

PPSP  
Internet-Draft  
Intended status: Informational  
Expires: October 22, 2011

J. Seedorf  
M. Stiernerling  
NEC  
M. Mellia  
Politecnico di Torino  
R. Lo Cigno  
C. Kiraly  
University of Trento  
April 20, 2011

**Design Considerations for a Peer-to-Peer Streaming Protocol  
draft-seedorf-ppsp-design-considerations-02**

Abstract

Streaming video on P2P overlays puts extremely high demands and stress on the underlying network, especially in case of TV and live streaming. The EU research project NAPA-WINE has devised an overall architecture for live video streaming that supports the needs of the users and content providers, while being respective of network-level needs, as reducing inter-AS traffic using ALTO-like services. In this document, we describe generic elements of this software architecture for P2P live streaming and derive the corresponding implications for standardizing a Peer-to-Peer streaming protocol.

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 October 22, 2011.

Copyright Notice

Copyright (c) 2011 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. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

- [1.](#) Introduction . . . . . [3](#)
- [2.](#) An Architecture for a P2P-based Live Streaming Application . . [5](#)
  - [2.1.](#) Application Layer . . . . . [5](#)
  - [2.2.](#) Topology Management . . . . . [5](#)
  - [2.3.](#) Chunk Scheduling . . . . . [6](#)
  - [2.4.](#) Monitoring Layer . . . . . [7](#)
  - [2.5.](#) Messaging Layer . . . . . [7](#)
  - [2.6.](#) Interaction with ALTO . . . . . [8](#)
- [3.](#) Summary of Considerations for PPSP Protocol Design . . . . . [9](#)
- [4.](#) Conclusions . . . . . [10](#)
- [5.](#) Security Considerations . . . . . [11](#)
- [6.](#) Acknowledgements . . . . . [12](#)
- [7.](#) Informative References . . . . . [13](#)
- Authors' Addresses . . . . . [15](#)



## **1. Introduction**

In recent years, Peer-to-peer (P2P) technology became increasingly popular for video streaming applications, including TV services (P2P-TV). Examples of commercial P2P-TV include SopCast, TVAnts, PPLive, UUSee, and TVUplayer. The interest of the research and standardization communities, content providers and network operators is also on the rise. Content providers see a novel opportunity to reach users, but at the same time they are concerned about the threats posed to their standard business models. Network operators are mainly worried by the stress posed by such a bandwidth-hungry and delay sensitive application on their infrastructure. The research community is stimulated by the opportunities offered by live P2P distribution and broadcasting, looking both for novel technical solutions and innovative business models.

NAPA-WINE (Network Aware P2P-TV Application over WISE Networks) is a three years project (STREP) within the 7-th Research Framework of the European Commission whose goal is finding innovative solutions for P2P live streaming to meet opportunities envisaged by content providers while soothing worries of network operators [[refs.napawebpage](#)].

In a P2P-TV system, a source divides the video stream into chunks of data, which are exchanged among nodes to distribute them to all participating peers. Peers form an overlay topology at the application layer, where neighbor peers are connected and exchange chunks using the underlying IP network. The IP and overlay layer are both "network" layers in that they both perform routing and forwarding of the data: packets in the IP layer and chunks (normally a sequence of packets) in the overlay. In this context, the NAPA-WINE project has developed an innovative, network cooperative P2P architecture that explicitly targets the optimization of the quality perceived by the users while minimizing the impact on the underlying transport network. Our architecture does not impose any structure on the overlay topology, which can be any type of generic mesh. The video distribution is chunk-based, but chunk construction is free enough to accommodate anything from a single video frame (even less if required) to large swaths of a video in case of nearly on-demand applications. The focus is on the design of a system empowering future P2P High Quality TV, a system where a source peer produces the video stream, chops it into chunks, and injects them in the overlay where peers cooperate to distribute them, without the need for the source to have enormous resources and bandwidth to support the service.

In this document, we summarize our architecture for P2P live streaming (a more detailed description can be found in



[[refs.napa-architecture](#)]). The goal of this draft is to derive the implications for standardizing a P2P-based streaming protocol from the aforementioned architecture. We thus highlight key design considerations for a P2P-based streaming protocol which we believe are important input to the IETF PPSP working group.

An open-source implementation of this P2P live streaming architecture has been released, entitled "WINESTREAMER". The latest release of the WINESTREAMER software can be retrieved at [[refs.winestreamer](#)].

## **2. An Architecture for a P2P-based Live Streaming Application**

The architecture we developed is based on five main building blocks:

- o Application Layer
- o Topology Management
- o Chunk Scheduling
- o Monitoring Layer
- o Messaging Layer

In addition, our architecture contains an external ALTO interface that can support the topology management providing information that cannot be measured at the application level. In the following we briefly outline the role and key features of each building block.

