Automatic measure of Quality of Experience with the PSQA approach

Gerardo Rubino

INRIA, Rennes,
France.

18th ITCSS, Karlskrona, Sweden, May 30, 2008

Outline

Outline

Overview

Measuring QoE

We develop a measuring technology (PSQA) able to

- quantify,
- automatically,
- the quality of audio/video systems,
- as perceived by the users,
- accurately,
- and in real-time if necessary;
- moreover, the technique has nice mathematical properties that can be exploited for standard performance and dependability modeling

Overview (cont'd)

Achievements

- PSQA for audio/video,
- for non-real-time and for interactive applications
- an application in network monitoring
- another "federating" application: use of PSQA in network design; area: P2P for live video

Our Context

Context

A video delivery reference service: www.adinetTV.com.uy

- Scalability problems due the bandwidth cost.
- There are no Quality assurance mechanisms.

We know the users' behavoir (log files) of this service.
AdinetTV is a Content Delivery Network (CDN). We want to extend it with a Peer-to-Peer (P2P) system.

Our Context

Problem

To offer the quality needed by the clients in a highly varying environment:

- Peers connect and disconnect very frequently, in an autonomous and completely asynchronous way.
- The perceived quality, the ultimate target, is difficult to measure accurately in real–time.
- The resources in the network grow with the popularity (scalability).

Our Context

Our approach

Design of a P2P-based system for live video distribution.
Divide and conquer design:

- PSQA for automatic perceived quality assessment;
- a centralized control approach using a meta-heuristic algorithm to mantain a robust structured P2P;
- delivery through a multi-source streaming approach: an optimization technique to maximize the expected Quality, as a way of facing the problem of the high peers dynamics;
- all the developments using open source code (VideoLAN player,...).

Our Contributions

Summary of the contributions in this dissertation

We can classify the main contributions of this work into the following points:

1. Quality of Experience
2. Multi-source Distribution using a P2P Approach
3. Efficient Search in Video Libraries
4. Quality-driven Dynamic Control of Video Delivery Networks

1/4: Quality of Experience

PSQA: Pseudo–Subjective Quality Assessment

- Originally developed for streaming video
- Very accurate
- Extended in many directions:
  - also for audio flows
  - also for interactive comms

Contributions in this area

- In-depth study of the PSQA methodology for video quality assessment
- Effects of failures on the perceived video quality, in particular the video frame loss effect, instead of the impact of packet losses (studied in all previous works)
- Impact of video's motion on quality

2/4: Multi-source Distribution using a P2P Approach

QoE based transmission design

Application of our video quality assessment methodology in network transmission design

Contributions in this area

- A generic multi-source streaming technique for networks with high probability of failures (such as P2P systems) and very low signalling overhead (in contrast with Bittorrent-like approaches)
- A distribution scheme that ensure a high QoE for end users when servers fail
- A specific streaming algorithm that maximizes the QoE based on the heterogeneous peers' lifetimes

3/4: Efficient Search in Video Libraries

Content discovery

The problem of the discovery of very dynamic content can not be solved with traditional techniques, like publications by video podcast or broadcatching

Contributions in this area

- In-depth study of search caching for Video on Demand (VoD) and MyTV complementary services
- Analysis of different caching strategies
- An optimal strategy that maximizes the number of correct answers to queries subject to bandwidth limitations

4/4: Quality-driven Dynamic Control of Video Delivery Networks

QoE based control design

Use of the PSQA technology to evaluate the perceived quality of the stream in real-time, in order to control or simply to monitor the system

Contributions in this area

- Design, implementation and validation of a generic monitor suite
- A centralized tree-based overlay topology for our P2P system, designed in order to diminish the impact of peers disconnection on quality

Outline

Quality of Experience vs Quality of Service

Quality of Experience

QoE is the overall performance of a system from the users' perspective.

- subjective measure
- end-to-end performance
- at the service level

Quality of Service

QoS is related to objective measures of performance at the network level and from the network point of view.

Perceived Quality

Perceived Video Quality is the main component of the QoE in video delivery services.

Quality of Experience vs Quality of Service

Quality of Experience

QoE is the overall performance of a system from the users' perspective.

- subjective measure
- end-to-end performance
- at the service level

Quality of Service

QoS is related to objective measures of performance at the network level and from the network point of view.

Perceived Quality

Perceived Video Quality is the main component of the QoE in video delivery services.

Factors Affecting the Perceived Video Quality

Factors that affect quality

Distribution (or network) parameters (loss rate, delay, jitter, retransmission,. . .)

Source / Receiver parameters (original video signal, codec, redundancy / buffer size,. . .)

Environment parameters (ambient noise, equipment quality,. . .)

Remarks

- We will ignore environment–related factors (we cannot control them).
- In a P2P system (over Internet), the loss rate is the most important factor due the peers disconnections.

Factors Affecting the Perceived Video Quality

Factors that affect quality

Distribution (or network) parameters (loss rate, delay, jitter, retransmission,. . .)

Source / Receiver parameters (original video signal, codec, redundancy / buffer size,. . .)

Environment parameters (ambient noise, equipment quality,. . .)

Remarks

- We will ignore environment–related factors (we cannot control them).
- In a P2P system (over Internet), the loss rate is the most important factor due the peers disconnections.

But. . . What Is the Quality of a video sequence?

Quality is a very subjective concept

- Difficult to provide a good definition, let alone a good estimation.
- We want a mean value.

The best way to evaluate it, is to ask the users

- Several normalized subjective assessment methods: ITU-R BT.500–10, draft ITU-R BT.700, DSL Forum WT-126
- We ask a group of people to rate the quality according to their own assessment, and we get a Mean Opinion Score (MOS).

Subjective Quality Assessment: Pros and Cons

Subjective assessment provides the real quality values

Indeed, the users ultimately decide what the quality is.
Standardized definition.

However. . .

- Expensive in manpower and time–consuming.
- Not automatic, not real–time.
- Useless for controlling purposes.

Objective Quality Assessment

In order to avoid the problems of subjective assessment

Objective assessment techniques, such as PSNR, VQM, MPQM, CMPQM, NVFM,. . . (and countless other fancy acronyms.)

- Algorithms and/or formulas (generally signal processing algorithms).
- Compute a sort of distance between the received sequence and the original one.

Objective Quality Assessment: Pros and Cons

Objective methods solve some issues with subjective assessment

- Cheap and fast.
- Automatic, possible for controlling purposes.

However. . .

- Generally, do not correlate well with human quality perception.
- Generally, it needs the original sequence $\Longrightarrow$ useless for real–time applications.

PSQA: Pseudo–Subjective Quality Assessment

Goals of PSQA

PSQA aims to provide quality assessments:

- as perceived by the user,
- accurately[a]
- automatically
- efficiently (in particular, in real time if needed)
- can be applied to several media types, under different networks and conditions.

[a] PSQA provides a value close enough to the *average* value that *would* be obtained from a panel of human observers.

PSQA Methodology

How does it work?

By learning the relation between some quality–affecting parameters, and quality itself.

PSQA in 3 stages

1. Quality–affecting factors and Distorted Video Database Generation
   - quality–affecting parameters selection
   - distorted video database generation
2. Subjective Quality Assessment
   - test campaign
   - Mean Opinion Score (MOS) calculation
3. Learning of the quality behavior with a statistical estimator
   - train and validate the estimator with the test results

PSQA Methodology

On the estimator used. . .

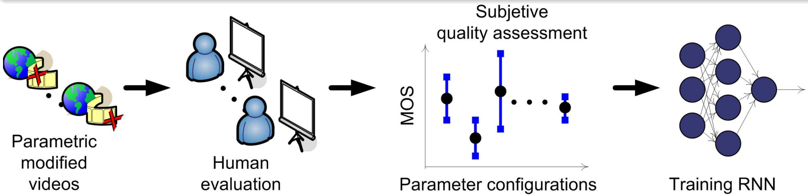
We implement PSQA with Random Neural Networks (RNNs).

Remarks. . .

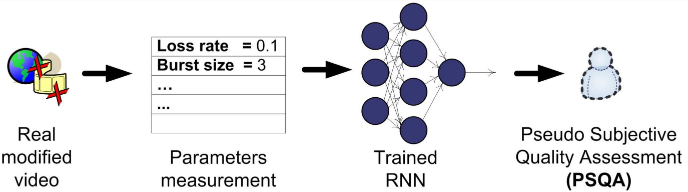
- at the beginning of the process, we must choose the parameters
- PSQA is specific to a type of network and/or application
- need a testbed:
  - to validate the quality–affecting parameters and
  - to generate the video database

The PSQA Process in a Picture

PSQA Training: only once!



Operation mode: very simple. . .



Using PSQA to Understand Video Quality

How does quality react. . .

- To an increase in loss rate?
- To the motion of the source video?
- To the addition of redundancy in the sender?
- To an increase of buffering in the receiver?
- To a combination of the points above!. . .

We have used PSQA

To answer these questions and others, under two different contexts.

Our PSQA Functions

“Simple” Function (used in some theoretical studies)

- MPEG-2 encoding
- 100 video sequences
- test made by five experts
- first study made with “frame level” parameters
- only distribution-oriented parameters considered

“Complex” Function (used in our GOL!P2P prototype)

- MPEG-4 (Xvid) encoding
- 204 video sequences
- test made by ten experts
- at “frame level”, discriminating frame type: I,P and B
- source–oriented and distribution-oriented parameters

Our PSQA *simple function*: loss rate distribution

Two parameters

two network-oriented input variables (that is, we fixed the characteristics of the stream, such as bandwidth, encoding,. . .):

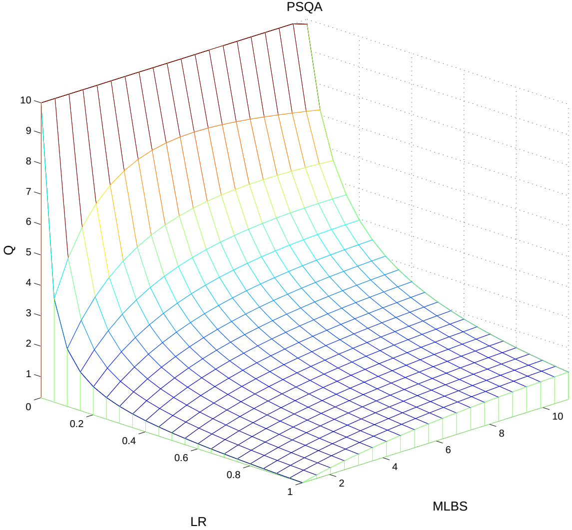
- the frame loss rate, denoted by *LR*
- the mean size of the bursts of frame losses, denoted by *MLBS*

We consider. . .

