (and IP addresses) of other peers that are participating to the SWARM and that the tracker has singled out for the requesting peer (this may be a subset of the all peers of the SWARM). In addition to this join operation, a peer may contact the tracker to request to renew the list of participating peers (FIND message), to periodically update its status to the tracker (STAT_REPORT message), etc.

Some of the operations carried out by the peer protocol are the following. Using the list of peers delivered by the tracker, a peer establishes a session with them (HANDSHAKE message). A peer periodically announces to neighboring peers which chunks it has available for download (HAVE message). Using these announcements, a peer requests missing chunks from neighboring peers (REQUEST messages), which will send back them (DATA message).

6.2.2. From PPSP messages to ICN named-data

An ICN provides users with data items exposed by names. The bundle name and data item is usually referred as named-data, named-content, etc. To transfer PPSP messages though an ICN the messages should be be wrapped as named-data items, and receivers should request them by name.

A PPSP entity receives messages from peers and/or tracker. Some operations require gathering the messages generated by another specific host (peer or tracker). For instance, if a peer A wishes to gain information about video chunks available from peer B, the former shall fetch the PPSP HAVE messages specifically generated by the later. We refer to these kinds of named-data as "located-named-data", since they should be gathered from a specific location (e.g. peer B).

For other PPSP operations, like to fetch a DATA message (i.e. a video chunk), what it is relevant for a peer is just to receive the requested content, independently from who is the endpoint that generate the data. We refer this information with the generic term "named-data".

The naming scheme differentiates named-data and located-named-data items. In case of named-data, the naming scheme only includes a content identifier (e.g. the name of the video chunk), without any prefix identifying who provides the content. For instance, a DATA message containing the video chunk n. 1 may be named as "ccnx:/swarmID/chunk/chunkID", where swarmID is a unique identifier of the streaming session, "chunk" is a keyword and chunkID is the chunk identifier (e.g. a integer number).

In case of located-named-data, the naming scheme includes a location-prefix, which uniquely identifies the host generating the data item. This prefix may be the PEER_ID in case the host was a peer or a tracker identifier in case the host was the tracker. For instance, a HAVE message generated by a peer B may be named as "ccnx:/swarmID/peer/PEER_ID/HAVE", where "peer" is a keyword, PEER_ID_B is the identifier of peer B and HAVE is a keyword.

6.2.3. Support of PPSP interaction through a pull-based ICN API

The PPSP procedures are based both on pull and push interactions. For instance, the distribution of chunks availability can be classified as a push-based operation, since a peer sends an "unsolicited" information (HAVE message) to neighboring peers. Conversely the procedure used to receive video chunks can be classified as pull-based, since it is supported by a request/response interaction (i.e. REQUEST, DATA messages).

As we said, we refer to an ICN architecture which provides a pull-based API. Accordingly, the mapping of PPSP pull-based procedure is quite simple. For instance, using the CCN architecture [5] a PPSP

DATA message may be carried by a CCN Data message and a REQUEST message can transferred by a CCN Interest.

Conversely, the support of push-based PPSP operations may be more difficult. We need of an adaptation functionality that carries out a push-based operation using the underlying pull-based service primitives. For instance, a possible approach is to use the request/response (i.e. Interest/Data) four ways handshakes proposed in [7]. Another possibility is that receivers periodically send out request messages of the named-data that neighbors will push and, when available, sender inserts the pushed data within a response message.

6.2.4. Abstract layering for PPSP over ICN

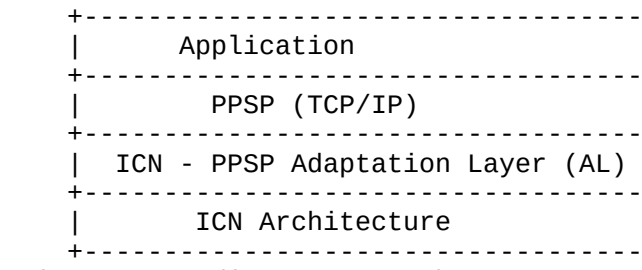

Figure 2: Mediator approach

Figure 2 provides a possible abstract layering for PPSP over ICN. The Adaptation Layer acts as a mediator (proxy) between legacy PPSP entities based on TCP/IP and the ICN architecture. In facts, the role the mediator is to use ICN to transfer PPSP legacy messages.

This approach makes possible to merely reuse TCP/IP P2P applications whose software includes also PPSP functionality. This "all-in-one" development approach may be rather common since the PPSP-Application interface is not going to be specified. Moreover, if the Operating System will provide libraries that expose a PPSP API, these will be initially based on a underlying TCP/IP API. Also in this case, the mediator approach would make possible to easily reuse both the PPSP libraries and the Application on top of an ICN.