### **2.1. Application Layer**

The Application Layer (or User Layer) is mainly responsible for of video encoding and its packaging into chunks (at the source) and de-chunkization and decoding at receivers. Standard encoding tools like ffmpeg can be used by the User Layer, whose goal is not implementing a proprietary video encoder, but supporting as many as possible standard formats (MPEG1/2/4, H.264, etc.). Depending on the type of video source this may include analog/digital conversion, encoding and any other video manipulation that the content provider wishes to do, like advertising introduction and similar.

The standardization of the application layer is out-of-scope for IETF PPSP work other than considerations on how to express and transport metadata information about the content.

### **2.2. Topology Management**

A P2P-TV client needs to communicate efficiently with other peers to receive and redistribute the huge amount of information embedded in a video stream. Information must arrive in nearly real-time and with small delay variation. The application goal is then to deliver all the video information to all peers in the system in the smallest possible amount of time. One of the key enabling factors is who are the peers to communicate with, i.e. the neighborhood selection.

The overlay network in P2P systems is the result of a distributed algorithm that builds and maintains the neighborhood at each peer. When a peer joins the system, it selects an initial set of neighbors,





then the set of neighbors of every node in the system is dynamically optimized over time. The bootstrapping phase can be helped by the source or a web server where the user selects a channel, which can behave as a bit-torrent like tracker. The maintenance of the topology is based on a gossiping protocol that enables discovery of peers in the overlay. Once peers are discovered, the optimal topology management is obtained through an topology manager which chooses which peers to connect to.

#### IMPLICATIONS ON STANDARDIZATION:

- o A tracker protocol is required that supports the goals of the topology management
- o The topology management algorithm can be standardized, but this would probably be beyond the scope of the PPSP WG
- o The definition of information about the employed topology management and the exchange as part of the PPSP WG can be standardized

### **2.3. Chunk Scheduling**

Strictly related to topology management is the problem of chunk trading. The goal of chunk trading is receiving the stream smoothly (and with small delay) and to cooperate in the distribution procedure.

Chunks transferring is the operation that most affects performance and network friendliness. It includes protocol and algorithmic problems. First of all, peers need to exchange information about their current status to enable scheduling decisions. The information exchanged refers to the state of the peer with respect to the flow, i.e., a map of which chunks are needed by a peer to smoothly playout the stream. This task means i) sending buffer maps to other nodes with the proper timing, ii) receiving them from other nodes and merging the information in the local buffer map data base, iii) negotiating if this and other information should be spread by gossiping protocols or not, and to which depth it should spread in the topology.

Besides the buffer map exchange, the signaling includes Offer/Request/Select primitives used to trade chunks. These messages can be piggybacked on chunks for efficiency. Another key protocol decision is about "Pushing" or "Pulling" information. A chunk is pushed when the peer owning the chunk decides to offer it to some other peer, while it is pulled when a peer needing the chunk requests it from another peer. From a theoretical point of view, as shown in



[[refs.opt-scheduling](#)], pushing is more effective. Regardless of the protocol and the signaling strategy used, the core of a scheduler is the algorithm to choose the chunks to be exchanged and the peers to communicate with. Many different strategies have been studied, including both fundamental algorithms and their adaptation to heterogeneous scenarios, multiple sub-streams etc.

#### IMPLICATIONS ON STANDARDIZATION:

- o The PPSP protocol design should allow to operate either with a push or pull regime
- o The PPSP protocol design should allow to select if push or pull is used in the PPSP system during runtime
- o The PPSP protocol design should allow to employ multiple chunk scheduling algorithms with the same protocol

### **[2.4.](#) Monitoring Layer**

Beside the information provided by e.g. an ALTO Server, both the chunk scheduler and the overlay manager can exploit timely information about the quality of the connectivity between peers collected in real time by monitoring modules. This includes, but is not limited to, the distance and the available bandwidth between two peers, or the presence of Network Address Translation (NAT). The Monitoring Layer may employ passive or active measurement. Passive measurements are performed by observing the messages that are exchanged between peers. Active measurements craft special probe messages which are sent to other peers at the discretion of the Monitoring Layer. Monitoring the network conditions is important for the peer-to-peer streaming application in order to judge the current state of the surrounding network environment

#### IMPLICATIONS ON STANDARDIZATION:

- o The PPSP protocol should allow the exchange of monitoring status information (e.g., in the spirit of "Do you see what I see?" [[refs.dywis](#)])
- o The PPSP protocol should support active and passive measurements between peers

### **[2.5.](#) Messaging Layer**

The Messaging Layer offers primitives to other modules for sending and receiving data to/from other peers. It provides an abstract interface to transport protocols (UDP/TCP) and the corresponding



service access points offered by the operating system by extending their capabilities and providing an application level addressing, i.e., assigning a unique identifier to each peer. For example, it provides the ability to send a chunk to another peer, which has to be segmented and then reassembled to fit into UDP datagrams. The messaging layer also provides an interface to the monitoring module invoked for passive measurements: whenever a message is sent or received an indication will be given to the monitoring module, which can update its statistics. The last important feature of the messaging layer is mechanisms for the traversal of NAT boxes.