- *LR* from 0.0 to 0.2 (quality is too bad after 20% of losses)
- *MLBS* from 1 to 10 frames

Our PSQA *simple function*: loss rate distribution

$LR, MLBS \mapsto$ quality



- observe the monotonicity of Q with LR and $MLBS$
- in particular, the worst quality corresponds to the value $MLBS = 1$
- observe the less sensitivity of Q w.r.t. $MLBS$

Our PSQA *complex function*: frame types

Five parameters

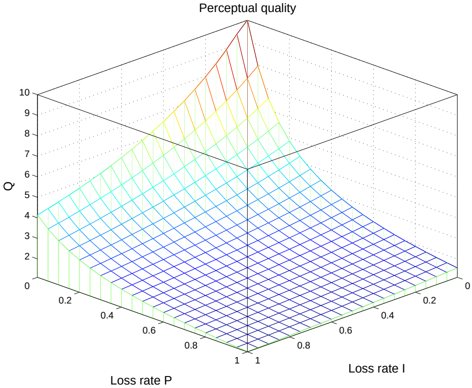
- network–oriented parameters: frame losses by type LR_I, LR_P, LR_B
- source–oriented parameters: the video motion (different metrics tested)
  GOP size and *frames P information ratio*

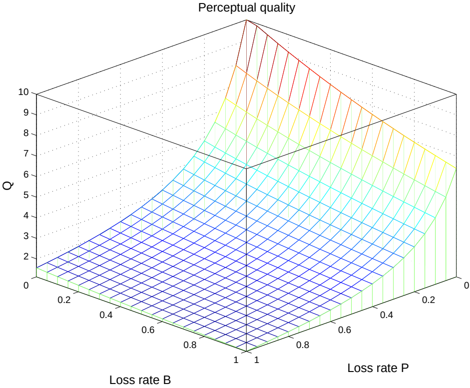
We consider...

- LR_I from 0.0 to 1.0
- LR_P and LR_B from 0.0 to 0.25
- *GOP size* from 25 to 350 frames
- *frames P information ratio* from 0.05 to 0.9

Our PSQA *complex function*: frame types

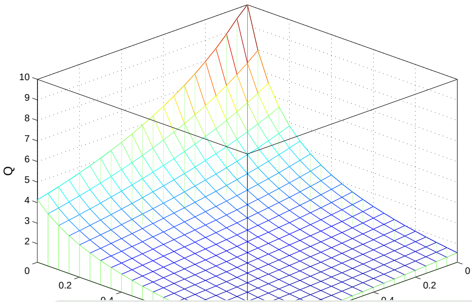
$$LR_I, LR_P, LR_B, \text{motion} \mapsto \text{quality}$$

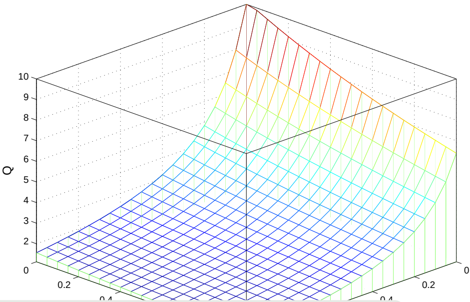




Our PSQA *complex function*: frame types

$$LR_I, LR_P, LR_B, \text{motion} \mapsto \text{quality}$$





- observe the monotonicity of Q w.r.t. LR's
- quality degrades quickly with LR_I and LR_P
- the impact of LR_P is a bit higher than for LR_I
- quality degrades slowly with LR_B

Outline

1. Introduction
2. Video Quality Assessment
3. **GOL!P2P Prototype**
4. Conclusions and Perspectives

P2P Network Communication

Exchanged data

- content (files, videos,...)
- control and routing (publications, searchs, connections/disconnections,...)

Methods to exchange data

- client/server
- hierarchically
- completely distributed

- if both (content and control) are distributed then the network is called pure, otherwise the network is call hybrid
- *usually pure networks do not scale well*

Control/Routing Layer: Overlay Network

Definition

The Overlay Network is a directed graph. The nodes are the peers. If a participating peer knows the location of another peer, then there is a directed edge from the former node to the latter.

P2P classification based on how the overlay is constructed:

- unstructured
- structured
  - tree-based (efficient transmission low signalling overhead)
  - mesh-based (good resilience to peer failures)

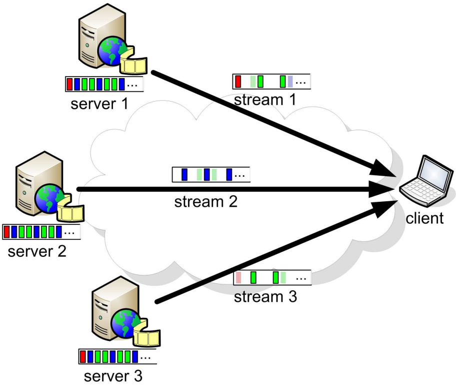
Content Delivery: Data Download

File Sharing (and Video on Demand)

- Bittorrent-like protocols

Live-video

- single source (streaming)
- multi–source:
  - Multiple Description Coding,
  - Network Coding,
  - ...



Gol!P2P Homepage

http://p2ptv.gforge.inria.fr

- Home
- Motivation
- Architecture
- Publication
- Members
- Software

GOLP2P - Peer to Peer Streaming Network

GOLP2P is a peer to peer distribution system, capable of distributing high-bandwidth live-content to all network peers preserving its quality. This project follows the multi-source approach where the stream is decomposed into several flows sent by different peers to each client. In order to meassure the peers perceived quality, it is used the recently proposed PSQA (Pseudo-Subjective Quality Assessment) technology.

Facultad de Ingeniería - Universidad de la República - Uruguay
IRISA Institut - Université de Rennes 1 - France

Our Gol!P2P project

Main design choices of GOL!P2P

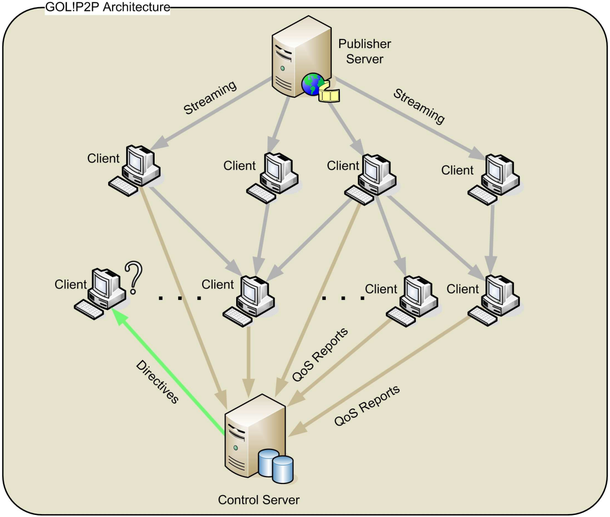
1. An hybrid P2P network with centralized control and distributed delivery
2. The quality perceived by each user is audited in real–time using PSQA.
3. It uses a simple tree-based structured overlay network
4. With a multi–source streaming technique

All the developments using open source code (VideoLAN player,. . .).
Prototype tested in PlanetLab.

P2P Network Model: Architecture

Components

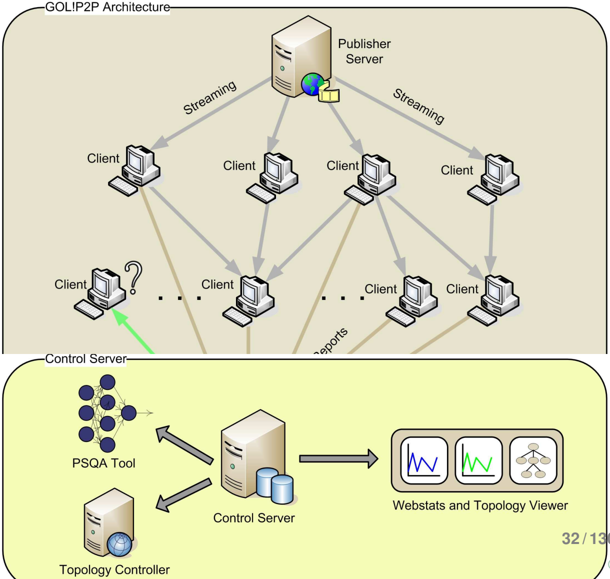
- Broadcaster Server(s).
- Peers.
- Control Server.



P2P Network Model: Architecture

Components

- Broadcaster Server(s).
- Peers.
- Control Server.



P2P Network Model: Streaming

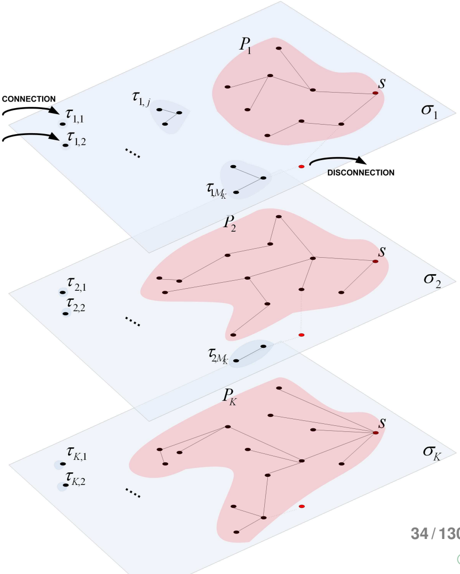
Architecture

- In a P2P system, nodes are clients and also servers.
- One broadcaster node *s* for the original stream.
- Stream is decomposed into K different substreams $\sigma_1, \sigma_2, \ldots, \sigma_K$ encoded with constant bitrate bw_k.
- A peer (acting as client) receives $\sigma_1, \sigma_2, \ldots, \sigma_K$ from K different peers (acting as servers) and reconstructs original stream.
- Quality of the reconstructed stream depends on which substream arrives (streaming scheme may allow for some redundancy).
- A peer may be a server for other peers, sending one or more σ_k, depending on its upload bandwidth capability BW^{out}.

