QoE optimisation in Software-Defined Networks

Jérémie Leguay
Head of Traffic and Network Optimization Team
Mathematical and Algorithmic Sciences Lab
Huawei FRC, Paris

February 2018, IRT System X
Network and Traffic Optimization Team

**Vision**

Solve **global**, **online** and **data-driven** optimization problems into next-generation network controllers

<table>
<thead>
<tr>
<th>Jeremie Leguay (TL)</th>
<th>Lei Shi (PM)</th>
<th>Stefano Paris</th>
<th>Paolo Medagliani</th>
<th>Lorenzo Maggi</th>
<th>Mathieu Leconte</th>
<th>Lazaros Gkatzikis</th>
<th>Zaid Allybokus (PhD)</th>
</tr>
</thead>
</table>

**Tools and Skills**

Optimization theory (combinatorial, stochastic), Game theory, Graph theory, Control theory, Statistical Learning, Algorithms.

**Topics**

Routing, Resource allocation, Placement, Monitoring
The (new) paradigm: SDN

Traditional networking

Software-Defined Networking

Data-plane

Control-plane

Switch

Programmable Switch

Data-plane

Control-plane

Data-plane

Control-plane

Data-plane

Control-plane

Data-plane

Data-plane

Data-plane
Network Automation and Optimization with SDN

- **Main properties of SDN / PCE**
  - **Offload** the control plane to (powerful) external x86 servers
  - Provide **network programmability** through abstractions

- **Operational benefits**
  - Advanced **automation** O&M 60%
  - Global **optimization and control**
    - Network efficiency **10 times**
Intelligence (ML) Defined Network - Towards Full Autonomic Network

**Manually Defined Network**
- **Basic approach**
  - Manually plan/configure/optimize/diagnose/debug network
- **Characteristics**
  - Fully depend on human’s experience and knowledge
  - High cost, low efficiency
  - Resource waste
  - Configuration mistakes
  - Network throughput is empirical

**Software Defined Network**
- **Basic approach**
  - Derive/abstract forwarding table from application requirements, use Netconf/YANG or OpenFlow to configure the network forwarding table
- **Characteristics**
  - Semi-autonomic
  - Cost reduced
  - Network throughput is improved, but not yet optimal

**Intelligence Defined Network**
- **Basic approach**
  - Machine learning: learn traffic and application patterns
  - Extensive use of optimization tools
  - Re-define and drive the network by machine learning technologies.
- **Characteristics**
  - Fully-autonomic
  - Minimum cost
  - Maximum throughput
Routing systems in next generation controllers

Global Routing Optimization Platform

Service Requests
- Admission Control
  - Multi-constraints, inclusion, Protection, Fast recovery
- Routing solver
  - Multi-objectives, protection, flow splitting, placement...
- Deployment policy
  - Hot-re-routing sequence, deadlock avoidance, stability
- Analysis system
  - Fault Analysis
    - Fault analysis, multi failures, fault analysis based on prediction information...
  - Traffic Prediction
    - Traffic prediction, congestion...
  - Traffic Estimation
    - Traffic matrix estimation, flow detection...
-A - Monitoring system
  - Monitoring
    - Real-time monitoring, packet loss, packet jitter,...
  - Topology
    - Topology discovery, updates, SRLG...

North Bound Interface

South Bound Interface

Our focus
- Online routing optimization
- Real-time and fair resource allocation
- Routing with traffic predictions
- Experience-driven routing

Ongoing transformations of network control planes:
- Convergence of monitoring, traffic analysis and routing systems
- (Logically) centralized on powerful software platforms

Better algorithms
- Real-time, more informed, computer intensive, scalable
Built-in Machine Learning

e.g., Time Series Data Repository in ODL

https://wiki.opendaylight.org/view/Project_Proposals:Time_Series_Data_Repository

Embedded Machine Learning tools are already available in network controllers
Built-in distributed / parallel computing
From routing protocols to distributed routing platforms

