NetInf Live Video Specification
draft-ahlgren-icnrg-netinf-live-video-00

Abstract

This document specifies how the NetInf information-centric network service can be used for transport of live video streaming. To illustrate this it describes a prototype system that was developed to be used at "events with large crowds", e.g., sports events. The specification defines how the used video format is mapped to NetInf named data objects (NDOs). It also describe how NetInf messages are used to transfer the NDOs.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on July 2, 2016.

Copyright Notice

Copyright (c) 2015 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
1. Introduction

This specification documents how a prototype NetInf-based live video streaming system makes use of the NetInf information-centric network (ICN) service. The prototype has been field-tested and demonstrated at numerous occasions [Malik2015a] [Malik2015b]. The Android client software that is part of the prototype system has been made available on Sourceforge [1] as open source with an Apache 2.0 license.

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].

2. System Architecture

Figure 1 shows the architecture of the NetInf live video streaming system. Users can record and publish video streams at a live event and at the same time other users can watch the streams live. The architecture also facilitates streaming to a device anywhere on the internet so that users not present at the venue can also publish or play streams.
Recording and playing clients at the event venue can connect to the system using local WiFi or mobile internet (3G/4G). Before a client can start publishing or playing, it first connects to a NetInf router. Consequently this router acts as the first hop NetInf node for the client. Clients on local WiFi at the event venue connect to a NetInf router in the local access network. Clients on the internet connect to a NetInf router in the NetInf core network.
The NetInf core network hosts a Name Resolution Service (NRS). This service is responsible for resolving object names into locators. It also provides search function for the registered Named Data Objects (NDOs).

ICN employs ubiquitous caching. Therefore every NetInf router in the architecture is coupled with a local cache. These routers cache NDOs on-path and serve them to corresponding GET requests when there is a cache hit. This ensures that clients are served data from the local network (if the data is cached) and that the edge links (like the one between the NetInf access network and the NetInf core network as seen in Figure 1) are not choked with traffic.

3. Video Format

The transport of video chunks using NetInf described in this specification can be used for video with any type of encoding. As an example we describe the mpeg4/H.264 encoding used at the above referenced events.

When transporting video over an best effort network, such an ICN network, there are two main goals: to minimize the delay and to avoid rebuffering events. Unfortunately the means to achieve these goals are conflicting. Small buffers are needed to minimize delay, large buffers are needed to minimize the risk of rebuffering.

At the events referenced above the video generated by the recording device were encoded using H.264/AVC video format and had the mime type "video/avc". The video is split into chunks so that each chunk fits into an NDO. Chunking is done at I-frames in order to ensure that video chunks are independent of each other. The H.264 encoded chunks are packaged into MP4 containers. The frame rate and I-frame intervals are configurable. The parameters can be tweaked to achieve different tradeoffs between the playback latency and the NDO header overhead. The I-frame interval controls the size of the generated video chunks. A typical setting of 30 fps and 2 seconds, respectively, was used with the largest available 16:9 resolution less than 864x480, resulting in a video data rate of about 1 Mbps. Each chunk corresponds, in this case, to a video playout of 2 seconds. To avoid rebuffering the playing application buffers of video data was typically set to 10 seconds.

4. Video Transport using NetInf Protocol

This section describes how NetInf named data objects (NDOs) are created from the video chunks in the previous section, and how the NDOs are made available to clients using the NetInf protocol [I-D.kutscher-icnrg-netinf-proto] with extensions for subscriptions
using the NetInf SUBSCRIBE and NOTIFY messages. As described in the NetInf protocol specification, a NetInf NDO consists of the object’s data and, optionally, associated JSON-encoded meta-data.

Two kinds of NetInf NDOs are created for a video stream: one carrying the video chunks, and another carrying a "header", or manifest, that describes the stream of video chunks. The latter is described in the next section.

4.1. Video chunk NDOs

Each video chunk is turned into one NetInf NDO. The bytes of the video chunk are directly put into the data content of the NDO.