P2P Network Model: System Dynamics

System Dynamics

- Some nodes leave, possibly disconnecting other clients in some substreams.
- Some nodes enter the network requesting for connection.
- Network reconfigured at discrete points in time, to reconnect disconnected nodes and to connect new arrivals.



But. . . Which nodes are assigned closer to the root?

Robust Assignment Model

- To decide which client will serve another one, some degree of intelligence and knowledge about the peers and the network state is needed
- Robust Design: minimizing the impact of the peers behavior on perceived quality
- We formalize the reconnection procedure with a Mathematical Programming Model

Looking for a Solution

- Model in general not tractable.
- Three centralized heuristic solutions:
  - Two Greedy algorithms and a GRASP metaheuristic.

Using the Solution for a Hybrid Structured P2P service

Comparison on a real-life based scenario

- We compare P2P vs. traditional Content Distribution Network (CDN) architectures.
- Case study based on simulation, scenario generated from statistical data from AdinetTV (10000 different users/month, 100 concurrent users per live-TV channel on average).

	CDN	P2P
Mean QoE	10	9.66
Servers BW	50 Mbps	5.6 Mbps
Clients BW	0 Mbps	0.6 Mbps/client

Tree-based Overlay Remarks

Remarks...

- PSQA allows to address the “ultimate target”, the QoE, in a quantitative way.
- P2P multi-source schemes greatly improve the scalability of video streaming solutions, diminishing bandwidth requirements at servers.
- Dynamic nature of P2P network introduces degradations in users’ QoE.
- Optimization models and metaheuristic algorithms allow to design rationally the P2P network connectivity, minimizing the impact of the peers behavior on perceived quality.

In-depth Technology: The Multi-Source Streaming

Design Questions. . .

1. Is the multi-source streaming technique is useful?
2. In a very dynamic peers context:
   Can we ensure some video quality level using the multi-source streaming technique?
3. How can we take advantage of the peers' heterogeneity?

We have used Markov Models and PSQA Functions

To answer these questions and others. . .

Is the multi-source streaming technique is useful?

Brief Answer

Yes, if we use some level of redundancy in the substreams.

Long Answer...

- Peers are independent and homogenous (in failures).
- We model the failure process on each server with a simplifed Gilbert model.
- We compare three extreme multi-source streamings policies: *single*, *copy* and *split*.
- We compute the perceived quality of each policy with the PSQA *simple function*: $Q = f\,(LR, MLBS)$
- With high burst sizes ($MLBS = 4$), the *split* policy behaves worse than the *single* policy (quality increases when losses are concentrated).

Is the multi-source streaming technique is useful?

Brief Answer

Yes, if we use some level of redundancy in the substreams.

Long Answer…

- Peers are independent and homogenous (in failures).
- We model the failure process on each server with a simplifed Gilbert model.
- We compare three extreme multi-source streamings policies: *single*, *copy* and *split*.
- We compute the perceived quality of each policy with the PSQA *simple function*: $Q = f(LR, MLBS)$
- With high burst sizes ($MLBS = 4$), the *split* policy behaves worse than the *single* policy (quality increases when losses are concentrated).

Can we ensure some video quality level?

Brief Answer

Yes, statistically. Moreover, we determine the number of servers K to ensure a given quality.

Long Answer...

- Peers are independent and homogenous (in failures).
- The network reconfigurates every T units of time.
- We model the failure process of the set of servers with a pure-death Markov chain.
- We use a *redundant split* streaming policy (sending information twice).
- We compute the perceived average quality at the client with the PSQA *simple function*.

Can we ensure some video quality level?

Brief Answer

Yes, statistically. Moreover, we determine the number of servers K to ensure a given quality.

Long Answer…

- Peers are independent and homogenous (in failures).
- The network reconfigurates every T units of time.
- We model the failure process of the set of servers with a pure-death Markov chain.
- We use a *redundant split* streaming policy (sending information twice).
- We compute the perceived average quality at the client with the PSQA *simple function*.

How can we take advantage of the peers' heterogeneity?

Brief Answer

Improving the multi-source streaming technique, sending the most important data from the most reliable peers.

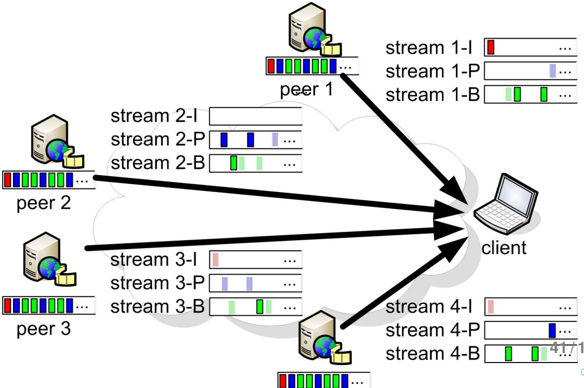
How can we take advantage of the peers' heterogeneity?

Brief Answer

Improving the multi-source streaming technique, sending the most important data from the most reliable peers.

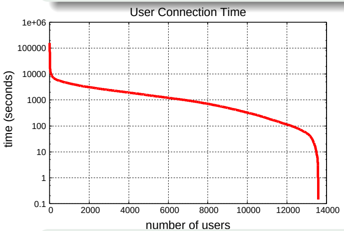
Improves

- discrimination per frame type
- unequal splitting



Classifying the clients

- we group clients in K clusters with the same number of clients each
- a *client peer* receives the video stream from one *server peers* of each cluster



the connection time of server k is exponentially distributed, with parameter λ_k, with the order $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_K$ (the best at the end)

- $N_k(t)$ is the binary r.v. equal to 1 iff server k is connected at t, and $\vec{N}(t)$ is the vector $\vec{N}(t) = (N_1(t), \cdots, N_K(t))$
- therefore, the probability of each *configuration* $\vec{n} \in \{0,1\}^K$ is $\Pr(\vec{N}(t) = \vec{n}) = \prod_{j:n_j=1} e^{-\lambda_j t} \prod_{j:n_j=0}(1 - e^{-\lambda_j t})$.

Improving the Multi-Source Technique

Unequal Splitting

- server k sends a fraction y_k of the stream
- we set $y_k = \gamma^{k-1} y_1$, with $y_1 = (\gamma - 1)/(\gamma^K - 1)$ and $\gamma > 1$
- if $\gamma = 1$ then a equal distribution (like the *split*)
  if $\gamma = 2$ then a exponential distribution
- we add redundancy to the global flow, $r \in [0, 1]$; $r = 0$ means no redundancy, $r = 1$ means that any frame is sent twice (as in the *redundant split* method)
- proportional redundancy distribution with the weights y_k
- each frame is sent either once or twice (no more than twice)

Discrimination per frame type

We have γ's and r's for each frame type.

Improving the Multi-Source Technique

Unequal Splitting

- server k sends a fraction y_k of the stream
- we set $y_k = \gamma^{k-1} y_1$, with $y_1 = (\gamma - 1)/(\gamma^K - 1)$ and $\gamma > 1$
- if $\gamma = 1$ then a equal distribution (like the *split*)
  if $\gamma = 2$ then a exponential distribution
- we add redundancy to the global flow, $r \in [0, 1]$; $r = 0$ means no redundancy, $r = 1$ means that any frame is sent twice (as in the *redundant split* method)
- proportional redundancy distribution with the weights y_k
- each frame is sent either once or twice (no more than twice)

Discrimination per frame type

We have γ's and r's for each frame type.

Quality Analysis in Multi-Source Streaming

Quality Evaluation

- for each frame type, the total loss rate at configuration $\vec{n}$ is $LR_{\vec{n}} = \sum_{j:n_j=0} LR_j = 1 - \left(\sum_{i:n_i=1} y_i\right)\left(1 + r\sum_{j:n_j=0} \frac{y_j}{1-y_j}\right)$.
- Let $f()$ be the PSQA **complex function**: $Q = f\left(LR_{I,\vec{n}}, LR_{P,\vec{n}}, LR_{B,\vec{n}}\right)$
- when at least one server is down, the average quality is $\mathbf{E}(Q_K) = \sum_{\vec{n}\neq\vec{1}} Pr(\vec{N}(T) = \vec{n})Q(LR_{I,\vec{n}}, LR_{P,\vec{n}}, LR_{B,\vec{n}})$.

Parameter Effect

we can now analyze the effect of the parameters (K, γ's, and r's) on the quality perceived at the client node. But, how configure them?

Quality Analysis in Multi-Source Streaming

Quality Evaluation

- for each frame type, the total loss rate at configuration $\vec{n}$ is $LR_{\vec{n}} = \sum_{j:n_j=0} LR_j = 1 - \left(\sum_{i:n_i=1} y_i\right)\left(1 + r\sum_{j:n_j=0} \frac{y_j}{1-y_j}\right)$.
- Let $f()$ be the PSQA **complex function**: $Q = f\left(LR_{I,\vec{n}}, LR_{P,\vec{n}}, LR_{B,\vec{n}}\right)$
- when at least one server is down, the average quality is $\mathbf{E}(Q_K) = \sum_{\vec{n}\neq\vec{1}} Pr(\vec{N}(T) = \vec{n})Q(LR_{I,\vec{n}}, LR_{P,\vec{n}}, LR_{B,\vec{n}})$.

Parameter Effect

we can now analyze the effect of the parameters (K, γ's, and r's) on the quality perceived at the client node. But, how configure them?

Optimal Quality Analysis in Multi-Source Streaming

Optimization Problem

- the average quality is our final target
- we have seven parameters to optimize: K, γ_I, γ_P, γ_B, r_I, r_P and r_B
- two scenarios:
  - No upload bandwidth limitations: Total Bandwidth: $BW^{red} = (1 + r)B$ Kbps
  - Equal upload bandwidth limitation: Indivdual Bandwidth: BW^{red}/K Kbps
- the optimization results were obtained using the *fminsearch* function of Matlab[a]

[a] © 1984-2007 The MathWorks, Inc.