**Main challenges:**
- Break down optimization problems into many small ones which could be solved in parallel
- Take real-time decisions

**Architectural features:**
- Multi-domain networks
- Domain controllers are computer grids
- Hierarchical controllers

**Controllers perform collaborative tasks such as:**
- Routing optimization
- Flow migration
- Bandwidth allocation

**Domain controllers**
- SDN controller hierarchy
- Super Controller
- Domain controllers

**SDN controller hierarchy**
- Domain controller

**Architectural features:**
- Multi-domain networks
- Domain controllers are computer grids
- Hierarchical controllers

**Controllers perform collaborative tasks such as:**
- Routing optimization
- Flow migration
- Bandwidth allocation

**e.g., ODL SDNi**

**Diagram:**
- Super Controller
- Domain controller
- Domain controllers
Presentation outline

- Team introduction
- Short Intro to Video streaming and QoE
- QoE-aware Routing in Software Defined Networks
- QoE prediction with Machine Learning
Video traffic is predominant

Mobile Video Will Generate More Than Three-Quarters of Mobile Data Traffic by 2021

Source: Cisco Visual Networking Index
HTTP Adaptive Steaming (HAS)

The standard “de facto” for video stream retrieval is DASH

**What:** Video streaming solution where small pieces of video streams/files (*chunks*) are requested with HTTP and spliced together by the client. Client entirely controls delivery.

**Why:** Reuse widely deployed standard HTTP servers/caches for scalable delivery, e.g. existing Internet CDNs; traverse NAT/Firewalls; simple rate adaptation; fixed-mobile convergence; convergence of services, etc.

A complete description of the available chunks is provided into a Media Presentation Description (MPD) file exposed by the media server and consulted by each client.
DASH - What is specified and what is not?

Media Representations on HTTP server

- Segment
- Resources located by HTTP-URLs
- HTTP/1.1
- DASH access client
- Media engines
- DASH access engine
- on-time HTTP requests to segments

DASH client

- Media presentation description
- HTTP access client

Media engines

4 Mbits
1 Mbits

2s
2s
2s
2s
Buffer playback time evolution

- chunk arrival at times $t$
- stall
- video skip at times $s$
Quality of Experience measures

- **QoE factors (quantitative measurements)**
  - *Average video bitrate* of the downloaded chunks.
  - *Average bitrate variation*: the average of bitrate variations between consecutive chunks.
  - *Re-buffering ratio*: freezing (or stalling) time over the duration of the video session.

- **QoE scores (qualitative measurements)**
  - *MOS*: Mean Opinion Score
QoE-aware Routing in Software Defined Networks

Giacomo Calvigioni *, Ramon Aparicio-Pardo *, Lucile Sassatelli *, Jeremie Leguay +, Stefano Paris +, Paolo Medagliani +

(*) I3S Lab, Universite Cote d'Azur & CNRS
(+) Mathematical and Algorithmic Sciences Lab, Paris Research Center, Huawei
QoE-routing Setup

HAS Server
- Video split into chunks of 2s
- Multiple representations (qualities) of chunks

HAS Client
- Control HTTP request rate (e.g., buffer, buffer-rate)

Available bandwidth on path p
QoS parameters of path p (e.g., delay, hops, loss)
Client parameters of path p (e.g., HAS policy, content, resolution)

$QoE_n \left( x_p, \theta_p, \theta^n_{HAS} \right) = f^n_{HAS} \left( g^n_{TCP} \left( x_p, \theta_p \right), \theta^n_{HAS} \right)$

- Client/Server side: HAS control policies maximize the QoE, but are limited by resource allocation decisions (e.g., available bandwidth, latency, loss)
- SDN controller decisions play a key role for QoE maximization.
Mapping QoE to QoS: two ways

- Implicit functions (e.g., SVR, decision trees, etc.)
  - Requires a large amount of data (or time)
  - More difficult to introduce in an optimization model

- Explicit functions (e.g., log-logistic regression model)
  - Requires a fine knowledge of the system
  - Easy to introduce in an optimization model

Approach in this work
QoE Modeling – Linearization of $f_{HAS}^n(b, \theta_{HAS}^n)$

We want to express **compression** $x_c$ and **freezing** $x_f$ (i.e., rebuffering) as explicit functions of QoS metrics such as available bandwidth.