#### IMPLICATIONS ON STANDARDIZATION:

- o The PPSP protocol should allow to negotiate or select different transport protocols, e.g., between plain TCP and LEDBAT.
- o The PPSP protocol or framework should support peers in NAT traversal.

## **2.6. Interaction with ALTO**

Application Layer Traffic Information (ALTO) [[refs.alto](#)] is an important means for improving the resource provider selection in P2P applications. The goal of an ALTO service is to provide applications with useful information (e.g. for P2P neighbor selection) which these applications cannot measure themselves [[RFC5693](#)]. Simulations have shown that the usage of information provided by an ALTO service can reduce operational costs associated with the transmission of P2P live streaming traffic [[refs.etm2010](#)]. The topology management in the NAPA-WINE architecture therefore fully supports ALTO guidance through the integration of an ALTO client within the topology manager. Further, the information retrieved via the integrated ALTO client can be used in neighbor selection, i.e. peers select the links to other peers in their neighbor list (partially) based on ALTO information.

#### IMPLICATIONS ON STANDARDIZATION:

- o The PPSP protocol should allow peers to interact with an ALTO server and to retrieve ALTO information.
- o The PPSP protocol should enable the use of ALTO information in peer selection.



### **3. Summary of Considerations for PPSP Protocol Design**

In this section we summarize the design considerations we derived from our experience in designing a P2P live streaming system and which we motivated in the previous section.

#### DESIGN CONSIDERATIONS FOR A PPSP PROTOCOL:

- o A tracker protocol is required that supports the goals of the topology management
- o The topology management algorithm can be standardized, but this would probably be beyond the scope of the PPSP WG
- o The definition of information about the employed topology management and the exchange as part of the PPSP WG can be standardized
- o The PPSP protocol design should allow to operate either with a push or pull regime
- o The PPSP protocol design should allow to select if push or pull is used in the PPSP system during runtime
- o The PPSP protocol design should allow to employ multiple chunk scheduling algorithms with the same protocol
- o The PPSP protocol should allow the exchange of monitoring status information (e.g., in the spirit of "Do you see what I see?" [[refs.dywis](#)])
- o The PPSP protocol should support active and passive measurements between peers
- o The PPSP protocol should allow to negotiate or select different transport protocols, e.g., between plain TCP and LEDBAT.
- o The PPSP protocol or framework should support peers in NAT traversal.
- o The PPSP protocol should allow peers to interact with an ALTO server and to retrieve ALTO information.
- o The PPSP protocol should enable the use of ALTO information in peer selection.





#### **4. Conclusions**

Video Streaming applications exploiting the P2P communication paradigm are a commercial reality, but their overall architecture is still biased by file-sharing applications and they operate without any coordination with the IP network, often resulting in poor, even wasteful resource usage. This will prevent them to support High Quality TV in the future, or to make the transition to High Definition, which will require several Mbit/s per peer.

This document presented the NAPA-WINE architecture for a P2P-TV system, which has been developed with the goal of efficiency and cooperation between the application and both the network operators and the content providers. Prototypes of the peers and system are already running on the Internet and are demonstrated at major venues [[refs.demo-iptcomm2010](#)] [[refs.demo-p2p2010](#)] [[refs.demo-infocom2011](#)].

In addition, an open-source implementation of the P2P live streaming architecture presented in this document has been released, entitled "WINESTREAMER". The latest release of the WINESTREAMER software can be retrieved at [[refs.winestreamer](#)].

The overlay topology management and the chunk scheduling of information have been identified as important features for the application to be network-friendly. The first function enables building efficient and rational overlay topologies that are correctly mapped on top of the transport network structure (e.g., considering minimal number of hops between neighbors, locality w.r.t. Autonomous Systems, etc.). The second function guarantees that the network capacity is exploited without waste (e.g., by minimizing retransmissions and pursuing an efficient distribution of chunks, etc.).

Based on our experience in designing and implementing a P2P live streaming system, we highlighted key implications for standardization. We believe that these key design considerations we derived based on our architecture will be important input to the IETF PPSP working group for standardizing a P2P live streaming protocol.



## **5. Security Considerations**

Security considerations will be detailed in future versions of this draft.

## 6. Acknowledgements

The authors would like to thank all the people participating in NAPA-WINE and contributing to the project success with their work and research.