Multi–source Streaming Remarks

Summary of Results

1. With a minimum number of 7 *server peers*, $\mathbf{E}(Q_K)$ is ≥ 9.0, which means an excellent quality.
2. We can see that, for P-frames, the largest quality is achieved when γ_P is around 2. Server peers that stay longer into the system will be responsible for delivering the most important information.
3. After the P-frame in importance order with respect to the quality, we have I-frames and then B-frames
4. The r values are counterbalanced by the γ values. If the largest part of the information is delivered by the most stable peers, it is not necessary to use a high redundancy factor.

Outline

Some concluding remarks

PSQA has been validated

- PSQA works accurately in several contexts:
  - for audio and for video
  - for one-way and for interactive communications
- PSQA couples well with standard analytical models
  - both for performance evaluation
  - and for dependability analysis
  - typically models are queuing models, Markov chains, ...
- PSQA's capacity of real-time work can be exploited
  - we developed a first prototype of AUDIT monitoring system
  - we explored in deep QoE-based network control based on PSQA for P2P networks
- Many papers (and 3 PhD) already published
  - new Web site under construction
  - papers can be sent under request

Some perspectives

Several research directions

- Improve quality assessment in accuracy.
  - by combining on some signal-level analysis
  - by improving the mathematical framework
- Improve our GOL!P2P prototype:
  - Enhanced synchronism mechanism for our multi-source streaming technique in order to diminish the connection delay
  - Analysis extension in the buffering strategy
  - Deeper study of the tree-based overlay
  - Add MyTV and VoD services (and our search caching technique)
  - Security, Access Control,. . .
- Try it in a real environment: AdinetTV?
- Exploring a mesh-based overlay $\Longrightarrow$ GOALBIT project

Goalbit Homepage

http://goalbit.sourceforge.net

fig-portal-goalbit3

Some offers

Some opportunities

- PhD and/or post-doc work on different methodological aspects of the project ...
  - mathematical bases
  - automatic measuring of QoE
  - QoE-based network control
- ... or in specific application areas
  - on P2P systems for video
  - on MobileTV systems
  - on future video codecs
- and industrial transfer perspectives
  - we look for partnerships for developing measuring / monitoring systems
  - this transfer should probably take different forms depending on the technological ares (type of network, and/or service ...

Questions?

Thank you!

For your attention.

Outline

AdinetTV Homepage

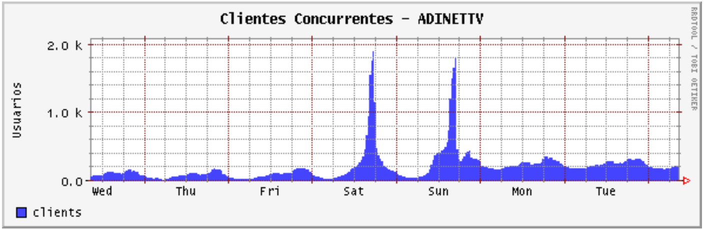
http://adinettv.com.uy

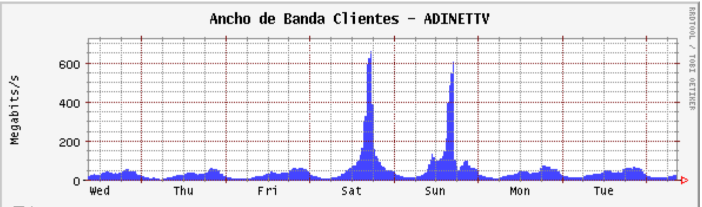
Video Delivery Reference Service

- Our video delivery reference service: www.adinetTV.com.uy
- We want to improve adinetTV: in scalability, and in quality
- adinetTV is a Content Delivery Network (CDN), we want to extened it with a P2P system

Video Delivery Reference Service (2)

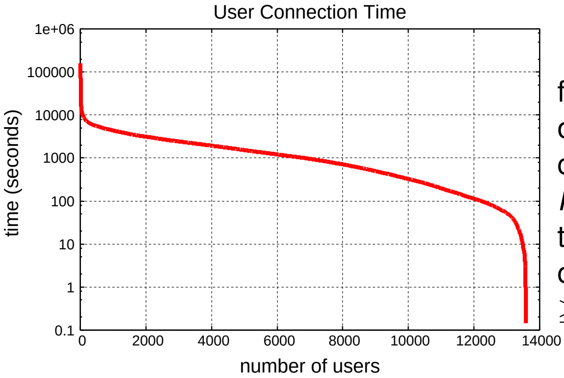
- More than 50K users from around the world
- with 4K simultaneous users (peak)
- football (soccer) fans: the popularity truth...





Video Delivery Reference Service (3)

we know the clients' dynamics:



for a given number of users x, the curve gives $F(x) = y$ meaning that x users have connection time $\geq y$.

Outline

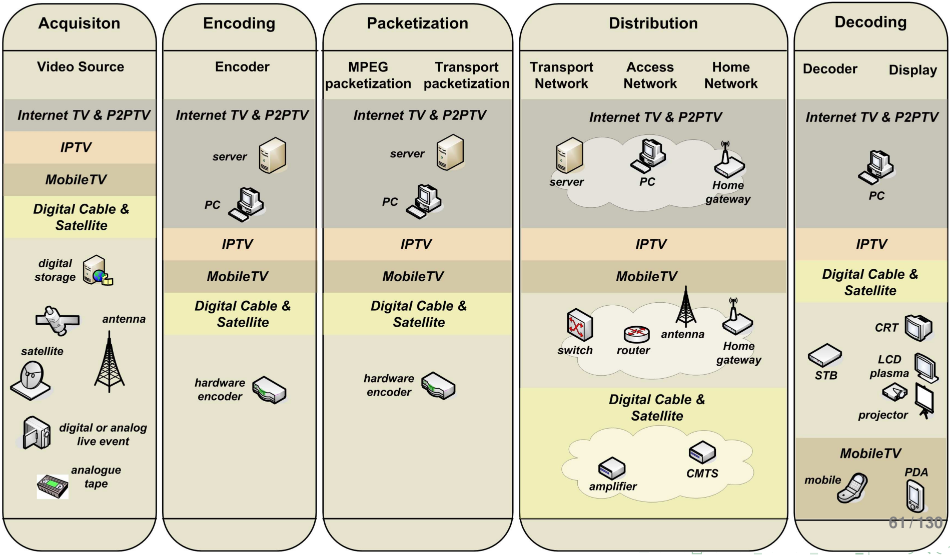
Video Delivery Networks

Definition

A VDN is a system that delivers video transparently to end users.

- Services:
  - Video on Demand
  - live TV
  - ...
- Architectures:
  - **Internet Television**: msnTV, YouTube, Jumptv, myTVPal, etc.
  - **P2PTV**: iBBC, Joost, PPlive, TVUnetwork, PPstream, SopCast, TVAnts, etc.
  - **IPTV**: Alcatel, Siemens, Cisco, etc.
  - **MobileTV**: IMS architectures.
  - **Digital Cable and Satellite**: DirectTV, etc.

VDN Components influence in the QoE



Video Delivery Networks

VDN Protocols influence in the QoE

Acquisiton

Video Source

Internet TV & P2PTV

IPTV

MobileTV

Digital Cable & Satellite

no relevant standards or protocols

Encoding

Encoder

Internet TV & P2PTV

MPEG-2
MPEG-4 ASP (divx, xvid,..)
MPEG-4 AVC (H.264)
WMV9 (SMPTE VC-1)
RealVideo,
Theora
NSV
etc...

IPTV

Digital Cable & Satellite

MPEG-2
MPEG-4 AVC (H.264)
SMPTE VC-1 (WM9)

MobileTV

3GPP video codecs:
ITU-T H.263 video,
MPEG-4 simple visual profile video

Packetization

MPEG packetization	Transport packetization
Internet TV & P2PTV	
MPEG-TS MPEG-PS MOV rm AVI/ASF OGG FLV	RTP UDP TCP HTTP MMS OGG flash bittorrent etc...
IPTV	
MPEG-TS (SPTS)	RTP UDP
AVI/ASF	MMS
MobileTV	
MPEG-TS (SPTS)	RTP
Digital Cable & Satellite	
MPEG-TS (MPTS)	

Distribution

Transport Network	Access Network	Home Network
Internet TV & P2PTV		
Public Internet IP Overlay P2P Public Internet IP	Ethernet ADSL2+ VDSL WiMAX	*copper/coax/ power* Ethernet Wi-Fi
IPTV		
MPLS/IP ATM	*copper/fiber* Ethernet ADSL2+ VDSL WiMAX	Ethernet Wi-Fi
MobileTV		
MPLS/IP	W-CDMA CDMA2000 GSM, GPRS DVB-H, DMB, TDtv, ISDB-T	
Digital Cable & Satellite		
QAM PSQK	*coax/fiber* DVB-C DVB-S	*coax*

Decoding

Decoder **Display**

Internet TV & P2PTV

IPTV

MobileTV

Digital Cable & Satellite

no relevant standards or protocols

Outline

Content Networks

Definition

A Content Network is an overlay IP network that supports content routing, that is, messages are routed on the basis of their contents rather than the IP address of their destination (Kung & Wu 2002, "Content networks: taxonomy and new approaches", Oxford University Press).

- Some characteristics:
  - virtual networks over the Internet or over a corporative network,
  - no fixed links between contents and hosts storing them,
  - content re-allocations, replications, and deletions.

Examples

- Examples:
  peer-to-peer networks, collaborative networks, content delivery networks, cooperative Web caching, subscribe-publish networks, backup networks, instant messaging, multiplayer games, domain name system, etc.
- Motivations:
  - to share resources,
  - to take into account different application requirements,
  - to gracefully scale with the number of users,
  - to avoid centralized, single failure points or bottlenecks.

Examples (2)

Characteristics

- Addressing and routing based on content description, not on explicit location.
- Content network = knowledge network, where knowledge is information about content locations.
- Network objective: to discover content locations in an effective and efficient way.
- Design and deployment of content networks have motivated empirical studies and the development of analytical models to better understand and predict network behavior.