- Simulation of log-logistic model from ITU Rec. P.1202.2. for different
  - <HAS policy, content type, resolution> $\Rightarrow \theta_{HAS}^n$
  - Available bandwidth $\Rightarrow b$ (bandwidth seen by HTTPS)
- Piecewise linearization of the measured $QoE_n = f_{HAS}^n(b, \theta_{HAS}^n)$

$$QoE_n = \frac{1}{1 + a \left( a_{c_n} z_{c_n} b_{c_n}^{b_{c_n}} + a_{f_n} z_{f_n} b_{f_n}^{b_{f_n}} \right)^\beta}$$

$\forall n \leq N$

- Compression
- Freezing

$f_{HAS}^n(b, \text{<Buffer-Based-BufferLength>}, \text{‘Big Buck Bunny’, 1080p>})$

$f_{HAS}^n(b, \text{<Buffer-Based-2s>}, \text{‘Big Buck Bunny’, 1080p>})$
TCP–HAS Interplay

- Chunks download in HAS can be modeled as short-lived TCP connections [1]
  - On-off pattern (sequential HTTP req. every few seconds)
  - Mainly slow-start (specially at start/end of download)
  - High impact of packet loss

- Penalty for downloading video using HAS (small requests) vs. FTP (long requests):

\[
\alpha_p = -1.279 \times 10^{-3} + 1.0011 \cdot \text{RTD}_p
\]

Round Trip Delay (path p)

- Goodput-to-throughput ratio of HAS (path p)
  - It depends heavily on the RTD/RTT
  - It is worse than FTP

(*) The drop depends on the ns3 implementation of TCP cubic

---

Maximal HAS QoE-based Routing (Max-HQR) problem formulation

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>( a_k^d \in \mathbb{I}_{\geq 0} )</td>
<td>Slope of the straight-line section ( k \in K_d ) for the utility of demand ( d \in D )</td>
</tr>
<tr>
<td>( b_k^d \in \mathbb{I}_{\geq 0} )</td>
<td>y-intercept of the straight-line section ( k \in K_d ) for the utility of demand ( d \in D )</td>
</tr>
<tr>
<td>( r_d^{\text{min}} \in \mathbb{I}_{\geq 0} )</td>
<td>Required average bandwidth to get worst HAS representation allowed for demand ( d \in D )</td>
</tr>
<tr>
<td>( r_d^{\text{max}} \in \mathbb{I}_{\geq 0} )</td>
<td>Required average bandwidth to get best HAS representation allowed for demand ( d \in D )</td>
</tr>
<tr>
<td>( c_e \in \mathbb{I}_{\geq 0} )</td>
<td>Capacity of link ( e \in E )</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Variable</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>( u^d \in \mathbb{I}_{\geq 0} )</td>
<td>Utility value (QoE) of demand ( d \in D )</td>
</tr>
<tr>
<td>( x_p^d \in \mathbb{I}_{\geq 0} )</td>
<td>Available bandwidth on path ( p \in P ) to serve demand ( d \in D )</td>
</tr>
<tr>
<td>( z_p^d \in {0,1} )</td>
<td>Whether path ( p ) is used to serve demand ( d )</td>
</tr>
<tr>
<td>( z_{dpi} \in {0,1} )</td>
<td>Whether ( i \in \mathbb{Z}_{\geq 0} ) competitive videos shares the bottleneck of path ( p \in P ) used to serve demand ( d \in D )</td>
</tr>
<tr>
<td>( n_e \in \mathbb{I}_{\geq 0} )</td>
<td>Number of competitive videos sharing link ( e \in E )</td>
</tr>
</tbody>
</table>
Max-HQR – Mathematical Formulation

Assumption:

- Flow rates are determined by TCP fair-share of bottleneck link (not controlled)

\[
\begin{align*}
\text{max} & \quad \sum_{d \in D} \sum_{p \in P_d} \sum_{i>0} U_{dpi} \cdot z_{dpi} \\
\text{s.t.} & \quad \sum_{p \in P_d} z_p^d = 1 \\
& \quad \sum_{d \in D} \sum_{p \in P_d} z_p^d = n_e \\
& \quad z_p^d \cdot n_e \leq \frac{c_e}{C} \sum_{i>0} i \cdot z_{dpi} \\
& \quad \sum_{i>0} z_{dpi} = 1 \\
\end{align*}
\]

Utility (QoE) maximization

\[
x_p^d = \frac{c_e}{i} \Rightarrow U_{dpi} = U_d \left( \alpha_p \cdot \min \left\{ \frac{c_e}{i} : e \in E \right\} \right)
\]

Number of competing flows

Number of competing videos flows

Number of flows on the bottleneck of a demand

Non-linear constraint

Not efficiently solvable
Max-QoE – Mathematical Formulation

Note
• Control of the bandwidth reserved to HAS connections (in addition to path selection)

\[
\max_{\{x,z,u\}} \sum_{d \in D} u^d \quad \text{Utility (QoE) maximization}
\]

\[
s.t. \quad \sum_{d \in D} \sum_{p \in P_d} x_p^d \leq c_e \quad \forall e \in E \quad \text{Capacity constraint}
\]

\[
a_k^d \left( \sum_{p \in P_d} \alpha_p \cdot x_p^d \right) + b_k^d \geq u^d \quad \forall d \in D, k \in K_d \quad \text{Rate allocation linearization}
\]

\[
\sum_{p \in P_d} \alpha_p \cdot x_p^d \geq r_d^{\min} \cdot z_p^d \quad \forall d \in D \quad \text{Range of allocated bandwidth (between min and max bitrate)}
\]

\[
\sum_{p \in P_d} \alpha_p \cdot x_p^d \leq r_d^{\max} \cdot z_p^d \quad \forall d \in D
\]

\[
\sum_{p \in P_d} z_p^d = 1 \quad \forall d \in D \quad \text{Single path}
\]
Max-QoE – Lagrangian Relaxation

\[
\max_{\{x,u,z,\lambda\}} -L(x,u,z,\lambda) = \sum_{d \in D} u^d - \sum_{d \in D} \sum_{p \in P_d} \lambda_p \cdot x^d_p + \sum_{e \in E} \lambda_e \cdot c_e
\]

s.t.
\[
a_k^d \left( \sum_{p \in P_d} \alpha_p \cdot x^d_p \right) + b^d_k \geq u^d \quad \forall d \in D, k \in K_d
\]
\[
\sum_{p \in P_d} \alpha_p \cdot x^d_p \geq r^\text{min}_d \cdot z^d_p \quad \forall d \in D
\]
\[
\sum_{p \in P_d} \alpha_p \cdot x^d_p \leq r^\text{max}_d \cdot z^d_p \quad \forall d \in D
\]
\[
\sum_{p \in P_d} z^d_p = 1 \quad \forall d \in D
\]

Coupling constraints have been eliminated

Subproblems can be solved independently
Max-QoE – Lagrangian Relaxation

Primal subproblem

\[
\begin{align*}
\max_{\{x,z,u\}} & \quad u^d - \sum_{p \in P_d} \lambda_p \cdot x_p^d \\
\text{s.t.} & \quad a_k \left( \sum_{p \in P_d} \alpha_p \cdot x_p^d \right) + b_k^d \geq u^d \quad k \in K_d \\
& \quad \sum_{p \in P_d} \alpha_p \cdot x_p^d \geq r_{d \min} \cdot z_p^d \\
& \quad \sum_{p \in P_d} \alpha_p \cdot x_p^d \leq r_{d \max} \cdot z_p^d \\
& \quad \sum_{p \in P_d} z_p^d = 1
\end{align*}
\]

- **Path selection**: the path with the smallest ratio \( \frac{\lambda_p}{\alpha_p} \) is optimal.
- **Flow allocation**: optimal flow allocation:
  - if \( a_{k+1} \leq \frac{\lambda_p}{\alpha_p} \leq a_k \) \( \Rightarrow \) the bitrate of the intersection point of the linear pieces \( k \) and \( k+1 \)
  - if \( \frac{\lambda_p}{\alpha_p} \geq a_0 \) (or \( \frac{\lambda_p}{\alpha_p} \leq a_{|K|-1} \)) \( \Rightarrow \) the worst bitrate \( r_{k \min} \) (or the best bitrate \( r_{k \max} \))

Dual subproblem

\[
\begin{align*}
\min_{\{\lambda,\pi,\mu,v\}} & \quad \sum_{k \in K} \pi_k^d \cdot b_k^d - \mu_d \cdot r_{d \min} + v_d \cdot r_{d \max} \\
\text{s.t.} & \quad \sum_{k \in K} \pi_k^d \geq 1 \\
& \quad \frac{\lambda_p}{\alpha_p} \geq \left( \sum_{k \in K} \pi_k^d \cdot a_k^d - \mu_d + v_d \right) \quad \forall p \in P_d \\
& \quad \pi_k^d \in \mathbb{R}^+ \quad \forall k \in K \\
& \quad \mu_d \in \mathbb{R}^+ \\
& \quad v_d \in \mathbb{R}^+
\end{align*}
\]
Dual Subgradient based on Lagrangian Relaxation (DGLR)

- **Initialization:**
  - \( \text{LB} = -\infty; \lambda_e = 0, \forall e \in E \)
  - Compute K shortest paths for each demand \( \forall d \in D \) (delay as link metric)

- **While** \( (t \leq N_{\text{max}}) \) and (dual gap \( \leq \vartheta \))
  1. Solve \(|D|\) subproblems (compute \( L_d^{(t)}(x, \lambda) \), \( \forall d \in D \))
     - Select the path \( p \) with the smallest ratio \( r_p(t) = \frac{\lambda_p(t)}{\alpha_p(t)} = \frac{\sum_{e \in p} \lambda_e(t)}{\alpha_p(t)} \)
     - Perform the flow allocation \( x \) (compare \( r_p \) with slopes \( a_k \))
  2. Aggregate subproblems: \( L^{(t)}(x, \lambda) = \sum_{d \in D} L_d^{(t)}(x, \lambda) + \sum_{e \in E} \lambda_e c_e \)
  3. Update lower bound: \( \text{LB} = \begin{cases} 
  L^{(t)}(x, \lambda) & \text{if } L^{(t)}(x, \lambda) > \text{LB} \\
  \text{LB} & \text{if } L^{(t)}(x, \lambda) \leq \text{LB} 
  \end{cases} \)
  4. Compute the gradient vector: \( G_e(t) = \sum_{d \in D} \sum_{p \in P_e} x_p^d(t) - c_e \)
  5. Update multipliers: \( \lambda_e(t + 1) = max\{0, \lambda_e(t) + \gamma \cdot G_e(t)\} \)
  6. Update iteration counter: \( t = t + 1; \)

- **Constant stepsize**
Numerical Results

- **Fully controllable environment at network and streaming levels**
  - Adaptive Multimedia Streaming Simulator Framework (**AMust**) in **NS3**
  - Implements a HTTP client and server for DASH
  - Integrates **LibDASH** (reference software of ISO/IEC MPEG-DASH standard)

In the **network**, we can control:
- The topology and link properties (latency, jitter, loss rate, capacity)
- TCP parameters (e.g., congestion control algorithm, segment size)
- Drop policy (e.g., Red)
- ...

On each **client**, we can control:
- Sessions (video id, start time, stop time)
- Screen properties (width, height)
- DASH parameters (AdaptationLogic, AllowDownscale, AllowUpscale, Buffer duration)

Large set of DASH policies: Rate based, Buffer based, Rate and Buffer based, AlwaysLowest, no adaptation, custom.

**Code is available here:** [https://github.com/sassatelli/QoErouting](https://github.com/sassatelli/QoErouting)
Numerical Results

- **Scenario**
  - 3 representative movies:
    - Big Buck Bunny (BBB) – cartoon with a mix of low and high motion scenes
    - The Swiss Account (TSA) – sport video with high motion
    - Red Bull Play Street (RBPS) - sport documentary with regular motion
  - **GEANT network** (22 nodes and 36 links)
    - downsized link capacity to 10 Mbps, one-way latency in [1; 10] ms
    - **DASH servers** are attached to 1 node
    - **DASH clients** are randomly attached to others.

- **Performance benchmarks**
  - Optimal solution solving the MILP with CPLEX
  - QoS routing solution using **LARAC [1]**
    - Each demand is routed over the residual capacity using solving a min cost with QoS constraint problem (e.g., latency)
    - Lagrange Relaxation Based Aggregated Cost (LARAC) algorithm

Numerical Results

- **Performance indicators**
  - **Average video bitrate** of the downloaded chunks.
  - **Average quality variation**: the standard deviation of the quality index which quantifies quality changes over the different downloaded chunks.
  - **Average video quality [0,1]**: Average on all downloaded chunks of a normalized quality index indicating to which representation they belong.
  - **Average quality fairness [0,1]**: Jain’s index over the average quality index of all video sessions.
  - **Re-buffering ratio**: freezing (or stalling) time over the duration of the video session.

<table>
<thead>
<tr>
<th>Representation 1: 500k</th>
<th>Representation 2: 1M</th>
</tr>
</thead>
<tbody>
<tr>
<td>(500k + 1M + 500k + 500k)/4</td>
<td>(500k + 500k)/2</td>
</tr>
<tr>
<td>(0.5 + 1 + 0.5 + 0.5)/4</td>
<td>20%</td>
</tr>
</tbody>
</table>

![Diagram showing time t1, t2, t3, t4, t5]
Numerical Results

- Static traffic scenario on GEANT (100s – all video sessions active)
Numerical Results

- Dynamic scenario on GEANT (160s – all video sessions active)
  - 60 demands with mean duration 30s (Poisson distributed).
Future Works

1. The QoE depends on the available bandwidth, but…
   - The model of QoE is calibrated before the optimization
   - The optimization decides the available bandwidth using the QoE model

Model calibration

\[ QoE_n(\hat{x}_p, \theta_p, \theta_{HAS}^n) \]

- Bandwidth \( x_p \) is fixed
- We measure QoE
- We get linear coefficients of QoE

Optimization

\[ \max_{\{x,z\}} QoE_n(x_p, \theta_p, \theta_{HAS}^n) \]

s.t. some constraints

- We get the optimal \( x^*_p \)

Observation

\[ QoE_n(x^*_p, \theta_p, \theta_{HAS}^n) \]

- We measure the real QoE
- We want:

\[ QoE_n(\hat{x}_p, \theta_p, \theta_{HAS}^n) = QoE_n(x^*_p, \theta_p, \theta_{HAS}^n) \]

Model recalibration

1. Allocate a minimum reserved bandwidth to each session
   1. Requires a proper test environment (no support from ns3 at the moment)
2. Machine learning for model recalibration
   - We can use the new observation to learn the unknown parameters of the QoE model
QoE prediction with Machine Learning

Vladislav Vasilev, Jérémie Leguay, Stefano Paris, Lorenzo Maggi, Merouane Debbah

Mathematical and Algorithmic Sciences Lab
Huawei FRC, Paris
The importance of QoS for QoE

- QoS is a poor predictor of experienced QoE
- QoE is the only relevant parameter for end users
  - 3 main parameters to measure user-perceived video\[^{[1]}\]:
    - **Rebuffering ratio**: how often rebuffering occurs during playback.
    - **Average video bitrate**: How much information is received in the play out buffer.
    - **Average video bitrate variation**: how much the player increases or decreases the bitrate every time it switches bitrate during playback.
  - Monitoring **rebuffering and bitrate change** ratios is a good metric to quantify video abandonment rates for short videos such as YouTube
- However,
  - network controllers only deal with QoS parameters
  - Most of video traffic is now encrypted

Can we predict QoE based on QoS observations and exploit this prediction for routing?