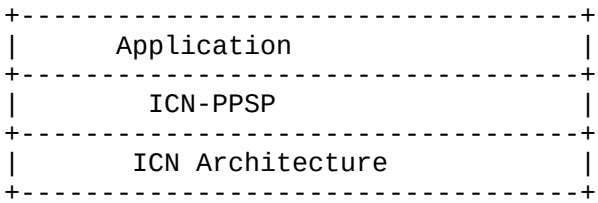

Figure 3: Clean-slate approach

Figure 3 sketches a clean-slate layering approach in which the application directly includes or interacts with a PPSP version based on ICN. Likely such a PPSP_ICN integration could yield a simplier development, also because it does not require implementing a TCP/IP to ICN translation as in the Mediator approach. However, the clean-slate approach requires developing the application (in case of embedded PPSP functionality) or the PPSP library from scratch, without exploiting what might already exist for TCP/IP.

Overall, the Mediator approach may be considered as the first step of a migration path towards ICN native PPSP applications.

6.2.5. PPSP interaction with the ICN routing plane

Upon the ICN API a user (peer) requests a content and the ICN sends it back. The content is gathered by the ICN from any source, which could be the closest peer that disposes of the named-data item, an in-network cache, etc. Actually, "where" to gather the content is controlled by an underlying ICN routing plane, which sets up the ICN forwarding tables (e.g. CCN FIB [5]).

A cross-layer interaction between the ICN routing plane and the PPSP may be required to support a PPSP session. Indeed, ICN shall forward request messages (e.g. CCN Interest) towards the proper peer that can handle them. Depending on the layering approach, this cross-layer interaction is controlled either by the Adaptation Layer or by the ICN-PPSP. For example, if a peer A receives a HAVE message indicating that peer B disposes of the video chunk named "ccnx:/swarmID/chunk/chunkID", then former should insert in its ICN forwarding table an entry for the prefix "ccnx:/swarmID/chunk/chunkID" whose next hop locator (e.g. IP address) is the network address of peer B [17].

6.2.6. ICN deployment for PPSP

The ICN functionality that supports a PPSP session may be "isolated" or "integrated" with the one of a public ICN.

In the isolated case, a PPSP session is supported by an instance of an ICN (e.g. deployed on top of IP), whose functionalities operate only on the limited set of nodes participating to the swarm, i.e. peers and the tracker. This approach resembles the one followed by current P2P application, which usually form an overlay network among peers of a P2P application. And intermediate public IP routers do not carry out P2P functionalities.

In the integrated case, the nodes of a public ICN may be involved in the forwarding and in-network caching procedures. In doing so, the swarm may benefit from the presence of in-network caches so limiting uplink traffic on peers and inter-domain traffic too. These are distinctive advantages of using PPSP over a public ICN, rather than over TCP/IP. In addition, such advantages aren't likely manifested in the case of isolated deployment.

However, the possible interaction between the PPSP and the routing layer of a public ICN may be dramatic, both in terms of explosion of the forwarding tables and in terms of security. These issues specifically take place for those ICN architectures for which the name resolution (i.e. name to next-hop) occurs en-route, like the CCN architecture.

For instance, using the CCN architecture, to fetch a named-data item offered by a peer A the on-path public ICN entities have to route the request messages towards the peer A. This implies that the ICN forwarding tables of public ICN nodes may contain many entries, e.g. one entry per video chunk, and these entries are difficult to be aggregated since peers avail sparse parts of a big content, whose names have a same prefix (e.g. "ccnx:/swarmID"). Another possibility is to wrap all PPSP messages into a located-named-data. In this case the forwarding tables should contain "only" the PEER_ID prefixes (e.g. "ccnx:/swarmID/peer/PEER_ID"), so scaling down the number of entries from number of chunks to number of peers. However, in this case the ICN mechanisms recognize a same video chunk offered by different peers as different contents, so vanishing caching and multicasting ICN benefits. Moreover, in any case routing entries should be updated either the base of the availability of named-data items on peers or on the presence of peers, and these events in a P2P session is rapidly changing so possibly hampering the convergence of the routing plane. Finally, since peers have an impact on the ICN forwarding table of public nodes, this may open obvious security issues.

6.3. Impact of MPEG DASH coding schemes

The introduction of video rate adaptation may valuably decrease the effectiveness of P2P cooperation and of in-network caching, depending of the kind of the video coding used by the MPEG DASH stream.

In case of a MPEG DASH streaming with MPEG AVC encoding, a same video chunk is independently encoded at different rates and the encoding output is a different file for each rate. For instance, in case of a video encoded at three different rates R1,R2,R3, for each

segment S we have three distinct files: S.R1, S.R2, S.R3. These files are independent of each other. To fetch a segment coded at R2 kbps, a peer shall request the specific file S.R2. The estimation of the best coding rate is usually handled by receiver-driven algorithms, implemented by the video client.

The independence among files associated to different encoding rates and the heterogeneity of peer bandwidths, may dramatically reduce the interaction among peers, the effectiveness of in-network caching (in case of integrated deployment), and consequently the ability of PPSP to offload the video server (i.e. a seeder peer). Indeed, a peer A may select a coding rate (e.g. R1) different from the one selected by a peer B (e.g. R2) and this prevents the former to fetch video chunks from the later, since peer B avails of chunks coded at a rate different from the ones needed by A. To overcome this issue, a common distributed rate selection algorithm could force peers to select the same coding rate [17]; nevertheless this approach may be not feasible in the in case of many peers.

The use of SVC encoding (Annex G extension of the H.264/MPEG-4 AVC video compression standard) should make rate adaptation possible, meanwhile neither reducing peer collaborations nor the in-network caching effectiveness. For a single video chunk, a SVC encoder produces different files for the different rates (roughly "layers"), and these files are progressively related each other. Starting from a base-layer which provides the minimum rate encoding, the next rates are encoded as an "enhancement layer" of the previous one. For instance, in case the video is coded with three rates R1 (base-layer), R2 (enhancement-layer n.1), R3 (enhancement-layer n.2), then for each DASH segment we have three files S.R1, S.R2 and S.R3. The file S.R1 is the segment coded at the minimum rate (base-layer). The file S.R2 enhances S.R1, so as S.R1 and S.R2 can be combined to obtain a segment coded at rate R2. To get a segment coded at rate R2, a peer shall fetch both S.R1 and S.R2. This progressive dependence among files that encode a same segment at different rates makes peer cooperation possible, also in case peers player have autonomously selected different coding rates. For instance, if peer A has selected the rate R1, the downloaded files S.R1 are useful also for a peer B that has selected the rate R2, and vice versa.

7. IPTV and ICN

7.1. IPTV challenges

IPTV refers to the delivery of quality content broadcast over the Internet, and is typically associated with strict quality

requirements, i.e., with a perceived latency of less than 500 ms and a packet loss rate that is multiple orders lower than the current loss rates experienced in the most commonly used access networks. We can summarize the major challenges for the delivery of IPTV service as follows.

Channel change latency represents a major concern for the IPTV service. Perceived latency during channel change should be less than 500ms. To achieve this objective over the IP infrastructure, we have multiple choices:

(i) receiving fast unicast streams from a dedicated server (most effective but not resource efficient);
(ii) connecting to other peers in the network (efficiency depends on peer support, effective and resource efficient, if also supported with a dedicated server);
(iii) connecting to multiple multicast sessions at once (effective but not resource efficient, and depends on the accuracy of the prediction model used to track user activity).

The second major challenge is the error recovery. Typical IPTV service requirements dictate the mean time between artifacts to be approximately 2 hours. This suggests the perceived loss rate to be around or less than 10^-7. Current IP-based solutions rely on the following proactive and reactive recovery techniques: (i) joining the FEC multicast stream corresponding to the perceived packet loss rate (not efficient as the recovery strength is chosen based on worst-case loss scenarios), (ii) making unicast recovery requests to dedicated servers (requires active support from the service provider), (iii) probing peers to acquire repair packets (finding matching peers and enabling their cooperation is another challenge).

7.2. ICN benefits for IPTV delivery

ICN presents significant advantages for the delivery of IPTV traffic. For instance, ICN inherently supports multicast and allows for quick recovery from packet losses (with the help of in-network caching). Similarly, peer support is also provided in the shape of in-network caches that typically act as the middleman between two peers, enabling therefore earlier access to IPTV content.

However, despite these advantages, delivery of IPTV service over Information Centric Networks brings forth new challenges. We can list some of these challenges as follows:

. Messaging overhead: ICN is a pull-based architecture and relies on a unique balance between requests and responses. A user needs to make a request for each data packet. In the case of IPTV, with rates up to, and likely to be, above 15Mbps, we observe significant traffic upstream to bring those streams. As the number of streams increase (including the same session at different quality levels), so as the burden on the routers. Even if the majority of requests are aggregated at the core, routers close to the edge (where we observe the biggest divergence in user requests) will experience a significant increase in overhead to process these requests. The same is true at the user side, as the uplink usage multiplies in the number of sessions a user requests (for instance, to minimize the impact of bandwidth fluctuations).
. Cache control: As the IPTV content expires at a rapid rate (with a likely expiry threshold of 1s), we need solutions to effectively flush out such content to also prevent degradatory impact on other cached content, with the help of intelligently chosen naming conventions. However, to allow for fast recovery and optimize access time to sessions (from current or new users), the timing of such expirations needs to be adaptive to network load and user demand. However, we also need to support quick access to earlier content, whenever needed, for instance, when the user accesses the rewind feature (note that in-network caches will not be of significant help in such scenarios due to overhead required to maintain such content).
. Access accuracy: To receive the up-to-date session data, users need to be aware of such information at the time of their request. Unlike IP multicast, since the users join a session indirectly, session information is critical to minimize buffering delays and reduce the startup latency. Without such information, and without any active cooperation from the intermediate routers, stale data can seriously undermine the efficiency of content delivery. Furthermore, finding a cache does not necessarily equate to joining a session, as the look-ahead latency for the initial content access point may have a shorter lifetime than originally intended. For instance, if the

user that has initiated the indirect multicast leaves the session early, the requests from the remaining users need to experience an additional latency of one RTT as they travel towards the content source. If the startup latency is chosen depending on the closeness to the intermediate router, going to the content source in-session can lead to undesired pauses.