Taxonomy

Architecture:

- Decentralization
- Content aggregation
- Content placement
- Ad-Hoc Connectivity, Self-Organization
- Autonomy
- Heterogeneity
- Interoperability

Behavior:

- Anonymity
- Performance
- Scalability
- Transparency and Usability
- Security
- Fault Resilience

Read more: ”Redes de Contenido: Taxonomía y Modelos de evaluación y diseño de los mecanismos de descubrimiento de contenido.“
http://www.fing.edu.uy/inco/pedeciba/bibliote/tesis/tesis-

Content Networks for video

Two main architectures:

- Content Delivery Network (CDN):
  - Pros: the control side; Cons: the resource sharing side
  - **Internet Television**: msnTV, youTube, Jumptv, myTVPal, Akamai (Apple trailers), Kontiki, vitalStream, etc.
  - **IPTV**: Alcatel, Siemens, Cisco, etc.
- Peer to Peer (P2P):
  - Pros: the resource sharing side; Cons: the control side
  - **P2PTV**: iBBC, Joost, PPlive, TVUnetwork, PPstream, SopCast, TVAnts, etc.
- Hybrid (between CDN and P2P):
  - Pros: distributed resources, centralized control; Cons: ???
  - **our GOL!P2P project**
  - Coopnet, and others...

Peer-to-Peer

Definition

A P2P is a Content Network that takes advantage of the free resources available in the Internet edges (i.e. the final users).

- Some characteristics:
  - peers connect and disconnect very frequently, in an autonomous and completely asynchronous way
  - the resources in the network grow with the popularity (scalability)
- Design goal of a P2P solution:
  to offer the quality needed by the clients in a highly varying environment

P2P classification based on its application

- **Content Sharing**
  - File delivery
  - Multimedia Distribution
  - Distributed Storage
  - Caching, Edge Services
  - Information Mgmt: discover, aggregate, filter, organize,..
- **Collaboration**
  - Communication: chat, messaging
  - Co-review/edit/author/create
  - Gaming
  - Discovery
- **Distributed Computing**
  - Internet/Intranet Distributed Computing
  - Grid Computing

P2P Network Communication

- Two types of exchanged data:
  - content (files, videos,...)
  - control and routing (publications, searchs, connections/disconnections,...)
- Three methods to exchange data:
  - client/server
  - hierarchically
  - completely distributed
- if both (content and control) are distributed then the network is called pure, otherwise the network is call hybrid
- *usually pure networks do not scale well*

Overlay Network: the control or routing layer

Definition

The Overlay Network is a directed graph. The nodes are the peers. If a participating peer knows the location of another peer, then there is a directed edge from the former node to the latter.

P2P classification based on how the overlay is constructed:

- unstructured
- structured

Overlay Network: the control or routing layer (2)

P2P classification based on how the overlay is constructed:

- **unstructured**:
  - links are established arbitrarily (flooding)
  - undeterministic and unefficient search
  - high amount of signalling traffic, poor scalability
  - anonymity
  - Examples: Gnutella and FastTrack
- **structured**:
  - they use globally consistent protocols to route a search of nodes or content
  - they usually are distributed hash tables (DHT)
  - deterministic and efficient search
  - degradation with the peers dynamics
  - Examples: Kademlia, Chord, CAN

Content Delivery: the data download

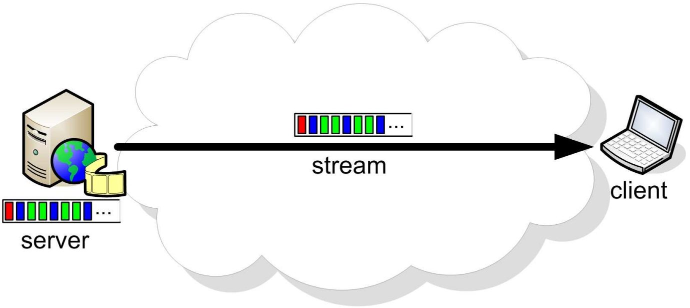
- for file sharing (and Video on Demand):
  - Bittorrent-like protocols
- for live video:
  - single source (streaming)
  - multi–source (Multiple Description Coding, Network Coding,...)

Outline

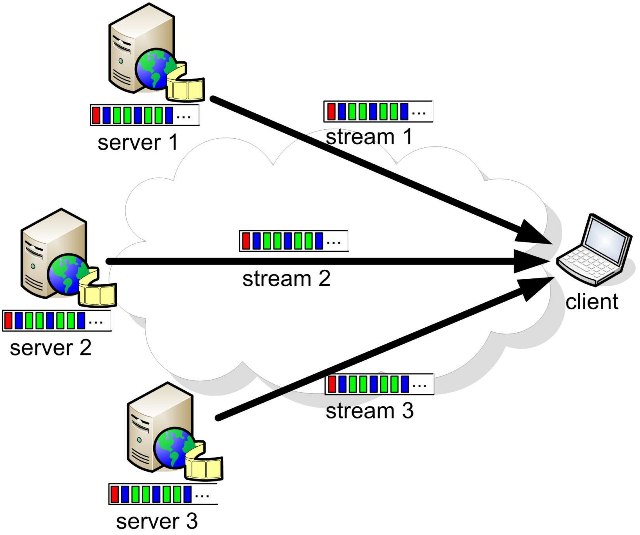
Multi-source approach: the client viewpoint

- In a P2P system, most nodes (peers) are clients and also servers.
- We consider the following streaming policies (seen from the client's point of view):
  - single: each peer/client receives the complete stream from another peer/server;
  - copy: each client receives K copies of the stream from K different servers;
  - simple split: each client receives K different (and disjoint) substreams from K different servers and reconstructs the whole stream from them;
  - $K-1$ redundant split: each client receives K different substreams from K different servers, and is able to reconstruct the whole stream from any $K-1$ among them.

“Single” streaming policy

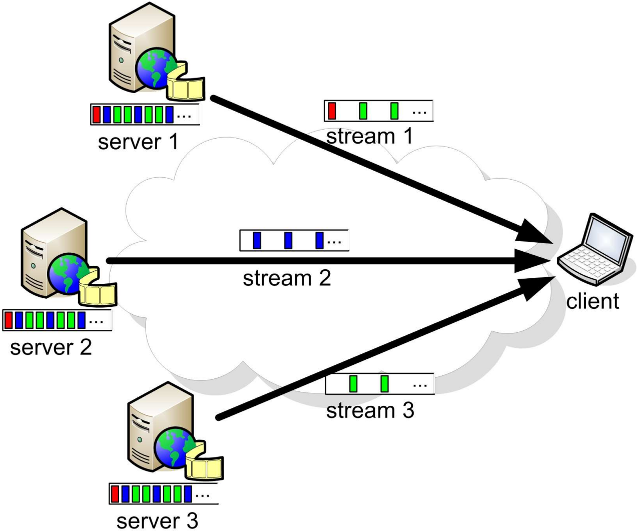


“Copy” streaming policy

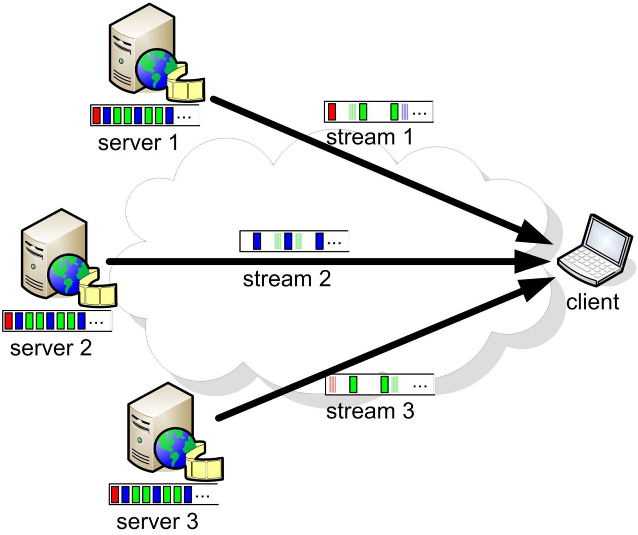


Multi-Source Streaming Technique

“Simple split” streaming policy



“$K-1$ Redundant split” streaming policy

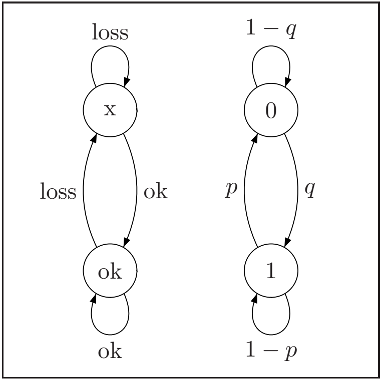


Multi-Source Streaming Technique

The multi-source streaming technique is useful?

Server failure and "single" streaming

- we will look at the consequences of the type of sending scheme (single, copy, split...) on *LR* and *MLBS*
- we model the failure process on a "single" server streaming with the so-called simplified Gilbert model



- a 2-state Markov chain
- p is the probability of a a loss after a correct transmission, and q is the probability of a correct transmission after a loss
- after some algebra, for the "single" case, we get: $LR = \frac{p}{p+q}$, $MLBS = \frac{1}{q}$

Cases of “copy” and “split”

- we only consider the failures of the servers (because, by far, they have much more impact on losses than congestion, in a P2P network)
- the loss process at any server is represented by the same simplified Gilbert model, with parameters p and q
- “copy” method:
  - assume K copies of the same stream travel following independent and stochastically equivalent paths to the same terminal:
    $LR_K^{copy} = \left(\frac{p}{p+q}\right)^K = LR^K$,
    $MLBS_K^{copy} = \frac{1}{1-(1-q)^K} = \frac{1}{1-(1-MLBS^{-1})^K}$
- “split” method:
  - consider the following simple version: server 1 sends frames 1, $K+1$, $2K+1$, ..., server 2 sends frames 2, $K+2$, etc.:
    $LR_K^{split} = \frac{p}{p+q} = LR$, $MLBS_K^{split} = 1 + \frac{p}{q} = \frac{1}{1-LR}$

Quality as a function of the streaming policies

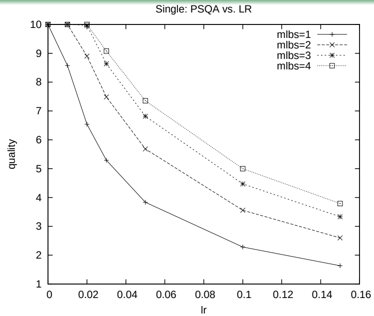
- we can now compare the effect of the different streaming policies on the quality perceived at the client side.
- for this, we use the same frame loss process at the server side (same parameters p and q).
- let $f()$ be the PSQA **simple function**, it is possible to compute:

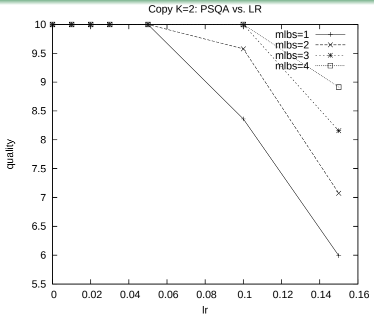
$$Q = f\left(LR, MLBS\right)$$

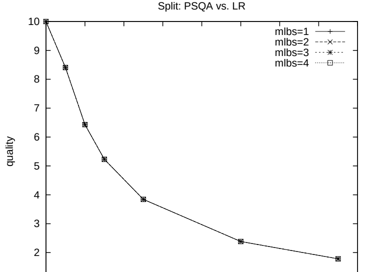
$$Q_K^{copy} = f\left(LR_K^{copy}(LR), MLBS_K^{copy}(MLBS)\right)$$

$$Q_K^{split} = f\left(LR_K^{split}(LR), MLBS_K^{split}(LR)\right)$$

Effect of the streaming policies on the QoE



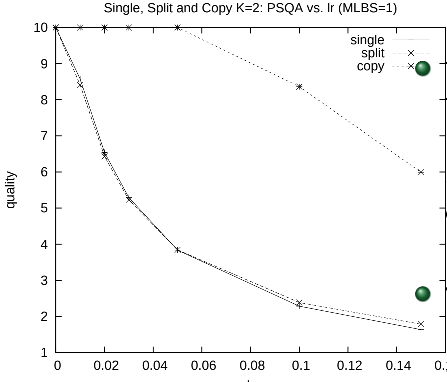




- in the three cases, the perceived quality deteriorates quickly with increasing values of *LR*
- in the copy policy, the effect of *MLBS* is significant: we get better quality with larger *MLBS*

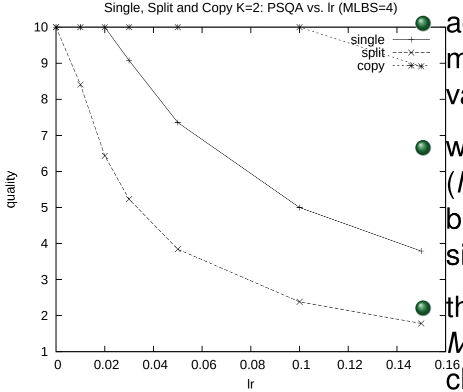
Comparing the QoE in our streaming policies

- general result:
  - in the three cases, the perceived quality deteriorates quickly with increasing values of *LR*
  - in the copy policy, the effect of *MLBS* is significant: we get better quality with larger *MLBS* values.



- the copy policy, which transmits every frame twice, profits from this redundancy and copes much better with quite high values of *LR*, keeping good quality results
- with low burst sizes ($MLBS = 1$), the single and split policy give the same

Comparing the QoE, when increasing the *MLBS*



- again copy policy copes much better with quite high values of *LR*
- with high burst sizes ($MLBS = 4$), the split policy behaves worse than the single policy
- this result holds if $MLBS \leq 10$ (recall the characteristics of the PSQA experiments...)

Concluding remarks

- PSQA allows to address the “ultimate target”, the QoE, in a quantitative way
- multi-source schemes allow to improve the QoE in a P2P architecture, by providing some protection against network and server errors
- the previously described methodology can be easily improved:
  - taking into account more parameters (more than just *LR* and *MLBS*)
  - using more accurate models of the peers dynamics (on-off behavior, multi-class models...)

In a very dynamic peers context:

Can we ensure some video quality level
using the multi-source streaming technique?

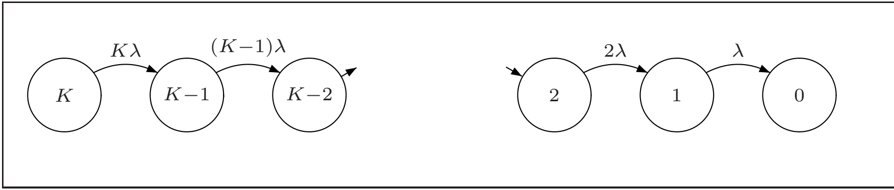
Taking into account the peers' dynamics

- assume each peer remains connected for an exponential distributed amount of time, independently of other peers' behaviors, with parameter λ (the same for all the servers)
- the network reconfigurates every T units of time (structured P2P with central control)
- after a reconfiguration, we assume that every client receives the stream from K servers (using "copy", "split" or "redundant split")
- after a reconfiguration and until next one, servers can fail (leave), randomly
- numerical values computed from logs of our video delivery reference service:

$$1/\lambda = 900 \text{ sec}; \; T = 10 \text{ sec}$$

Evolution of the number of active servers

With the previous assumptions, the evolution of the number of servers for a fixed client, assuming “last configuration at time 0” and “next one is actually removed and never happens”, is the following simple (pure-death) absorbing Markov chain:



The number of active servers at t is Binomial; if $p_{K,i}(t)$ is the probability of having i servers active at time t, we have

$$p_{K,i}(t) = \binom{K}{i} e^{-i\lambda t}(1 - e^{-\lambda t})^{K-i}, \quad K \geq i \geq 0, \quad \lambda > 0, t \geq 0.$$

LR and *MLBS* characterization for the "copy" streaming policy

- From the client viewpoint, network state at an arbitrary time is given by the pair (K, i) (total number of substreams, number of connected servers).
- Let us look at the *LR* and *MLBS* values for each policy when the network is at state (K, i).
- In the case of the "copy" policy, the Loss Rate with K servers among which i are connected (alive) is

$$LR^{copy}_{K,i} = \begin{cases} 1 & \text{if } i = 0 \\ 0 & \text{otherwise} \end{cases}, \quad K \geq i \geq 0, \; K \geq 1.$$

- The Mean Loss Burst Size is

$$MLBS^{copy}_{K,i} = \begin{cases} \infty & \text{if } i = 0 \\ \nexists & \text{otherwise} \end{cases}, \quad K \geq i \geq 0, \; K \geq 1.$$

"split" streaming policy

- The Loss Rate with K servers in the proposed split mode, with i connected, is

$$LR_{K,i}^{split} = \frac{K-i}{K}, \quad K \geq i \geq 0,\ K \geq 1.$$

- The Mean Loss Burst Size is much more involved to compute than in the previous case. Since our goal is to guarantee some quality level, we use (trivial) lower and upper bounds, observing that, by definition,

$$1 \leq MLBS_{K,i}^{split} \leq K - i, \quad K \geq i \geq 0,\ K \geq 1.$$

"$K-1$ Redundant split" streaming policy

- The Loss Rate with K servers, among which i are still connected, is

$$LR^{red}_{K,i} = \frac{K-i}{K} - \frac{(K-i)}{K}\frac{i}{K-1} = \frac{(K-i)(K-1-i)}{K(K-1)}.$$

- For the evaluation of the Mean Loss Burst Size we use the same kind of trivial lower and upper bounds as in the "split" case:

$$1 \leq MLBS^{red}_{K,i} \leq K-1, \quad K \geq i \geq 0,\; K \geq 1.$$

Quality as a function of the streaming policies

- We can now compare the effect of the different streaming policies on the quality perceived at the client side.
- Let $f()$ be the PSQA **simple function**:

$$Q = f\,(LR, MLBS)$$

- We can compute the PSQA measure associated with every state (K, i) of the network:

$$f\,(LR(K, i), MLBS(K, i))$$

  for every (K, i) and for the three policies, together with the probabilities $p_{K,i}(t)$ of observing state (K, i)

Quality as a function of the streaming policies (2)

- We do not include the (long) numerical tables corresponding to our case studies.
- Some observations for this particular scenario:
  - (i) in the simple split policy (with no redundancy), when the number of servers grows, the subjective quality degrades very quickly;
  - (ii) in the redundant split policy, passing from one server to two servers improves greatly the quality levels; adding additional servers can lead to slight decreases in perceived quality, but the behavior is globally very robust;
  - (iii) the copy policy has always the best perceived quality levels, but at a high price because of the transmission overhead.
- These values can be used afterwards to support more sophisticated decision making.

Average quality as a function of the number of servers

- The mean quality (considering the whole client population) is, with our assumptions, given by:

$$\mathrm{E}(Q_K) = \sum_{i=1}^{K} Q_{K,i}(LR_{K,i}, MLBS_{K,i}) p_{K,i}(T).$$

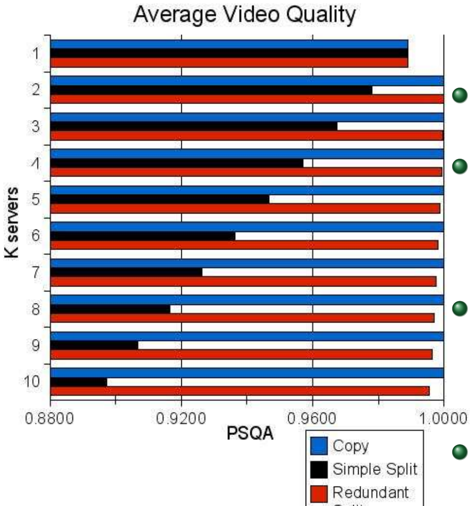
- We can now compare the average video quality for the three policies[1]

[1] The data was computed using the lower bound for the perceived quality, the difference with the upper bound is very small and can be neglected.