Jan Seedorf, Marco Mellia, Renato Lo Cigno, and Csaba Kiraly are partially supported by the NAPA-WINE project (Network-Aware P2P-TV Application over Wise Networks, <http://www.napa-wine.org>), a research project supported by the European Commission under its 7th Framework Program (contract no. 214412). The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the NAPA-WINE project or the European Commission.

Martin Stiernerling is partially supported by the COAST project (COntent Aware Searching, retrieval and sTreaming, <http://www.coast-fp7.eu>), a research project supported by the European Commission under its 7th Framework Program (contract no. 248036). The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the COAST project or the European Commission.



## **7. Informative References**

[refs.napa-architecture]

Birke, R., Leonardi, E., Mellia, M., Bakay, A., Szemethy, T., Kiraly, C., Lo Cigno, R., Mathieu, F., Muscariello, L., Niccolini, S., Seedorf, J., and G. Tropea, "Architecture of a Network-Aware P2P-TV Application: the NAPA-WINE Approach", IEEE Communication Magazine, to appear.

[refs.opt-scheduling]

Abeni, L., Kiraly, C., and R. Lo Cigno, "On the Optimal Scheduling of Streaming Applications in Unstructured Meshes", IFIP Networking 2009.

[refs.demo-iptcomm2010]

Seedorf, J., Niccolini, S., Lo Cigno, R., and C. Kiraly, "Prototypical Implementation of ALTO Client and ALTO Server and Integration into a P2P Live Streaming Software", Demonstration, IPTComm 2010.

[refs.demo-p2p2010]

Abeni, L., Bakay, A., Biazzini, M., Birke, R., Leonardi, E., Lo Cigno, R., Kiraly, C., Mellia, M., Niccolini, S., Seedorf, J., Szemethy, T., and G. Tropea, "Network Friendly P2P-TV: The Napa-Wine Approach", Live Demonstration and Extended Abstract, IEEE P2P 2010.

[refs.demo-infocom2011]

Abeni, L., Bakay, A., Birke, R., Birke, R., Leonardi, E., Lo Cigno, R., Kiraly, C., Mellia, M., Niccolini, S., Seedorf, J., Szemethy, T., and G. Tropea, "WineStreamer(s): Flexible P2P-TV Streaming Applications", Live Demonstration and Extended Abstract, IEEE INFOCOM 2011.

[refs.dywis]

Miao, X., Schulzrinne, H., Singh, V., and Q. Deng, "Distributed Self Fault-Diagnosis for SIP Multimedia Applications", Springer Real-Time Mobile Multimedia Services, 2007.

[refs.etm2010]

Seedorf, J., Niccolini, S., Stiemerling, M., Ferranti, E., and R. Winter, "Quantifying Operational Cost-Savings through ALTO-Guidance for P2P Live Streaming", 3rd Workshop on Economic Traffic Management (ETM 2010), LNCS 6236.



## [refs.alto]

Peterson, J., Gurbani, V., Marocco, E., and et. al., "IETF ALTO WG charter",  
<https://datatracker.ietf.org/wg/alto/charter/>.

## [refs.winestreamer]

"Napa-Wine Winestreamer latest release", <http://www.napa-wine.eu/cgi-bin/twiki/view/Public/NapaWineShowRoom>.

## [refs.napawebpage]

"Napa-Wine Project Website", <http://www.napa-wine.eu>.

[RFC5693] Seedorf, J. and E. Burger, "Application-Layer Traffic Optimization (ALTO) Problem Statement", [RFC 5693](#), October 2009.





Authors' Addresses

Jan Seedorf  
NEC Laboratories Europe, NEC Europe Ltd.  
Kurfuersten-Anlage 36  
Heidelberg 69115  
Germany

Phone: +49 (0) 6221 4342 221  
Email: [jan.seedorf@neclab.eu](mailto:jan.seedorf@neclab.eu)  
URI: <http://www.nw.neclab.eu>

Martin Stiernerling  
NEC Laboratories Europe / University of Goettingen  
Kurfuersten-Anlage 36  
Heidelberg 69115  
Germany

Phone: +49 (0) 6221 4342 113  
Email: [martin.stiernerling@neclab.eu](mailto:martin.stiernerling@neclab.eu)  
URI: <http://ietf.stiernerling.org>

Marco Mellia  
Politecnico di Torino, Italy  
Corso Duca degli Abruzzi 24  
Torino 10129  
ITALY

Phone:  
Email: [mellia@tlc.polito.it](mailto:mellia@tlc.polito.it)  
URI:

Renato Lo Cigno  
University of Trento  
Via Sommarive 14  
Povo 38123  
ITALY

Phone:  
Email: [locigno@disi.unitn.it](mailto:locigno@disi.unitn.it)  
URI:



Csaba Kiraly  
University of Trento  
Via Sommarive 14  
Povo 38123  
ITALY

Phone:

Email: [kiraly@disi.unitn.it](mailto:kiraly@disi.unitn.it)

URI: