

Internet Research Task Force (IRTF)
Request for Comments: 7933
Category: Informational
ISSN: 2070-1721

C. Westphal, Ed.
Huawei
S. Lederer
D. Posch
C. Timmerer
Alpen-Adria University Klagenfurt
A. Azgin
W. Liu
Huawei
C. Mueller
BitMovin
A. Detti
University of Rome Tor Vergata
D. Corujo
Instituto de Telecomunicacoes Aveiro
J. Wang
City University of Hong Kong
M. Montpetit
MIT
N. Murray
Athlone Institute of Technology
August 2016

Adaptive Video Streaming over Information-Centric Networking (ICN)

Abstract

This document considers the consequences of moving the underlying network architecture from the current Internet to an Information-Centric Networking (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. The wide range of scenarios covered includes the following: evolving Dynamic Adaptive Streaming over HTTP (DASH) to work over ICN and leverage the recent ISO/IEC Moving Picture Experts Group (MPEG) standard, layering encoding over ICN, introducing distinct requirements for video using Peer-to-Peer (P2P) mechanisms, adapting the Peer-to-Peer Streaming Protocol (PPSP) for ICN, creating more stringent requirements over ICN because of delay constraints added by Internet Protocol Television (IPTV), and managing digital rights in ICN. Finally, in addition to considering how existing mechanisms would be impacted by ICN, this document lists some research issues to design ICN-specific video streaming mechanisms.

Status of This Memo

This document is not an Internet Standards Track specification; it is published for informational purposes.

This document is a product of the Internet Research Task Force (IRTF). The IRTF publishes the results of Internet-related research and development activities. These results might not be suitable for deployment. This RFC represents the consensus of the Information-Centric Networking Research Group of the Internet Research Task Force (IRTF). Documents approved for publication by the IRSG are not a candidate for any level of Internet Standard; see [Section 2 of RFC 7841](#).

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at <http://www.rfc-editor.org/info/rfc7933>.

Copyright Notice

Copyright (c) 2016 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document.

Table of Contents

1. Introduction	4
2. Conventions Used in This Document	4
3. Use Case Scenarios for ICN and Video Streaming	5
4. Video Download	6
5. Video Streaming and ICN	7
5.1. Introduction to Client-Driven Streaming and DASH	7
5.2. Layered Encoding	8
5.3. Interactions of Video Streaming with ICN	8
5.3.1. Interactions of DASH with ICN	8
5.3.2. Interaction of ICN with Layered Encoding	10
5.4. Possible Integration of Video Streaming and ICN Architecture	11
5.4.1. DASH over CCN	11
5.4.2. Testbed, Open-Source Tools, and Dataset	13

6.	P2P Video Distribution and ICN	14
6.1.	Introduction to PPSP	14
6.2.	PPSP over ICN: Deployment Concepts	16
6.2.1.	PPSP Short Background	16
6.2.2.	From PPSP Messages to ICN Named-Data	16
6.2.3.	Support of PPSP Interaction through a Pull-Based ICN API	17
6.2.4.	Abstract Layering for PPSP over ICN	18
6.2.5.	PPSP Interaction with the ICN Routing Plane	19
6.2.6.	ICN Deployment for PPSP	19
6.3.	Impact of MPEG-DASH Coding Schemes	20
7.	IPTV and ICN	21
7.1.	IPTV Challenges	21
7.2.	ICN Benefits for IPTV Delivery	22
8.	Digital Rights Management in ICN	24
8.1.	Broadcast Encryption for DRM in ICN	24
8.2.	AAA-Based DRM for ICN Networks	27
8.2.1.	Overview	27
8.2.2.	Implementation	28
9.	Future Steps for Video in ICN	28
9.1.	Large-Scale Live Events	29
9.2.	Video Conferencing and Real-Time Communications	29
9.3.	Store-and-Forward Optimized Rate Adaptation	29
9.4.	Heterogeneous Wireless Environment Dynamics	30
9.5.	Network Coding for Video Distribution in ICN	32
9.6.	Synchronization Issues for Video Distribution in ICN	32
10.	Security Considerations	33
11.	Conclusions	33
12.	References	34
12.1.	Normative References	34
12.2.	Informative References	34
	Acknowledgments	38
	Authors' Addresses	39

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 ICNRG research 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 near future. If ICN is expected to become prominent, it will have to support video streaming efficiently.

As such, it is necessary to discuss going in two separate 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 focuses on the first question in an attempt to define the use cases for video streaming and some requirements. It also focuses on a few scenarios (namely, Netflix-like video streaming, P2P 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.

In addition to this document, other works have considered specific aspects of dynamic adaptive streaming in ICN [[Lederer13b](#)] [[Lederer13a](#)] [[Grandl13](#)] [[Detti12](#)]. This document is informed by these works, as well as others.

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

3. Use Case Scenarios for ICN and Video Streaming

For ICN-specific descriptions, we refer to the other research group documents [[RFC7476](#)]. For our purpose, we assume here that "ICN" refers to an architecture where content is retrieved by name and with no binding of content to a specific network location.

Both live and on-demand consumption of multimedia content come with timing requirements for the delivery of the content. Additionally, real-time use cases (such as audio-video conferencing [[Jacobson09a](#)], game streaming, etc.) come with stricter timing requirements. Long startup delays, buffering periods, poor video quality, etc., should be avoided to achieve a better Quality of Experience (QoE) for the consumer of the content. For a definition of QoE in the context of video distribution, please refer to [[LeCallet13](#)]. The working definition is as follows:

Quality of Experience (QoE) is the degree of delight or annoyance of the user of an application or service. It results from the fulfillment of his or her expectations with respect to the utility and/or enjoyment of the application or service in the light of the user's personality and current state.

Of course, these requirements are heavily influenced by routing decisions and caching, which are central parts of ICN and that have to be considered when streaming video in such infrastructures.

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

- o a video download architecture similar to that of Apple iTunes, where the whole file is being downloaded to the client and can be replayed there multiple times;
- o a video streaming architecture for playing back movies, which 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;
- o a P2P architecture for sharing videos, which 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 (for P2P definitions and taxonomy, please refer to [RFC 5694](#); and
- o IPTV, which 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 even though they are in scope. Also, events of mass-media distribution, such as a large crowd at a live event, add new requirements to be included in later versions.

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: [Section 4](#) discusses video download; [Section 5](#) briefly describes DASH [ISO-DASH] and Layered Encoding (Modification Detection Code (MDC), Scalable Video Coding (SVC)); [Section 6](#) focuses on P2P and PPSP; [Section 7](#) highlights the requirements of IPTV; [Section 8](#) describes the issues of DRM; and [Section 9](#) lists some research issues to be solved for ICN-specific video delivery mechanisms.

Video-conferencing and real-time-video communications will be described in further detail in future works. Mass distribution of content at live, large-scale events (stadiums, concert halls, etc.) for which there is no clearly adopted existing protocol is another topic for further research.

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, Secure Copy (SCP), FTP, or over HTTP), which would need to be replaced by new ICN-specific protocols to retrieve content in ICNs.

However, current mechanisms are built atop existing transport protocols. Some ICN proposals (Context-Centric Network (CCN) or Named Data Networking (NDN), for instance) attempt to leverage the work done upon these transport protocols. One proposal is to use 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 HTTP and, in a further consequence, streaming over the TCP, has become omnipresent in today's Internet. Content providers such as Netflix, Hulu, and Vudu do not deploy their own streaming equipment: they use the existing Internet infrastructure as it is and 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 use of 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 [[ISO-DASH](#)] is to use segments of media content, which can be encoded at different resolutions, bit rates, 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 for adaptation of 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 bit rate, resolution, and

timeline, the Media Presentation Description (MPD) is used, which is an 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 consists in using layered encoding. Namely, scalable video coding formats the video stream into different layers: a base layer that can be decoded to provide the lowest bit rate for the specific stream and enhancement layers that 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 for adaptation to the network conditions. This is used in an 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. Interactions of DASH with ICN

Video streaming (DASH in particular) has been designed with a goal that is aligned with that of most ICN proposals: it is a client-based mechanism that requests items (in this case, chunks of a video stream) by name.

ICN and MPEG-DASH [[ISO-DASH](#)] have several elements in common:

- o the client-initiated pull approach;
- o the content being dealt with in pieces (or chunks);
- o the support of efficient replication and distribution of content pieces within the network;
- o 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; and
- o the support for potentially multiple source locations.

For the last point, DASH may list multiple source URLs in a manifest, and ICN is agnostic to the location of a copy it is receiving. We do not imply that current video streaming mechanisms attempt to draw the

content from multiple sources concurrently. This is a potential benefit of ICN but is not considered in the current approaches mentioned in this document.

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 (note that some of the issues are related to caching and are not necessarily unique to ICN):

- 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.
- o 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 a stream using a single encoding, the cache size must be scaled up to include all the possible representations.
- o Caching introduces oscillatory dynamics as it may modify the estimation of the available bandwidth between the end user and the repository from which it is getting the chunks. For instance, if an edge cache holds a low resolution representation near the user,

the user getting these 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.

- o 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.
- o The usage of multiple network interfaces is possible in ICN, enabling a seamless handover between them. For the combination with DASH, an intelligent strategy that 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.
- o 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 (including 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 [[Jacobson09b](#)].

5.3.2. Interaction of ICN with Layered Encoding

Issues of interest to an ICN architecture in the context of layered video streaming include:

- o Caching of the multiple layers. The caching priority should go to the base layer and to defining caching policy in order to decide when to cache enhancement layers;
- o 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). Video and audio-video streams must be synchronized, and this includes both intra-layer synchronization (for the layers of the same video or

audio stream) and inter-stream synchronization (see [Section 9](#) for other synchronization aspects to be included in the "Future Steps for Video in ICN"); and

- o 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 today's 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 with respect to 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 Wi-Fi. 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 [\[Det112\]](#), 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. Since such a proxy is already described in [\[Det112\]](#), this document

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 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; thus, it 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 compete in a bottleneck and when caching influences 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 that may impact the QoE at the client, as shown in [Mueller12]. 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 adaptation logic use bandwidth measurements and related heuristics, the adaptation decisions are no longer valid when changing origin servers (or links), and these decisions potentially cause playback interruptions and, consequently, stalling. Additionally, ICN supports the usage of multiple interfaces. A seamless handover between these interfaces (and different sources for the content) comes together with changes in performance, e.g., due to switching between fixed and wireless, 3G/4G and Wi-Fi networks, different types of servers (say with/without Shared Secret Data (SSD) or hardware acceleration), 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 conditions. 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 [ITEC-DASH]. 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 are located 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 front end, 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 the form of a CCNx repository. It includes 15 different quality representation of the well-known Big Buck Bunny Movie, ranging from 100 kbps to 4500 kbps. The content is split into segments of two seconds and is described by an associated MPD using the presented naming scheme in [Section 5.1](#). This repository can be downloaded from [\[ITEC-DASH\]](#) and is also provided by a publicly 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

Peer-to-Peer distribution is another form of distributing content -- and video in particular -- that ICNs need to support. 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 (P2PVS) is a popular approach to redistribute live media over the Internet. The proposed P2PVS solutions can be roughly classified in two classes:

- o Push/Tree-based
- o Pull/Mesh-based

The Push/Tree-based solution creates an overlay network among Peers that has a tree shape [\[Castro03\]](#). 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 higher the number of received streams, the higher the video quality. A peer controls the video rate by either fetching or not fetching the streams delivered over 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., the set of participating peers). A peer forms a mesh overlay network with a subset of peers and exchanges 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 [Jacobson09b] [Dettl1] [Chai11] [NETINF]. In addition, Pull/Mesh-based P2PVS are more robust than the Push/Tree-based one [Magharei07], and the PPSP working group [IETF-PPSP] is also proposing a Pull/Mesh-based solution.

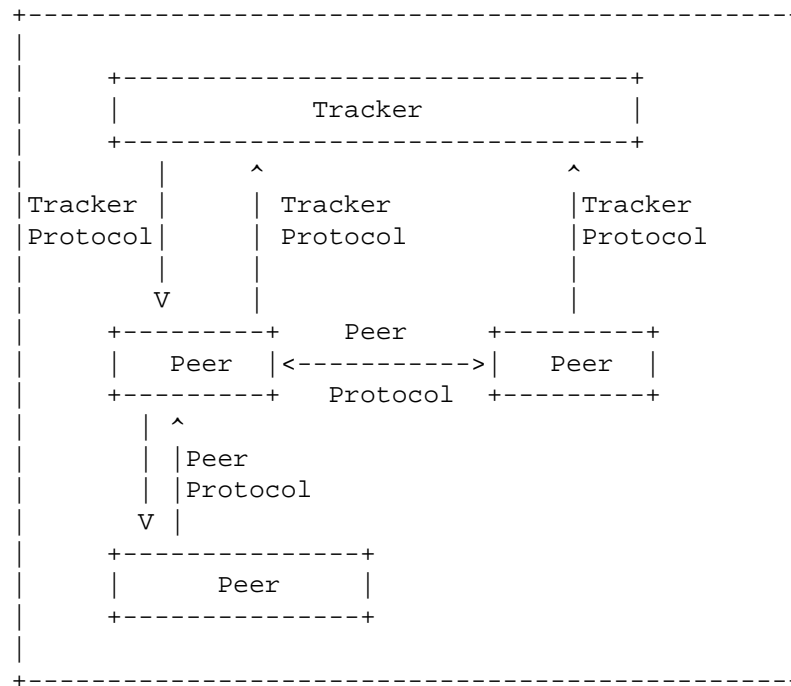


Figure 1: PPSP System Architecture [RFC6972]

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 standardizes 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

The Peer-to-Peer Streaming Peer Protocol (PPSPP) is defined in [Bakker15] and the Peer-to-Peer Streaming Tracker Protocol (PPSP-TP) is defined in [RFC7846].

Some of the operations carried out by the Tracker Protocol are the following: 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), and so on.

Some of the operations carried out by the Peer Protocol include 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 be sent back to 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 through an ICN, the messages should be wrapped as named-data items and receivers should request them by name.

A PPSP entity receives messages from peers and/or a tracker. Some operations require gathering the messages generated by another specific host (peer or tracker). For instance, if 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 latter. 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, such as fetching a DATA message (i.e., a video chunk), as long as a peer receives the requested content, it doesn't matter which endpoint generated the data. We refer to this information with the generic term "named-data".

The naming scheme differentiates named-data and located-named-data items. In the 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 "#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., an 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 "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 that provides a pull-based API. Accordingly, the mapping of PPSP pull-based procedure is quite simple. For instance, using the CCN architecture [Jacobson09b], a PPSP DATA message may be carried by a CCN DATA message and a REQUEST message can be transferred by a CCN Interest.

Conversely, the support of push-based PPSP operations may be more difficult. We need 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-way handshakes proposed in [Jacobson09a]. Another possibility is that receivers periodically send out request messages of the named-data that neighbors will push and, when available, the sender inserts the pushed data within a response message.

6.2.4. Abstract Layering for PPSP over ICN

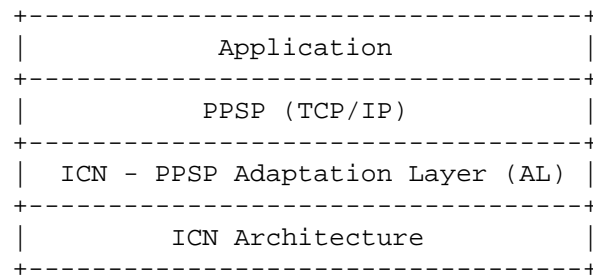


Figure 2: Mediator Approach

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 fact, the role the mediator is to use ICN to transfer PPSP legacy messages.

This approach makes it 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 an underlying TCP/IP API. Also, in this case, the mediator approach would make it possible to easily reuse both the PPSP libraries and the Application on top of an ICN.

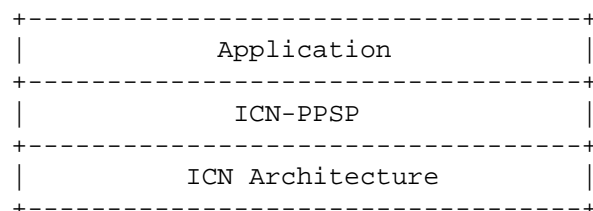


Figure 3: Clean-Slate Approach

Figure 3 sketches a clean-slate layering approach in which the application directly includes or interacts with a PPSP version based on ICN. It's likely such a PPSP_ICN integration could yield a simpler 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 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 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 [Jacobson09b]).

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 the 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 [Dettil3].

6.2.6. ICN Deployment for PPSP

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

In the isolated case, a PPSP session is supported by an instance of an ICN (e.g., deployed on top of an 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 a current P2P application, which usually forms an overlay network among peers of a P2P application; 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, thus 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 may have available only 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"), thus scaling down the number of entries from number of chunks to number of peers. However, in this case, the ICN mechanisms recognize the same video chunk offered by different peers as different content, thus losing caching and multicasting ICN benefits. In any case, routing entries should be updated either on the basis of the availability of named-data items on peers or on the presence of peers, and these events in a P2P session are rapidly changing and 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 significantly 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 an MPEG-DASH streaming with MPEG AVC encoding, the 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, and R3; for each segment S, we have three distinct files: S.R1, S.R2, and 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. Receiver-driven algorithms, implemented by the video client, usually handle the estimation of the best coding rate.

The independence among files associated with 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 from fetching video chunks from the latter since Peer B only has chunks available that are coded at a rate different from the ones needed by Peer A. To overcome this issue, a common distributed rate selection algorithm could force peers to select the same coding rate [Detti13]; nevertheless, this approach may be not feasible in the case of many peers.

The use of an SVC encoding (Annex G extension of the H.264/MPEG-4 Advanced Video Coding (AVC) video compression standard) should make rate adaptation possible while neither reducing peer collaborations nor the in-network caching effectiveness. For a single video chunk, an SVC encoder produces different files for the different rates (roughly "layers"), and these files are progressively related to each other. Starting from a base layer that 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), and 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 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 the 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 (see [ATIS-IIF]). 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 500 ms. To achieve this objective over the IP infrastructure, we have multiple choices:

- i receive fast unicast streams from a dedicated server (most effective but not resource efficient);
- ii connect to other peers in the network (efficiency depends on peer support, effective and resource efficient, if also supported with a dedicated server); and
- iii connect 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 (see [ATIS-IIF]). This suggests the perceived loss rate to be less than or equal to 10^{-7} . Current IP-based solutions rely on the following proactive and reactive recovery techniques: (i) joining the Forward Error Correction (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, therefore enabling earlier access to IPTV content.

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

- o 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 15 Mbps, we observe significant traffic upstream to bring those streams. As the number of streams increases (including the same session at

different quality levels and other formats), so does 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).

- o Cache control: As the IPTV content expires at a rapid rate (with a likely expiry threshold of 1 s), we need solutions to effectively flush out such content to also prevent degradation 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 the overhead required to maintain such content).
- o 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.

It should be noted that IPTV includes more than just multicast. Many implementations include "trick plays" (fast forward, pause, rewind) that often transform a multicast session into multiple unicast sessions. In this context, ICN is beneficial, as the caching offers an implicit multicast but without tight synchronization constraints in between two different users. One user may rewind and start playing forward again, thus drawing from a nearby cache of the

content recently viewed by another user (whereas in a strict multicast session, the opportunity of one user lagging off behind would be more difficult to implement).

8. Digital Rights Management in ICN

This section discusses the need for DRM functionalities for multimedia streaming over ICN. It focuses on two possible approaches: modifying Authentication, Authorization, and Accounting (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)) content 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 in sharing content with their families, friends, colleges, or others and have no intention making a profit. However, the second class of business model requires DRM, because these entities are primarily profit oriented. For example, large on-demand streaming platforms (e.g., 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 creator 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 ICNs.

8.1. Broadcast Encryption for DRM in ICN

This section discusses Broadcast Encryption (BE) as a suitable basis for DRM functionalities in conformance to the ICN communication paradigm (network-inherent caching, 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 from decrypting and consuming 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; 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 distributor. 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 BE [[Fiat94](#)] as proposed in [[Poschl3](#)]. 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. These schemes are responsible for the generation, exchange, storage, and replacement of cryptographic keys. The most interesting characteristics of BE schemes are:

- o BE schemes typically use 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.
- o 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.

- o 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.
- o 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 BE schemes.

This listing already illustrates why BE is suitable to control the access to data that is distributed via an ICN. BE enables the usage of a single session key for confidential data transmission between a dynamically changing subset of 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:

- o The updates of the session key can be delayed until a set of compromised secrets has been gathered. Note that secrets may become compromised because of two reasons: first, a traitor could have illegally revealed the secret; second, the subscription of an entity expired. Delayed revocation temporarily enables some illegitimate 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 trade-off. The longer the interval lasts, 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 have been encrypted with the prior session key, the publisher could indicate a lifetime for transmitted packets.
- o 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; therefore, no resources are wasted. Furthermore, if the updates are triggered in non-peak times, the maximum amount of resources needed at one point in time can be lowered effectively since in peak times generally more diverse content is requested.

- o 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. In regard to a lack of computation time or bandwidth, the cloud service could be used to scale up to overcome shortages.

Figure 4 shows the potential usage of BE in a multimedia delivery framework 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.

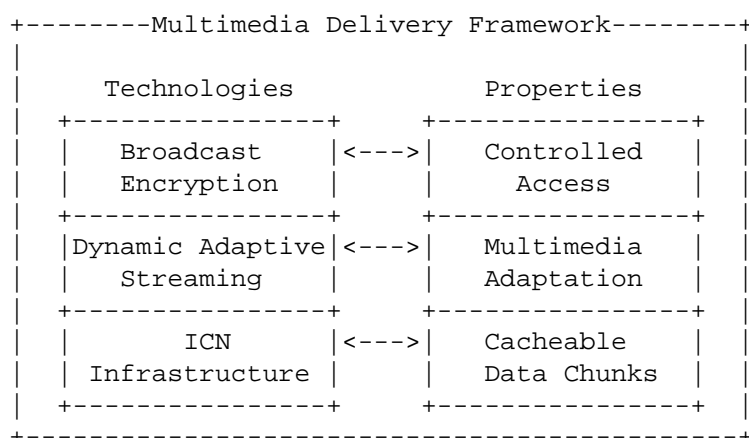


Figure 4: A Potential Multimedia Framework Using BE

8.2. AAA-Based DRM for ICN Networks

8.2.1. Overview

Recently, a novel approach to DRM has emerged to link DRM to usual network management operations, hence linking DRM to 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 ITU Carrier Code (ICC); 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 proposal of the BE, proposed earlier in this document. The BE proposes a technology, as this solution is more operational.

8.2.2. Implementation

With the proposed AAA-based DRM, when content is requested by name from a specific destination, the request could link back to both the content provider and the caching 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 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 (Media Access Control (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 ICN mechanisms that leverage Video Adaptation. 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

Distributing content, and video in particular, using local communications in large-scale events such as sporting events in a stadium, a concert, or a large demonstration, is an active area of investigation and a potential use case where ICN would provide significant benefits.

Such use cases involve 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 takes input from them to identify the key research issues. Real-time communications add timing constraints (both in terms of delay and in terms of synchronization) to the scenario discussed above.

An Access Router (AR) and a Virtual Router (VR), and immersive multimedia experiences in general, are clearly an area of further investigation, as they involve combining multiple streams of data from multiple users into a coherent whole. This raises issues of multisource, multidestination multimedia streams that ICN may be equipped to deal with in a more natural manner than IP, which is inherently unicast.

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 reuse, 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. Further, assume that the capacity of each of the links (both the wireless and the WAN logical links) varies 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.

If, for instance, during Time Period 1 the wireless capacity is 1 and the wired capacity is 2 and during Time Period 2 the wireless capacity is 2 (due to some hotspot) and the wired capacity 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 Time Period 2 there is 1 unit of data at the cache and another unit of data that can be streamed from the server; therefore, the rate that can be achieved is 2 units of data. In this case, the average bandwidth rises from 1 to 1.5 over the two periods.

This straw-man example illustrates 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 to take advantage of this gain. End-to-end rate adaptation cannot take advantage of the cache availability. The authors of [Rainer16] showed that buffer-based adaptation mechanisms can be one approach to tackle this challenge. As buffer-based adaptation does not estimate the available bandwidth resources (but solely considers the video buffer fill state), measured bandwidth fluctuations caused by cache hits are not existent. Therefore, they cannot negatively impact the adaptation decisions (e.g., frequent representation switching).

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 Worldwide Interoperability for Microwave Access (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 that best suits them according to different criteria, such as available bandwidth, battery consumption, access to different link conditions according to the user profile, or even access to different content. Ultimately, these aspects contribute to the QoE 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 [MEDIEVAL], where link events reporting on wireless conditions and available alternative connection points were combined with video requirements and traffic optimization mechanisms towards the production of a joint network and mobile terminal mobility management decision. Concretely, in [Fu13], 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 [Corujo12]).

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.

Enabling such scenarios requires further research, with the main points highlighted as follows:

- o how to develop 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;
- o 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;

- o how to develop the previous research items using intrinsic ICN mechanisms (i.e., naming and strategy layers);
- o how to leverage intelligent pre-caching of content to prevent stalls and poor quality phases, which lead to a worse QoE for 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 [Crabtree13]; and
- o how to take advantage of the multipath opportunities over the heterogeneous wireless interfaces.

9.5. Network Coding for Video Distribution in ICN

An interesting research area for combining heterogeneous sources is to use network coding [Montpetit13b]. Network coding allows for asynchronous combining of multiple sources by having each of them send information that is not duplicated by the other but that 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.

Network coding has shown promise in reducing buffering events in unicast, multicast, and P2P settings. [Medard12] considers strategies using network coding to enhance QoE for multimedia communications. Network coding can be applied to multiple streams, but also within a single stream as an equivalent of a composable erasure code. Clearly, there is a need for further investigation of network coding in ICN, potentially as a topic of activity in the research group.

9.6. Synchronization Issues for Video Distribution in ICN

ICN decouples the fetching of video chunks from their locations. This means an audio chunk may be received from one network element (cache/storage/server), a video chunk may be received from another, while yet another chunk (say, the next one, or another layer from the same video stream) may come from a third element. This introduces disparity in the retrieval times and locations of the different elements of a video stream that need to be played at the same (or almost same) time. Synchronization of such delivery and playback may require specific synchronization tools for video delivery in ICN.

Other aspects involve synchronizing:

- o within a single stream, for instance, the consecutive chunks of a single stream or the multiple layers of a layered scheme when sources and transport layers may be different.
- o re-ordering the packets of a stream distributed over multiple sources at the video client, or ensuring that multiple chunks coming from multiple sources arrive within an acceptable time window;
- o multiple streams, such as the audio and video components of a video stream, which can be received from independent sources; and
- o multiple streams from multiple sources to multiple destinations, such as mass distribution of live events. For instance, for live video streams or video conferencing, some level of synchronization is required so that people watching the stream view the same events at the same time.

Some of these issues were addressed in [Montpetit13a] in the context of social video consumption. Network coding, with traffic engineering, is considered as a potential solution for synchronization issues. Other approaches could be considered that are specific to ICN as well.

Traffic engineering in ICN [Su14] [Chanda13] may be required to provide proper synchronization of multiple streams.

10. Security Considerations

This is informational. There are no specific security considerations outside of those mentioned in the text.

11. Conclusions

This document proposes adaptive video streaming for ICN, identified potential problems, and presents 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 multilink and multisource 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, which are transparent to the actual DASH client.

12. References

12.1. Normative References

- [Rainer16] Rainer, B., Posch, D., and H. Hellwagner, "Investigating the Performance of Pull-based Dynamic Adaptive Streaming in NDN", IEEE Journal on Selected Areas in Communications (J-SAC): Special Issue on Video Distribution over Future Internet, Volume 34, Number 8, DOI 10.1109/JSAC.2016.2577365, August 2016.
- [RFC6972] Zhang, Y. and N. Zong, "Problem Statement and Requirements of the Peer-to-Peer Streaming Protocol (PPSP)", RFC 6972, DOI 10.17487/RFC6972, July 2013, <<http://www.rfc-editor.org/info/rfc6972>>.

12.2. Informative References

- [ATIS-IIF] "ATIS: IIF, IPTV Interoperability Forum", 2015, <<http://www.atis.org/iif/deliv.asp>>.
- [Bakker15] Bakker, A., Petrocco, R., and V. Grishchenko, "Peer-to-Peer Streaming Peer Protocol (PPSPP)", RFC 7574, DOI 10.17487/RFC7574, July 2015, <<http://www.rfc-editor.org/info/rfc7574>>.
- [Castro03] Castro, M., Druschel, P., Kermarrec, A., Nandi, A., and A. Rowstron, "SplitStream: High-Bandwidth Multicast in Cooperative Environments", Proceedings of the 19th ACM Symposium on Operating Systems Principles (SOSP '03), DOI 10.1145/945445.945474, October 2003.
- [Chai11] Chai, W., Wang, N., Psaras, I., Pavlou, G., Wang, C., de Blas, G., Ramon-Salguero, F., Liang, L., Spirou, S., Blefari-Melazzi, N., Beben, A., and E. Hadjioannou, "CURLING: Content-Ubiquitous Resolution and Delivery Infrastructure for Next Generation Services", IEEE Communications Magazine, Volume 49, Issue 3, DOI 10.1109/MCOM.2011.5723808, March 2011.

- [Chanda13] Chanda, A., Westphal, C., and D. Raychaudhuri, "Content Based Traffic Engineering in Software Defined Information Centric Networks", 2013 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), DOI 10.1109/INFCOMW.2013.6970717, April 2013.
- [Corujo12] Corujo, D., Vidal, I., Garcia-Reinoso, J., and R. Aguiar, "A Named Data Networking Flexible Framework for Management Communications", IEEE Communications Magazine, Volume 50, Issue 12, DOI 10.1109/MCOM.2012.6384449, December 2012.
- [Crabtree13] Crabtree, B., Stevens, T., Allan, B., Lederer, S., Posch, D., Mueller, C., and C. Timmerer, "Video Adaptation in Limited/Zero Network Coverage", CCNxCon 2013, Palo Alto Research Center (PARC), September 2013.
- [Detti11] Detti, A., Blefari-Melazzi, N., Salsano, S., and M. Pomposini, "CONET: A Content Centric Inter-Networking Architecture", Proceedings of the ACM SIGCOMM Workshop on Information-Centric Networking, DOI 10.1145/2018584.2018598, August 2011.
- [Detti12] Detti, A., Pomposini, M., Blefari-Melazzi, N., Salsano, S., and A. Bragagnini, "Offloading cellular networks with Information-Centric Networking: the case of video streaming", 2013 IEEE 14th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM), DOI 10.1109/WoWMoM.2012.6263734, June 2012.
- [Detti13] Detti, A., Ricci, B., and N. Blefari-Melazzi, "Peer-To-Peer Live Adaptive Video Streaming for Information Centric Cellular Networks", 2013 IEEE 24th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC), DOI 10.1109/PIMRC.2013.6666771, September 2013.
- [Fiat94] Fiat, A. and M. Naor, "Broadcast Encryption", Advances in Cryptology - CRYPTO '93 Proceedings, Lecture Notes in Computer Science, Volume 773, pp. 480-491, 1994.
- [Fu13] Fu, B., Kunzmann, G., Wetterwald, M., Corujo, D., and R. Costa, "QoE-aware traffic management for mobile video delivery", 2013 IEEE International Conference on Communications Workshops (ICC), DOI 10.1109/ICCW.2013.6649314, June 2013.

- [Grandl13] Grandl, R., Su, K., and C. Westphal, "On the Interaction of Adaptive Video Streaming with Content-Centric Networks", 2013 IEEE International Conference on Multimedia and Expo (ICME), DOI 10.1109/ICME.2013.6607500, July 2013.
- [IETF-PPSP] IETF, "Peer to Peer Streaming Protocol (ppsp)", <<https://datatracker.ietf.org/wg/ppsp/>>.
- [ISO-DASH] ISO, "Information technology -- Dynamic adaptive streaming over HTTP (DASH) -- Part 1: Media presentation description and segment formats", ISO/IEC 23009-1:2014, May 2014.
- [ITEC-DASH] "ITEC - Dynamic Adaptive Streaming over HTTP", DASH Research at the Institute of Information Technology, Multimedia Communication Group, Alpen-Adria Universitaet Klagenfurt, <<http://dash.itec.aau.at>>.
- [Jacobson09a] Jacobson, V., Smetters, D., Briggs, N., Plass, M., Stewart, P., Thornton, J., and R. Braynard, "VoCCN: Voice-over Content-Centric Networks", Proceedings of the 2009 Workshop on Re-architecting the Internet, DOI 10.1145/1658978.1658980, December 2009.
- [Jacobson09b] Jacobson, V., Smetters, D., Thornton, J., Plass, M., Briggs, N., and R. Braynard, "Networking Named Content", Proceedings of the 5th International Conference on Emerging Networking Experiments and Technologies (CoNEXT), DOI 10.1145/1658939.1658941, December 2009.
- [LeCallet13] Le Callet, P., Moeller, S., and A. Perkis, "Qualinet White Paper on Definitions of Quality of Experience", European Network on Quality of Experience in Multimedia Systems and Services, COST Action IC 1003, Version 1.2, March 2013.
- [Lederer13a] Lederer, S., Liu, Y., Geurts, J., Point, J., Lederer, S., Mueller, C., Rainer, B., Timmerer, C., and H. Hellwagner, "Dynamic Adaptive Streaming over CCN: A Caching and Overhead Analysis", 2013 IEEE International Conference on Communication (ICC), DOI 10.1109/ICC.2013.6655116, June 2013.

- [Lederer13b]
Lederer, S., Mueller, C., Rainer, B., Timmerer, C., and H. Hellwagner, "An Experimental Analysis of Dynamic Adaptive Streaming over HTTP in Content Centric Networks", 2013 IEEE International Conference on Multimedia and Expo (ICME), DOI 10.1109/ICME.2013.6607500, July 2013.
- [Magharei07]
Magharei, N., Rejaie, R., and Y. Guo, "Mesh or Multiple-Tree: A Comparative Study of Live P2P Streaming Approaches", IEEE INFOCOM 2007 - 26th IEEE International Conference on Computer Communications, DOI 10.1109/INFCOM.2007.168, May 2007.
- [Medard12] Medard, M., Kim, M., Parandeh-Gheibi, M., Zeng, W., and M. Montpetit, "Quality of Experience for Multimedia Communications: Network Coding Strategies", Laboratory of Electronics, Massachusetts Institute of Technology, March 2012.
- [MEDIEVAL] "MEDIEVAL: MultiMEDia transport for mobile Video Applications", 2010, <<http://www.ict-medieval.eu>>.
- [Montpetit13a]
Montpetit, M., Holtzman, H., Chakrabarti, K., and M. Matijasevic, "Social video consumption: Synchronized viewing experiences across devices and networks", 2013 IEEE International Conference on Communications Workshops (ICC), pp. 286-290, DOI 10.1109/ICCW.2013.6649245, 2013.
- [Montpetit13b]
Montpetit, M., Westphal, C., and D. Trossen, "Network Coding Meets Information-Centric Networking: An Architectural Case for Information Dispersion Through Native Network Coding", Proceedings of the 1st ACM Workshop on Emerging Name-Oriented Mobile Networking Design-Architecture, Algorithms, and Applications, DOI 10.1145/2248361.2248370, June 2013.
- [Mueller12]
Mueller, C., Lederer, S., and C. Timmerer, "A Proxy Effect Analysis and Fair Adaptation Algorithm for Multiple Competing Dynamic Adaptive Streaming over HTTP Clients", 2012 IEEE Visual Communications and Image Processing (VCIP), DOI 10.1109/VCIP.2012.6410799, November 2012.
- [NETINF] "NetInf: Network of Information", <<http://www.netinf.org>>.

- [Posch13] Posch, D., Hellwagner, H., and P. Schartner, "On-Demand Video Streaming based on Dynamic Adaptive Encrypted Content Chunks", Proceedings of the 8th International Workshop on Secure Network Protocols (NPSec '13), DOI 10.1109/ICNP.2013.6733673, October 2013.
- [RFC7476] Pentikousis, K., Ed., Ohlman, B., Corujo, D., Boggia, G., Tyson, G., Davies, E., Molinaro, A., and S. Eum, "Information-Centric Networking: Baseline Scenarios", RFC 7476, DOI 10.17487/RFC7476, March 2015, <<http://www.rfc-editor.org/info/rfc7476>>.
- [RFC7846] Cruz, R., Nunes, M., Xia, J., Huang, R., Ed., Taveira, J., and D. Lingli, "Peer-to-Peer Streaming Tracker Protocol (PPSTP)", RFC 7846, DOI 10.17487/RFC7846, May 2016, <<http://www.rfc-editor.org/info/rfc7846>>.
- [Su14] Su, K. and C. Westphal, "On the Benefit of Information Centric Networks for Traffic Engineering", 2014 IEEE International Conference on Communications (ICC), DOI 10.1109/ICC.2014.6883810, June 2014.

Acknowledgments

This work was supported in part by the European Community 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 document. 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.

Authors' Addresses

Cedric Westphal (editor)
Huawei

Email: Cedric.Westphal@huawei.com

Stefan Lederer
Alpen-Adria University Klagenfurt

Email: stefan.lederer@itec.aau.at

Daniel Posch
Alpen-Adria University Klagenfurt

Email: daniel.posch@itec.aau.at

Christian Timmerer
Alpen-Adria University Klagenfurt

Email: christian.timmerer@itec.aau.at

Aytac Azgin
Huawei

Email: aytac.azgin@huawei.com

Will (Shucheng) Liu
Huawei

Email: liushucheng@huawei.com

Christopher Mueller
BitMovin

Email: christopher.mueller@bitmovin.net

Andrea Detti
University of Rome Tor Vergata

Email: andrea.detti@uniroma2.it

Daniel Corujo
Instituto de Telecomunicacoes Aveiro

Email: dcorujo@av.it.pt

Jianping Wang
City University of Hong Kong

Email: jianwang@cityu.edu.hk

Marie-Jose Montpetit
MIT

Email: marie@mjmontpetit.com

Niall Murray
Athlone Institute of Technology

Email: nmurray@research.ait.ie