Average quality as a function of the number of servers (2)

K/Method (bandwidth)	"Copy" (KB)	"Split" (B)	"Redundant split" ($2B$)
1	0.9890	0.9890	0.9890
2	0.9999	0.9781	0.9999
3	1.0000	0.9675	0.9996
4	1.0000	0.9570	0.9993
5	1.0000	0.9466	0.9989
6	1.0000	0.9364	0.9983
7	1.0000	0.9264	0.9977
8	1.0000	0.9166	0.9970
9	1.0000	0.9068	0.9963
10	1.0000	0.8973	0.9955

Average quality as a function of the number of servers (3)



- we used $1/\lambda$ = 900 sec
- and $T = 10$ sec
- for the mean quality, we used the trivial lower bound of the *MLBS*...
- but the values are almost the same with the upper

Determining the number of servers to ensure a given quality

- How many servers are needed in each method to ensure that with a given probability (or confidence level), the quality of the transmission will be greater or equal than a pre-defined quality level?
- It is possible to see for example that if we want to ensure a perfect quality transmission with a given probability of 0.999 (an alternative equivalent interpretation is that we want to ensure that at least 99.9% of the users will perceive perfect quality), we need to chose a method and a K such that:

$$\Pr(Q_K > Q_{min}) = \sum_{i=1 / Q_{K,i} > Q_{min}}^{K} p_{K,i}(T) \geq 0.999.$$

Determining the number of servers to ensure a given quality (2)

K/Method (bandwidth)	"Copy" (KB)	"Split" (B)	"$K-1$ Redundant split" ($2B$)
1	0.988950	0.988950	0.988950
2	0.999878	0.978023	0.999878
3	0.999999	0.967216	0.999636
4	~1.000000	0.956529	0.999278
5	~1.000000	0.945959	0.998806
6	~1.000000	0.935507	0.998222
7	~1.000000	0.925170	0.997529
8	~1.000000	0.914947	0.996729
9	~1.000000	0.904837	0.995826
10	~1.000000	0.894839	0.994820

Concluding remarks

- Multi-source schemes allow to statistically ensure the QoE in a P2P architecture, by providing some protection against server failures.
- Simple analytical models can be used to understand the qualitative and quantitative behavior of the different policies.
- Other objective functions are of interest, all based on PSQA (for instance, we can look at something like "mean number of clients receiving at least some minimal quality value", etc.)
- The general methodology can be easily improved:
  - taking into account more parameters (more than just *LR* and *MLBS*),
  - including other aspects as costs or bandwidth limitations at the networks components.
- We have implemented a general scheme where the

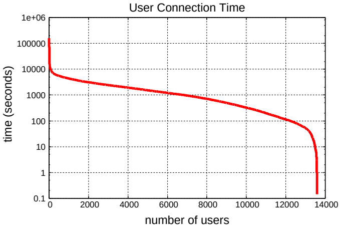
How can we take advantage of the peers' heterogeneity?

The main ideas

- a *client peer* receives the video stream from K heterogeneous and independent *server peers*
- Idea:
  - a peers' classification, based on the their connection-time in the network (peer's lifetime)
  - together with an improved multi-source streaming technique that send the most important data from the most reliable peers.

Classifying the clients of our Video Delivery Service

- the connection time of server k is exponentially distributed, with parameter λ_k, $k = 1, \cdots, K$



- assume the order $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_K$ (the best at the end)
- we group clients in clusters with the same number of clients each

- $N_k(t)$ is the binary r.v. equal to 1 iff server k is connected at t, and $\vec{N}(t)$ is the vector $\vec{N}(t) = (N_1(t), \cdots, N_K(t))$
- therefore, the probability of each *configuration* $\vec{n} \in \{0, 1\}^K$ is

Classifing the clients of our Video Delivery Service (2)

- we consider all values of K from 1 to 10

Table: Values of λ in the case where the peers having the content are divided into different classes, based on their connection-time.

λ_i/K	1	2	3	4	5
1	2.316e-03	4.170e-03	5.636e-03	6.806e-03	7.750e-03
2		4.617e-04	9.787e-04	1.535e-03	2.138e-03
3			3.336e-04	6.420e-04	9.333e-04
4				2.813e-04	5.088e-04
5					2.512e-04

Classifying the clients of our Video Delivery Service (3)

Table: The values for the parameter λ for the case where the peers that have the content are divided in different classes, based on their connection-time (cont).

λ_i / K	6	7	8	9	10
1	8.527e-03	9.167e-03	9.692e-03	1.014e-02	1.053e-02
2	2.747e-03	3.342e-03	3.919e-03	4.461e-03	4.967e-03
3	1.239e-03	1.579e-03	1.938e-03	2.307e-03	2.671e-03
4	7.184e-04	9.220e-04	1.132e-03	1.362e-03	1.607e-03
5	4.364e-04	5.994e-04	7.618e-04	9.173e-04	1.078e-03
6	2.315e-04	3.902e-04	5.223e-04	6.572e-04	7.895e-04
7		2.170e-04	3.578e-04	4.718e-04	5.841e-04
8			2.048e-04	3.337e-04	4.343e-04
9				1.963e-04	[illegible]

Probability that at least one server left the network

Table: Probability that at least one peer has left the network at time T

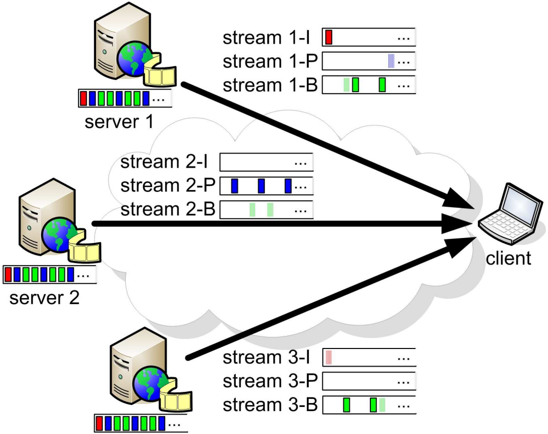
K	$\sum_{\vec{n} \neq \vec{1}} Pr(\vec{N}(T) = \vec{n})$
1	2.289e-02
2	4.526e-02
3	6.713e-02
4	8.848e-02
5	1.093e-01
6	1.297e-01
7	1.497e-01
8	1.691e-01
9	1.882e-01
10	2.068e-01

Improving the Multi-Source Technique

- server k sends a fraction y_k of the stream
- since we want that the best servers send most of the data, we set $y_k = \gamma^{k-1} y_1$, with $y_1 = (\gamma - 1)/(\gamma^K - 1)$ and $\gamma > 1$
- if $\gamma = 1$ then a equal distribution (like the “simple split”)
  if $\gamma = 2$ then a exponential distribution
- we also add some redundancy $r \in [0, 1]$ to the global flow; $r = 0$ means no redundancy, $r = 1$ means that any frame is sent twice (as in the “$K - 1$ redundant split” method)
- the redundancy is distributed proportionally to the weights y_k
- the technique implies that:
  - each frame is sent either once or twice, but no frame is sent more than twice
  - we have three parameters to optimize: K, γ, and r

Improving the “Redundant Split” streaming policy

- besides of the weights used, we extend the technique using different streams for each frame type
- actually the parameters are: K, γ_I, γ_P, γ_B, r_I, r_P and r_B



And again, the same type of analysis...

- for each frame type, the total loss rate at configuration $\vec{n}$ is

$$LR_{\vec{n}}^{red} = \sum_{j:n_j=0} LR_j^{red} = 1 - \Big(\sum_{i:n_i=1} y_i\Big)\Big(1 + r\sum_{j:n_j=0}\frac{y_j}{1-y_j}\Big).$$

- Let $f()$ be the PSQA **complex function**:

$$Q = f\left(LR_{I,\vec{n}}, LR_{P,\vec{n}}, LR_{B,\vec{n}}\right)$$

- the average quality (taking into a account servers failures), when at least one server is down, is

$$\mathbf{E}(Q_K) = \sum_{\vec{n}\neq\vec{1}} Pr(\vec{N}(T) = \vec{n})Q(LR_{I,\vec{n}}, LR_{P,\vec{n}}, LR_{B,\vec{n}}),$$

- we can now analyze the effect of the parameters (K, γ's, and r's) on the quality perceived at the client node

Best "Redundant Split" streaming policy

- the average quality is our final target
- we optimize the parameters (K, γ's, and r's) in two scenarios:
  - limitation on the **Total Bandwidth**:

$$BW^{red} = (1 + r)B \text{ Kbps}$$

  - limitation on the **Individual Bandwidth**:

$$BW^{red}/K \text{ Kbps}$$

- the optimization results were obtained using the *fminsearch* function of Matlab[2]

[2] © 1984-2007 The MathWorks, Inc.

Limitation on the Individual Bandwidth, with $r = 25\%$

Table: γ and redundancy factors Optimization (25% Redundancy)

# Servers	$\mathbf{E}(Q_K)$	γ_I	γ_P	γ_B
1	0	-	-	-
2	3.557	1.000e+00	1.960e+00	1.607e+00
3	5.442	1.296e+00	1.999e+00	1.771e+00
4	7.020	1.108e+00	1.999e+00	1.999e+00
5	7.972	1.220e+00	1.999e+00	1.797e+00
6	8.719	1.999e+00	1.997e+00	1.999e+00
7	9.064	1.409e+00	1.999e+00	1.661e+00
8	9.319	1.999e+00	1.999e+00	1.692e+00
9	9.510	1.995e+00	1.932e+00	1.974e+00
10	9.586	1.999e+00	1.996e+00	1.949e+00

Limitation on the Individual Bandwidth, with $r = 25\%$ (2)

Table: γ and redundancy factors Optimization (25% Redundancy) (cont)

# Servers	$\mathbf{E}(Q_K)$	r_I	r_P	r_B
1	0	-	-	-
2	3.557	9.913e-01	2.953e-01	5.410e-04
3	5.442	9.999e-01	1.705e-01	4.744e-01
4	7.020	9.999e-01	2.279e-01	2.610e-01
5	7.972	9.999e-01	1.988e-01	3.856e-01
6	8.719	8.286e-01	3.305e-01	9.977e-06
7	9.064	9.999e-01	2.462e-01	2.373e-01
8	9.319	8.124e-01	2.946e-01	6.276e-06
9	9.510	7.986e-01	3.428e-01	5.984e-05
10	9.586	1.550e-01	3.959e-01	4.255e-02

Concluding remarks

The following conclusions can be listed.