8. Digital Rights Managements in ICN

This section discusses the need for Digital Rights Management (DRM) functionalities for multimedia streaming over ICN. It focuses on two possible approaches: modifying AAA to support DRM in ICN, and using Broadcast Encryption.

It is assumed that ICN will be used heavily for digital content dissemination. It is vital to consider DRM for digital content distribution. In today’s Internet there are two predominant classes of business models for on-demand video streaming. The first model is based on advertising revenues. Non-copyright protected (usually user-generated content, UGC) is offered by large infrastructure providers like Google (YouTube) at no charge. The infrastructure is financed by spliced advertisements into the content. In this context DRM considerations may not be required, since producers of UGC may only strive for the maximum possible dissemination. Some producers of UGC are mainly interested to share content with their families, friends, colleges or others and have no intention to make profit. However, the second class of business models requires DRM, because they are primarily profit oriented. For example, large on-demand streaming platforms like Netflix establish business models based on subscriptions. Consumers may have to pay a monthly fee in order to get access to copyright protected content like TV series, movies or music. This model may be ad-supported and free to the content consumer, like YouTube Channels or Spotify. But the creater of the content expects some remuneration for his work. From the perspective of the service providers and the copyright owners, only clients that pay the fee (explicitly or implicitly through ad placement) should be able to access and consume the content. Anyway, the challenge is to find an efficient and scalable way of access control to digital content, which is distributed in information-centric networks.

8.1. Broadcast Encryption for DRM in ICN

The section discusses Broadcast Encryption (BE) as a suitable basis for DRM functionalities in conformance to the ICN communication paradigm. Especially when network inherent caching is considered the advantage of BE will be highlighted.

In ICN, data packets can be cached inherently in the network and any network participant can request a copy of these packets. This makes it very difficult to implement an access control for content that is distributed via ICN. A naive approach is to encrypt the transmitted data for each consumer with a distinct key. This prohibits everyone other than the intended consumers to decrypt and consume the data. However, this approach is not suitable for ICN's communication paradigm since it would reduce the benefits gained from the inherent network caching. Even if multiple consumers request the same content the requested data for each consumer would differ using this approach. A better but still insufficient idea is to use a single key for all consumers. This does not destruct the benefits of ICN's caching ability. The drawback is that if one of the consumers illegally distributes the key, the system is broken and any entity in the network can access the data. Changing the key after such an event is useless since the provider has no possibility to identify the illegal distributer. Therefore this person cannot be stopped from distributing the new key again. In addition to this issue other challenges have to be considered. Subscriptions expire after a certain time and then it has to be ensured that these consumers cannot access the content anymore. For a provider that serves millions of daily consumers (e.g. Netflix) there could be a significant number of expiring subscriptions per day. Publishing a new key every time a subscription expires would require an unsuitable amount of computational power just to re-encrypt the collection of audio-visual content.

A possible approach to solve these challenges is Broadcast Encryption (BE) [22] as proposed in [23]. From this point on, this section will focus only on BE as an enabler for DRM functionality in the use case of ICN video streaming. This subsection continues with the explanation of how BE works and shows how BE can be used to implement an access control scheme in the context of content distribution in ICN.

BE actually carries a misleading name. One might expect a concrete encryption scheme. However, it belongs to the family of key-management schemes (KMS). KMS are responsible for the generation, exchange, storage and replacement of cryptographic keys. The most

interesting characteristics of Broadcast Encryption Schemes (BES) are:

- A BES typically uses a global trusted entity called the licensing agent (LA), which is responsible for spreading a set of pre-generated secrets among all participants. Each participant gets a distinct subset of secrets assigned from the LA.
- The participants can agree on a common session key, which is chosen by the LA. The LA broadcasts an encrypted message that includes the key. Participants with a valid set of secrets can derive the session-key from this message.
- The number of participants in the system can change dynamically. Entities may join or leave the communication group at any time. If a new entity joins the LA passes on a valid set of secrets to that entity. If an entity leaves (or is forced to leave) the LA revokes the entity's subset of keys, which means that it cannot derive the correct session key anymore when the LA distributes a new key.
- -Traitors (entities that reveal their secrets) can be traced and excluded from ongoing communication. The algorithms and preconditions to identify a traitor vary between concrete BES.

This listing already illustrates why BE is suitable to control the access to data that is distributed via an information-centric network. BE enables the usage of a single session key for confidential data transmission between a dynamically changing subset or network participants. ICN caches can be utilized since the data is encrypted only with a single key known by all legitimate clients. Furthermore, traitors can be identified and removed from the system. The issue of re-encryption still exists, because the LA will eventually update the session key when a participant should be excluded. However, this disadvantage can be relaxed in some way if the following points are considered:

- The updates of the session key can be delayed until a set of compromised secretes has been gathered. Note that secrets may become compromised because of two reasons. First, if the secret has been illegally revealed by a traitor. Second, if the subscription of an entity expires. Delayed revocation temporarily enables some non-legitimate entities to consume content. However, this should not be a severe problem in home entertainment scenarios. Updating the session key in regular (not too short) intervals is a good tradeoff. The longer the interval last the less computational resources are required for content re-encryption and the better the cache utilization in the ICN will be. To evict old data from ICN caches that has

been encrypted with the prior session key the publisher could indicate a lifetime for transmitted packets.

. Content should be re-encrypted dynamically at request time. This has the benefit that untapped content is not re-encrypted if the content is not requested during two session key updates and therefore no resources are wasted. Furthermore, if the updates are triggered in non-peak times the maximum amount of resource needed at one point in time can be lowered effectively, since in peak times generally more diverse content is requested.

. Since the amount of required computational resources may vary strongly from time to time it would be beneficial for any streaming provider to use cloud-based services to be able to dynamically adapt the required resources to the current needs. Regarding to a lack of computation time or bandwidth the cloud service could be used to scale up to overcome shortages.

Figure 4 show the potential usage of BE in a multimedia delivery frameworks that builds upon ICN infrastructure and uses the concept of dynamic adaptive streaming, e.g., DASH. BE would be implemented on the top to have an efficient and scalable way of access control to the multimedia content.

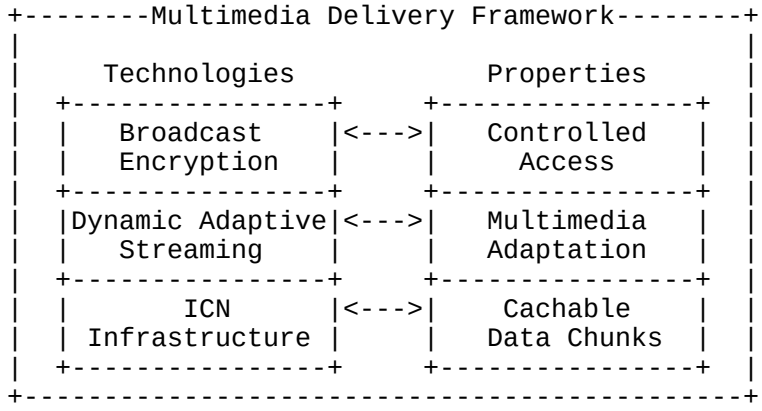

Figure 4: A potential multimedia framework using BE.

8.2 . AAA Based DRM for ICN Networks

8.2.1. Overview

Recently, a novel approach to Digital Rights Management (DRM) has emerged to link DRM to usual network management operations, hence linking DRM to authentication, authorization, and accounting (AAA) services. ICN provides the abstraction of an architecture where content is requested by name and could be served from anywhere. In DRM, the content provider (the origin of the content) allows the destination (the end user account) to use the content. The content provider and content storage/cache are at two different entities in ICC and for traditional DRM only source and destination count and not the intermediate storage. The proposed solution allows the provider of the caching to be involved in the DRM policies using well known AAA mechanisms. It is important to note that this solution is compatible with the proposes the Broadcast Encryption (BE) proposed earlier in this draft. The BE proposes a technology as this solution is more operational.

8.2.2. Implementation

With the proposed AAA-based DRM, when a content is requested by name from a specific destination, the request could link back to both the content provider and the cahing provider via traditional AAA mechanisms, and trigger the appropriate DRM policy independently from where the content is stored. In this approach the caching, DRM and AAA remain independent entities but can work together through ICN mechanisms. The proposed solution enables extending the traditional DRM done by the content provider to jointly being done by content provider and network/caching provider.

The solution is based on the concept of a "token". The content provider authenticates the end user and issues an encrypted token to authenticate the a named content ID or IDs that the user can access. The token will be shared with the network provider and used as the interface to the AAA protocols. At this point all content access is under the control of the network provider and the ICN. The controllers and switches can manage the content requests and handle mobility. The content can be accessed from anywhere as long as the token remains valid or the content is available in the network. In such a scheme the content provider does not need to be contacted every time a named content is requested. This reduces the load of the content provider network and creates a DRM mechanism that is much more appropriate for the distributed caching and peer-to-peer

storage characteristic of ICN networks. In particular, the content requested by name can be served from anywhere under the only condition that the storage/cache can verify that the token is valid for content access.

The solution is also fully customizable to both content and network provider's needs as the tokens can be issued based on user accounts, location and hardware (MAC address for example) linking it naturally to legacy authentication mechanisms. In addition, since both content and network providers are involved in DRM policies pollution attacks and other illegal requests for the content can be more easily detected. The proposed AAA-based DRM is currently under full development.

9. Future Steps for Video in ICN

The explosion of online video services, along with their increased consumption by mobile wireless terminals, further exacerbates the challenges of Video Adaptation leveraging ICN mechanisms. The following sections present a series of research items derived from these challenges, further introducing next steps for the subject.

9.1. Large Scale Live Events

An active area of investigation and a potential use case where ICN would provide significant benefits, is that of distributing content, and video in particular, using local communications in large scale events such as sports event in a stadium, a concert or a large demonstration.

Such use-case involves locating content that is generated on the fly and requires discovery mechanisms in addition to sharing mechanisms. The scalability of the distribution becomes important as well.

9.2. Video Conferencing and Real-Time Communications

Current protocols for video-conferencing have been designed, and this document needs to take input from them to identify the key research issues. Real-time communication add timing constraints (both in terms of delay and in terms of synchronization) to the scenario discussed above.

9.3. Store-and-Forward Optimized Rate Adaptation

One of the benefits of ICN is to allow the network to insert caching in the middle of the data transfer. This can be used to reduce the overall bandwidth demands over the network by caching content for

future re-use. But it provides more opportunities for optimizing video streams.

Consider for instance the following scenario: a client is connected via an ICN network to a server. Let’s say the client is connected wirelessly to a node that has a caching capability, which is connected through a WAN to the server. Assume further that the capacity of each of the links (both the wireless and the WAN logical links) vary with time.

If the rate adaptation is provided in an end-to-end manner, as in current mechanisms like DASH, then the maximal rate that can be supported at the client is that of the minimal bandwidth on each link.

For instance, if during time period 1, the wireless capacity is 1 and the wired capacity is 2, and during time period 2, the wireless is 2 due to some hotspot, and the wired is 1 due to some congestion in the network, then the best end-to-end rate that can be achieved is 1 during each period.

However, if the cache is used during time period 1 to pre-fetch 2 units of data, then during period 2, there is 1 unit of data at the cache, and another unit of data, which can be streamed from the server, and the rate that can be achieved is therefore 2 units of data. In this case, the average bandwidth rises from 1 to 1.5 over the 2 periods.

This straw man example illustrate a) the benefit of ICN for increasing the throughput of the network, and b) the need for the special rate adaptation mechanisms to be designed so as to take advantage of this gain. End-to-end rate adaptation can not take advantage of the cache availability.

9.4. Heterogeneous Wireless Environment Dynamics

With the ever-growing increase in online services being accessed by mobile devices, operators have been deploying different overlapping wireless access networking technologies. In this way, in the same area, user terminals are within range of different cellular, Wi-Fi or even WiMAX networks. Moreover, with the advent of the Internet of Things (e.g., surveillance cameras feeding video footage), this list can be further complemented with more specific short-range technologies, such as Bluetooth or ZigBee.

In order to leverage from this plethora of connectivity opportunities, user terminals are coming equipped with different

wireless access interfaces, providing them with extended connectivity opportunities. In this way, such devices become able to select the type of access which best suits them according to different criteria, such as available bandwidth, battery consumption, access do different link conditions according to the user profile or even access to different content. Ultimately, these aspects contribute to the Quality of Experience perceived by the end-user, which is of utmost importance when it comes to video content.

However, the fact that these users are mobile and using wireless technologies, also provides a very dynamic setting, where the current optimal link conditions at a specific moment might not last or be maintained while the user moves. These aspects have been amply analyzed in recently finished projects such as FP7 MEDIEVAL [18], where link events reporting on wireless conditions and available alternative connection points were combined with vide requirements and traffic optimization mechanisms, towards the production of a joint network and mobile terminal mobility management decision. Concretely, in [19] link information about the deterioration of the wireless signal was sent towards a mobility management controller in the network. This input was combined with information about the user profile, as well as of the current video service requirements, and used to trigger the decrease or increase of scalable video layers, adjusting the video to the ongoing link conditions. Incrementally, the video could also be adjusted when a new better connectivity opportunity presents itself.

In this way, regarding Video Adaptation, ICN mechanisms can leverage from their intrinsic multiple source support capability and go beyond the monitoring of the status of the current link, thus exploiting the availability of different connectivity possibilities (e.g., different "interfaces"). Moreover, information obtained from the mobile terminal’s point of view of its network link, as well as information from the network itself (i.e., load, policies, and others), can generate scenarios where such information is combined in a joint optimization procedure allowing the content to be forward to users using the best available connectivity option (e.g., exploiting management capabilities supported by ICN intrinsic mechanisms as in [20]).

In fact, ICN base mechanisms can further be exploited in enabling new deployment scenarios such as preparing the network for mass requests from users attending a large multimedia event (i.e., concert, sports), allowing video to be adapted according to content, user and network requirements and operation capabilities in a dynamic way.

The enablement of such scenarios require further research, with the main points highlighted as follows:

. Development of a generic video services (and obviously content) interface allowing the definition and mapping of their requirements (and characteristics) into the current capabilities of the network;

. How to define a scalable mechanism allowing either the video application at the terminal, or some kind of network management entity, to adapt the video content in a dynamic way;

. How to develop the previous research items using intrinsic ICN mechanisms (i.e., naming and strategy layers);

. Leverage intelligent pre-caching of content to prevent stalls and poor quality phases, which lead to bad Quality of Experience of the user. This includes in particular the usage in mobile environments, which are characterized by severe bandwidth changes as well as connection outages, as shown in [21].

9.5. Network Coding for Video Distribution in ICN

An interesting research area for combining heterogeneous sources is to use network coding [24]. Network coding allows to asynchronously combine multiple sources by having each of them send information that is not duplicated by the other but can be combined to retrieve the video stream.

However, this creates issues in ICN in terms of defining the proper rate adaptation for the video stream; securing the encoded data; caching the encoded data; timeliness of the encoded data; overhead of the network coding operations both in network resources and in added buffering delay, etc.

10. Security Considerations

This is informational. Security considerations are TBD.

11. IANA Considerations

This is informational. IANA considerations are TBD.

12. Conclusions

This draft proposed adaptive video streaming for ICN, identified potential problems and presented the combination of CCN with DASH as

a solution. As both concepts, DASH and CCN, maintain several elements in common, like, e.g., the content in different versions being dealt with in segments, combination of both technologies seems useful. Thus, adaptive streaming over CCN can leverage advantages such as, e.g., efficient caching and intrinsic multicast support of CCN, routing based on named data URIs, intrinsic multi-link and multi-source support, etc.

In this context, the usage of CCN with DASH in mobile environments comes together with advantages compared to today's solutions, especially for devices equipped with multiple network interfaces. The retrieval of data over multiple links in parallel is a useful feature, specifically for adaptive multimedia streaming, since it offers the possibility to dynamically switch between the available links depending on their bandwidth capabilities, transparent to the actual DASH client.

13. References

13.1. Normative References

[RFC6972] Y. Zhang, N. Zong, "Problem Statement and Requirements of the Peer-to-Peer Streaming Protocol (PPSP)", RFC6972, July 2013

13.2. Informative References

[1] ISO/IEC DIS 23009-1.2, Information technology - Dynamic adaptive streaming over HTTP (DASH) - Part 1: Media presentation description and segment formats

[2] Lederer, S., Mueller, C., Rainer, B., Timmerer, C., Hellwagner, H., "An Experimental Analysis of Dynamic Adaptive Streaming over HTTP in Content Centric Networks", in Proceedings of the IEEE International Conference on Multimedia and Expo 2013, San Jose, USA, July, 2013

[3] Liu, Y., Geurts, J., Point, J., Lederer, S., Rainer, B., Mueller, C., Timmerer, C., Hellwagner, H., "Dynamic Adaptive Streaming over CCN: A Caching and Overhead Analysis", in Proceedings of the IEEE international Conference on Communication (ICC) 2013 - Next-Generation Networking Symposium, Budapest, Hungary, June, 2013

[4] Grandl, R., Su, K., Westphal, C., "On the Interaction of Adaptive Video Streaming with Content-Centric Networks", eprint arXiv:1307.0794, July 2013.

[5] V. Jacobson, D. Smetters, J. Thornton, M. Plass, N. Briggs and R. Braynard, "Networking named content", in Proc. of the 5th int. Conf. on Emerging Networking Experiments and Technologies (CoNEXT '09). ACM, New York, NY, USA, 2009, pp. 1-12.

[6] A. Detti, M. Pomposini, N. Blefari-Melazzi, S. Salsano and A. Bragagnini, "Offloading cellular networks with Information-Centric Networking: The case of video streaming", In Proc. of the Int. Symp. on a World of Wireless, Mobile and Multimedia Networks (WoWMoM '12), IEEE, San Francisco, CA, USA, 1-3, 2012.

[7] V. Jacobson, D. K. Smetters, N. H. Briggs, M. F. Plass, P. Stewart, J. D. Thornton, and R. L. Braynard, "VoCCN: Voice over content-centric networks," in ACM ReArch Workshop, 2009

[8] Christopher Mueller, Stefan Lederer and Christian Timmerer, A proxy effect analysis and fair adaptation algorithm for multiple competing dynamic adaptive streaming over HTTP clients, In Proceedings of the Conference on Visual Communications and Image Processing (VCIP) 2012, San Diego, USA, November 27-30, 2012.

[9] DASH Research at the Institute of Information Technology, Multimedia Communication Group, Alpen-Adria Universitaet Klagenfurt, URL: http://dash.itec.aau.at

[10] A. Detti, N. Blefari-Melazzi, S. Salsano, and M. Pomposini, "CONET: A content centric inter-networking architecture," in ACM Workshop on Information-Centric Networking (ICN), 2011.

[11] W. K. Chai, N. Wang, I. Psaras, G. Pavlou, C. Wang, G. C. de Blas, F. Ramon-Salguero, L. Liang, S. Spirou, A. Beben, and E. Hadjioannou, "CURLING: Content-ubiquitous resolution and delivery infrastructure for next-generation services," IEEE Communications Magazine, vol. 49, no. 3, pp. 112-120, March 2011

[12] NetInf project Website http://www.netinf.org

[13] N. Magharei, R. Rejaie, Yang Guo, "Mesh or Multiple-Tree: A Comparative Study of Live P2P Streaming Approaches," INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE , vol., no., pp.1424,1432, 6-12 May 2007

[14] PPSP WG Website https://datatracker.ietf.org/wg/ppsp/

[15] A. Bakker, R. Petrocco, V. Grishchenko, "Peer-to-Peer Streaming Peer Protocol (PPSPP)", draft-ietf-ppsp-peer-protocol-08

[16] Rui S. Cruz, Mario S. Nunes, Yingjie Gu, Jinwei Xia, Joao P. Taveira, Deng Lingli, "PPSP Tracker Protocol-Base Protocol (PPSP-TP/1.0)", draft-ietf-ppsp-base-tracker-protocol-02

[17] A.Detti, B. Ricci, N. Blefari-Melazzi,"Peer-To-Peer Live Adaptive Video Streaming for Information Centric Cellular Networks", IEEE PIMRC 2013,London, UK, 8-11 September 2013

[18] http://www.ict-medieval.eu

[19] B. Fu, G. Kunzmann, M. Wetterwald, D. Corujo, R. Costa, "QoE-aware Traffic Management for Mobile Video Delivery", Proc. 2013 IEEE ICC, Workshop on Immersive & Interactive Multimedia Communications over the Future Internet (IIMC), Budapest, Hungary, Jun 2013.

[20] Corujo D., Vidal I., Garcia-Reinoso J., Aguiar R., "A Named Data Networking Flexible Framework for Management Communications", IEEE Communications Magazine, Vol. 50, no. 12, pp. 36-43, Dec 2012

[21] Crabtree B., Stevens T., Allan B., Lederer S., Posch D., Mueller C., Timmerer C., Video Adaptation in Limited or Zero Network Coverage, CCNxConn 2013,PARC, Palo Alto, pp. 1-2, 2013

[22] Fiat A., Naor M., "Broadcast Encryption", in Advances in Cryptology (Crypto'93), volume 773 of Lecture Notes in Computer Science, pages 480-491. Springer Berlin / Heidelberg, 1994.

[23] Posch D., Hellwagner H., Schartner P., "On-Demand Video Streaming based on Dynamic Adaptive Encrypted Content Chunks", in Proceedings of the 8th International Workshop on Secure Network Protocols (NPSec' 13), Los Alamitos, IEEE Computer Society Press, October, 2013.

[24] Montpetit M.J., Westphal C., Trossen D., "Network Coding Meets Information Centric Networks," in Proceedings of the workshop on Name-Oriented Mobility (NOM), jointly with ACM MobiHoc 2013, Hilton Head, SC, June 2013.

14. Authors' Addresses

Stefan Lederer, Christian Timmerer, Daniel Posch
Alpen-Adria University Klagenfurt
Universitaetsstrasse 65-67, 9020 Klagenfurt, Austria

Email: {firstname.lastname}@itec.aau.at

Cedric Westphal, Aytac Azgin, Sucheng (Will) Liu
Huawei
2330 Central Expressway, Santa Clara, CA95050, USA

Email: {cedric.westphal, aytac.azgin, liushucheng}@huawei.com

Christopher Mueller
bitmovin GmbH
Lakeside B01, 9020 Klagenfurt, Austria

Email: christopher.mueller@bitmovin.net

Andrea Detti
Electronic Engineering Dept.
University of Rome Tor Vergata
Via del Politecnico 1, Rome, Italy

Email: andrea.detti@uniroma2.it

Daniel Corujo,
Advanced Telecommunications and Networks Group
Instituto de Telecomunicacoes
Campus Universitario de Santiago
P-3810-193 Aveiro, Portugal

Email: dcorujo@av.it.pt

15. Acknowledgements

This work was supported in part by the EC in the context of the SocialSensor (FP7-ICT-287975) project and partly performed in the Lakeside Labs research cluster at AAU. SocialSensor receives research funding from the European Community's Seventh Framework Programme. The work for this document was also partially performed in the context of the FP7/NICT EU-JAPAN GreenICN project, http://www.greenicn.org. Apart from this, the European Commission has no responsibility for the content of this draft. The information in this document is provided as is and no guarantee or warranty is given that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability. The authors would like to Dr. Jianping Wang (City University Hong Kong) and Marie-Jose Montpetit of MIT for their help in writing the AAA for DRM section.