Design Considerations for a Peer-to-Peer Streaming Protocol

draft-seedorf-ppsp-design-considerations-01

Jan Seedorf, Martin Stiemerling
NEC Laboratories Europe, Heidelberg, Germany

Marco Mellia
Politecnico di Torino, Italy

Csaba Kiraly, Renato Lo Cigno
University of Trento, Italy
Introduction and Overview

The EU Research Project NAPA-WINE has designed and implemented a network-aware P2P Live Streaming System

- Design of Software Architecture and Chunk Scheduling Algorithms
- Open Source Implementation (currently running large-scale trials)

Goal of our draft:

- Derive the corresponding implications for standardizing a Peer-to-Peer streaming protocol
- Based on our design and implementation experience
Agenda

- Background and Motivation: EU Project NAPA-WINE
- P2P Live Streaming Architecture
- Implications on Standardization
THE NAPA-WINE VISION
NAPA-WINE Vision

Network-cooperative P2P architecture

- Cooperation between network providers and P2P applications
  - for P2P-TV or live-streaming in general
- Explicitly targeting the optimization of the quality perceived by the users while minimizing the impact on the underlying transport network
  - empowering future P2P High Quality TV
P2P LIVE STREAMING
ARCHITECTURE
High-Level Architecture Overview

- Design of core functional blocks of a generic peer
- Distributed monitoring function
  - Allows the application to continuously gather real time information on network conditions
  - used to trigger reconfiguration of the overlay or to drive chunk scheduling
Detailed Architecture Overview

User Module
- Content Ingestion
- Player
- Control Interface

Overlay Module
- Neighbour set
- Topology controller

Scheduler Module
- Chunk buffer
- Active peers’ InfoBase

Monitoring Module
- Monitoring Controller
- Pas. meas
- Act. meas

Messaging Layer + NAT/FW traversal
- IPv4 / IPv6 + UDP / TCP / SCTP / ...

© NEC Corporation 2010
Topology Management and Chunk Trading

- **Neighbourhood database (green)**
  - Links to other peers
  - Updated dynamically

- **Chunk scheduling (yellow)**
  - Local chunk buffer
  - Chunk map exchange
  - Push vs. Pull

- **Topology management (blue)**
  - Gossiping with other peers

![Diagram showing the interaction between Neighbourhood, Chunk scheduling, and Topology management.]

- **Buffermaps**
- **Chunk Buffer**
- **Select Chunk**
- **Select Peer**
- **Push or Pull Send / Receive**
- **Topo Algorithm**
- **Buffer Map Exchange**
- **Gossiping Strategy**

**User Module**

**Repositories**

**Messaging Layer**

- Inter-peer communication
- Local control

© NEC Corporation 2010
Status Quo of Implementation

All features and logical blocks have been implemented

Development Toolkit GRAPES
- Generic Resource Aware P2P Environment for Streaming
- entirely written in C
  - reuse and linking with any language

Several P2P live streamers have been implemented
- exploring different algorithms and techniques

For QoE evaluation, PSNR (Peak Signal to Noise Ratio) Tools have been developed

→ www.napa-wine.eu
IMPLICATIONS ON STANDARDISATION
Design Consideration for a PPSP Protocol (1)

1. Topology Management
   - Tracker protocol must support goals of topology management
   - Standardizing topology management algorithm?
     - probably out-of-scope for PPSP
     - Definition of information about the employed topology management and the exchange may be standardized within PPSP WG

2. Chunk Scheduling
   - The PPSP protocol design should allow to operate either with a push or pull regime
     - Selection of push or pull being used in the PPSP system during runtime
     - probably out-of-scope for PPSP
   - The PPSP protocol design should allow to employ multiple chunk scheduling algorithms with the same protocol
Design Consideration for a PPSP Protocol (2)

Monitoring Layer
- The PPSP protocol should allow the exchange of monitoring status information among peers

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

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

EU project NAPA-WINE designed and implemented a P2P Live Streaming Architecture

- Composed of different layers with dedicated functionality
- Overlay topology management and chunk scheduling are key components of P2P Live Streaming clients

draft-seedorf-ppsp-design-considerations-01

- Describes the NAPA-WINE architecture
- Highlights key implications for standardization, based on our experience in designing and implementing a P2P live streaming system

We believe that these key design considerations we derived based on our architecture will be important input to the PPSP working group for standardizing a P2P live streaming protocol
This work was partially supported by NAPA-WINE, 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.
Empowered by Innovation

NEC
APPENDIX: SCREENSHOTS OF THE NAPA-WINE SOFTWARE
Screenshots (1)
Screenshots (2)
Screenshots (4)