- (i) With a minimum number of 7 *server peers*, $\mathbf{E}(Q_K)$ is ≥ 9.0, which means an excellent quality.
- (ii) As expected, more bandwidth results in a better quality for the delivered video.
- (iii) We can see that, for P-frames, the largest quality is achieved when γ_P is around 2. This means that *server peers* that stay longer into the system will be responsible for delivering the most important information.
- (iv) After the P-frame in importance order with respect to the quality, we have I-frames and then B-frames

Concluding remarks (2)

- (v) The maximum $\mathbf{E}(Q_K)$ improves when the number of *server peers* increases. Nevertheless, increasing the number of *server peers* has practical problems.
- (vi) The redundancy value is counterbalanced by the γ value: for greater γ we can set a smaller value to the redundancy factor, with the following interpretation: *if the largest part of the information is delivered by the most stable peers, it is not necessary to use a high redundancy factor.*

Outline

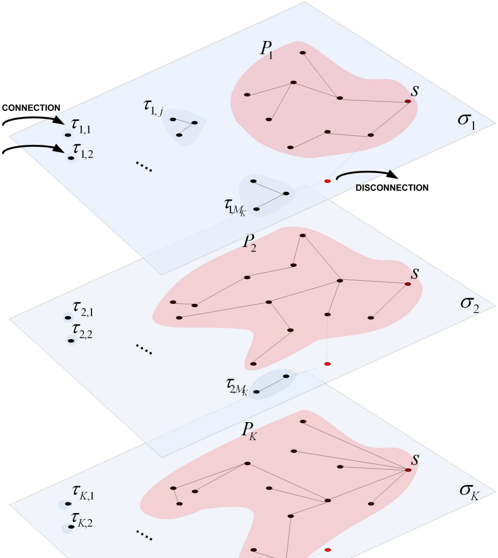
P2P network model: streaming

- In a P2P system, nodes (peers) are clients and also servers.
- Distribution scheme: one source *s* for the original stream (broadcaster node).
- Stream is decomposed in K different substreams $\sigma_1, \sigma_2, \ldots, \sigma_K$ encoded with constant bandwidth bw_k.
- A peer (acting as client) receives $\sigma_1, \sigma_2, \ldots, \sigma_K$ from K different peers (acting as servers) and reconstructs original stream.
- Quality of the reconstructed stream depends on which substreams arrive (streaming scheme may allow for some redundancy).
- The same peer may be server to other peers, sending one or more σ_k.

P2P network model: system dynamics

- To decide which client will serve another one, some degree of intelligence and knowledge about the peers and the network state is needed
- Different proposals are based on decentralized or centralized algorithms, with structured or unstructured delivery, etc.
- Network reconfigured at discrete points in time, every Δt (taken for simplicity as a time unit). We denote the tth interval $(t, t+1]$.
- Evolution from t to $t+1$:
    - some nodes leave, possibly disconnecting other clients in some substreams.
    - Some nodes enter the network requesting for connection, and remain isolated until $t+1$.
    - At $t+1$, the network is reconfigured, to reconnect disconnected nodes and to connect new arrivals.

Graphical model description



Model formalization for the reconnection procedure

- Consider the following binary variables, for each $k \in \{1, 2, \cdots, K\}$ and $i, j \in \mathcal{N}(t)$ the set of connected nodes:

$$x_{i,j}^k = 1 \text{ iff node } i \text{ sends } \sigma_k \text{ to node } j \text{ at time } t^+.$$

$$y_{i,j}^k = 1 \text{ iff } i \text{ precedes } j \text{ in plane } k \text{ at time } t^+.$$

$$z_i = 1 \text{ iff } i \text{ remains connected until } t+1.$$

$$\tilde{x}_{i,j}^k = 1 \text{ iff } i \text{ sends } \sigma_k \text{ to } j \text{ at time } t+1.$$

$$\tilde{y}_{i,j}^k = 1 \text{ iff } i \text{ precedes } j \text{ in plane } k \text{ at time } t+1.$$

- Quality at time t experienced by node $i \in \mathcal{N}(t)$ (as measured using PSQA) is $Q_i = f(y_{s,i}^1, y_{s,i}^2, .., y_{s,i}^K)$, where s is the video stream source node.
- Expected quality for node i at time $t+1$, $\mathrm{E}\{\tilde{Q}_i\}$, where

Mathematical programming model

$$\max \mathrm{E}\left\{\frac{\sum_{i=1}^{N}\tilde{Q}_i}{\sum_{i=1}^{N} z_i}\right\}$$ // global expected perceived quality

st:

$$y_{i,j}^k + y_{j,i}^k \leq 1, \forall i,j \in N, \forall k \in K,$$ // loops are not allowed

$$y_{i,j}^k = x_{i,j}^k + \sum_{l=1}^{N} y_{i,l}^k x_{l,j}^k, \forall i \in N, \forall j \in N - \{s\}, \forall k \in K,$$

// precedence constraints

$$\sum_{i=1}^{N} x_{i,j}^k \leq 1, \forall j \in N, \forall k \in K,$$ // each substream arrives from a single no

$$\sum_{k=1}^{K}\{bw^k \sum_{i=1}^{N} x_{i,j}^k\} \leq BW^{out}(i), \forall i \in N,$$ // bandwidth capacity constraints

Mathematical programming model (2)

$$\tilde{x}_{i,j}^{k} = z_i x_{i,j}^{k} z_j, \forall i \in N, \forall j \in N, \forall k \in K,$$

// a link is preserved if source and terminal do not leave

$$\tilde{y}_{i,j}^{k} = \tilde{x}_{i,j}^{k} + \sum_{l=1}^{N} z_i \tilde{y}_{i,l}^{k} \tilde{x}_{l,j}^{k}, \forall i \in N, \forall j \in N - \{s\}, \forall k \in K,$$

// $\tilde{y}$ represents the precedence relationships at $t+1$

$$\tilde{y}_{i,j}^{k} \leq y_{i,j}^{k}, \forall i \in N, \forall j \in N, \forall k \in K,$$

$$z_i \sim Bern(p_i), \forall i \in N,$$

// random variables with Bernoulli distribution (parameters p_i)

$$x_{i,j}^{k}, y_{i,j}^{k}, \tilde{x}_{i,j}^{k}, \tilde{y}_{i,j}^{k} \in \{0,1\}, \forall i \in N, \forall j \in N, \forall k \in K$$

Looking for a Solution

- Previous model not easily tractable.
- Three heuristic solutions:
    - two Greedy simple algorithms
    - a Greedy Randomized Adaptive Search Procedure (GRASP)

Solving with GRASP

- GRASP well known metaheuristic successfully used to solve many hard combinatorial optimization problems.
- Iterative process which operates in two phases: during *Construction Phase* an initial feasible solution is built whose neighborhood is then explored in the *Local Search Phase*.
- Construction phase: based on identifying disconnected subtrees $\tau_{k,i}$, and reconnecting them back (randomly and greedily) by connecting their roots to a node in main tree respecting bandwidth constraints.
- Local search phase: explores neighborhood obtained applying a random neural network to generate alternative trees at levels k.
- The construction and local search phase are repeated a previously fixed number of times, the best result is retained.

GRASP pseudo-code

Procedure GRASP_P2P_Network;
Input: G, k;

$a\_list \leftarrow$ empty_list;
$current\_psqa \leftarrow 0$; $best\_psqa \leftarrow -1$;
for $i = 1$ to MAXITER do
  $[\mathcal{G}_{sol}, a\_list] \leftarrow$ P2P_Const_Phase(G, a_list);
  $[\mathcal{G}_{sol}, a\_list] \leftarrow$ P2P_Local_Search($\mathcal{G}_{sol}$, k);
  $current\_psqa \leftarrow$ PSQA_Expected($\mathcal{G}_{sol}$) ;
  if ($current\_psqa > best\_psqa$) then
    $best\_psqa \leftarrow current\_psqa$;
    $\mathcal{G}_{best} \leftarrow \mathcal{G}_{sol}$;
  end_if;
  $a\_list \leftarrow$ empty_list;
end_for
return $\mathcal{G}_{best}$;

Procedure P2P_Const_Phase;
Input: G; a_list, n;

$p\_a\_list \leftarrow$ Possible_Assignm
;
while not_empty(p_a_list) do
  $RCL \leftarrow$ Compute_RCL(p_a_list, n);
  $current\_a \leftarrow$ SelectRandom(RCL);
  a_list
Add_Assignment(a_list, $current\_a$);
  p_a_list
Update_PA(G, $current\_a$, p_a_list);
end_while;
$\mathcal{G}_{sol} \leftarrow$ Apply_Assignments(G, a_list);
return $\mathcal{G}_{sol}$, a_list;

Figure: Customized GRASP and Construction Phase.

Comparison on a real-life based scenario

- We compare P2P vs. traditional Content Distribution Network (CDN) architectures.
- Dimensions of comparison: global perceived quality; total bandwidth consumption at broadcaster servers and other nodes.
- Case study based on simulation, scenario generated from statistical data from live video delivery service of a medium-sized ISP (10000 different users/month, 100 concurrent users per live-TV channel).

	CDN	P2P
Mean QoE	1	0.966
Servers BW	50 Mbps	5.6 Mbps
Clients BW	0 Mbps	0.6 Mbps/client

Under construction...

Concluding remarks

- PSQA allows to address the "ultimate target", the QoE, in a quantitative way.
- P2P multi-source schemes greatly improve the scalability of video streaming solutions, diminishing bandwidth requirements at servers.
- Dynamic nature of P2P network introduce degradations in users' QoE.
- Optimization models and metaheuristic algorithms allow to design rationally the P2P network connectivity, minimizing the impact of the peers behavior on perceived quality.
- Current work:
  - compare other algorithms for solving optimization problem.
  - development of a prototype for a P2P-based live video streaming system,
  - using multiple streams with redundancy policies,
  - with re-configuration of network topology based on