The following meta-data items MUST be supplied with video chunk NDOs:

chunk_number      The sequence number of the chunk in the video starting with zero. (Integer)

video_name        The name (ni hash [RFC6920]) of the video stream header NDO encoded with the URL-SAFE variant of BASE64 without padding. (ASCII string)

start_time        The starting time of the chunk in microseconds since the start of the video. (Long integer)

duration          The duration of the video chunk in microseconds. (Long integer)

signature         A digital signature for the chunk. The signature is an RSA signature on the SHA-256 hash of the content as defined in the PKCS #1: RSA Cryptography Standard. The result is BASE64 encoded without padding or wrapping, using the URL-SAFE encoding variant. (ASCII string)

4.2. Publishing video chunk NDOs

Video chunk NDOs are published using NetInf PUBLISH as described in the NetInf protocol specification [I-D.kutscher-icnrg-netinf-proto]. The JSON-encoded meta-data is supplied as a "meta" item of the "ext" parameter in the PUBLISH message.

In addition, a NetInf NOTIFY message MUST be sent for each successfully published video chunk. The NOTIFY message contains the ni URI [RFC6920] of the stream header NDO and the ni URI of the just published video chunk NDO. It also contains the sequence number and signature of the video chunk NDO.
The destination for both of these messages is normally a designated next-hop NetInf router for the client. How the client finds this destination is out of scope for this specification.

4.3. Receiving video chunk NDOs

Clients that wish to receive video chunks MAY register to receive the above notifications using the NetInf SUBSCRIBE extension. They will then receive the NetInf NOTIFY, and can then issue a NetInf GET to retrieve the newly published video chunk.

5. Video Manifest

This section describes the video stream header and how it is encoded and published as a NetInf NDO.

5.1. Video stream header NDOs

Each video stream MUST be described and identified by publishing a stream header NDO. It contains various information about the stream and a public key that can be used by clients to verify the authenticity of the video chunk NDOs. The hash part the NetInf "ni" name of the header NDO is used to name the video stream. That hash is included as meta-data of the video chunk NDOs.

The NDO data (content) MUST be constructed by concatenating the bytes of the following items in binary form:

**MAGIC** The ASCII character sequence "ENS0".

**Public key length** Four-byte integer with the length of the public key. (Note: needs to be network byte order/big endian, but the implementation likely uses little endian.)

**Public key** The public key for the stream encoded according to X.509 ASN.1 SubjectPublicKeyInfo.

**Contents** Any additional data provided by the client. (optional)

The following JSON-encoded meta-data MUST be included with header NDOs:

**title** A title for the video stream as provided by the client. (ASCII string)
The following JSON-encoded meta-data SHOULD be included with header NDOs:

- **stream-end**: The time of the end of the stream in Unix time in milliseconds. The NDO SHOULD be updated with this item when the recording has completed. (Long integer)
- **gps-x**: The X coordinate (longitude) of the location of the recorded video in degrees. (Double floating point number)
- **gps-y**: The Y coordinate (latitude) of the location of the recorded video in degrees. (Double floating point number)

### 5.2. Publishing video header NDOs

Video header NDOs are published using NetInf PUBLISH as described in the NetInf protocol specification [I-D.kutscher-icnrg-netinfproto]. The JSON-encoded meta-data is supplied as a "meta" item of the "ext" parameter in the PUBLISH message.

### 5.3. Browsing available video streams

Clients can use the NetInf SEARCH message to collect information about the published video stream headers, and with the result populate a browser listing in the user interface. The user can with that choose a video to play.

### 6. Security Considerations

TBD

### 7. Acknowledgements

The work behind and the writing of this document are supported by the activity ‘14082 Networks for Future Media Distribution’ within EIT Digital (formerly EIT ICT labs).
8. References

8.1. Normative References

[I-D.kutscher-icnrg-netinf,proto]


8.2. Informative References

[Dannewitz2013]

[Malik2015a]

[Malik2015b]

8.3. URIs


Authors’ Addresses