Production of high-fidelity data

- Fully controllable environment at network and streaming levels
  - Adaptive Multimedia Streaming Simulator Framework (AMust) in NS3
  - Implements a HTTP client and server for DASH
  - Integrates LibDASH (reference software of ISO/IEC MPEG-DASH standard)

On each client, we can control:
- Sessions (video id, start time, stop time)
- Screen properties (width, height)
- DASH parameters (AdaptationLogic, AllowDownscale, AllowUpscale, Buffer duration)

On the server, we can control:
- Media representations
- Fake or real video segments

In the network, we can control:
- The topology and link properties (latency, jitter, loss rate, capacity)
- TCP parameters (e.g., congestion control algorithm, segment size)
- Drop policy (e.g., Red)
- ...

Large set of DASH policies: Rate based, Buffer based, Rate and Buffer based, AlwaysLowest, no adaptation, custom.

Simulating a large number of scenarios, statistics for 80k video sessions (public)
Research questions in this work

- **Network-level measurements**
  - **TCP statistics**: number of packets, avg packet delay, avg packet jitter, avg packet loss
  - **Downloading rate** (sampled every 2s): average, 5, 10, 25, 50, 75, 90, 95 quantiles, standard deviation
  - **Inter-arrival times of segment requests**: average, 5, 10, 25, 50, 75, 90, 95 quantiles, standard deviation
  - **Network congestion**: number of concurrent streams, bottleneck capacity, bottleneck delay, bottleneck loss
  - **Player characteristics**: DASH policy, client resolution, max video buffer.

- **QoE factors**
  - **Average video bitrate**
  - **Average video bitrate variation**
  - **Rebuffering ratio**

  Question 1:
  Can we predict QoE from QoS measures on each stream?

  Question 2:
  Can we improve the prediction when context information is available?

Well correlated with the network rate

The difficult one
Preliminaries: Data Characteristics

Network rate
Sampled over bins of 2s

Inter-arrival times of HTTP requests
In mobile networks, all HTTP(s) requests go through HTTP proxies

- RebufferingRatio = 0 → No staling-
- 0 < RebufferingRatio < 0.1 → Mild staling
- RebufferingRatio ≥ 0.1 → Severe staling
Preliminaries: Using Hidden Variables
i.e. variables we may access only at training phase

Bagged Regression Tree and Bagged Random Forest are used for the additional variable prediction based on whether they are continuous or discrete.
The distribution of the target variables is somewhat exponential.

This exponential pattern occurred also for the input, hidden and simulation variables.
Preliminaries: Data Characteristics

The figures show the relations of 2 variables with high mutual information with 2 QoE variables. Clearly there are locally linear dependencies in the data.

Average Bitrate and Average Bitrate variations are the easy ones
StallLabel Prediction: State of the art.

Random Forest (RF)

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.96178</td>
<td>0.95525</td>
</tr>
<tr>
<td>0.05</td>
<td>0.7585</td>
<td>0.73587</td>
</tr>
<tr>
<td>0.1</td>
<td>0.43874</td>
<td>0.34211</td>
</tr>
</tbody>
</table>

Random Forest (RF) pruned to minimum leaf size of 50.

- A RF is a bagging of Decision Tree (DT) models.
- Each DT at a leaf node greedily selects and splits an input variable given all the accumulated splits from the root to the leaf.
- The bagging minimizes the effect of local optimality caused by the greedy split procedure.

The performance on the SevereStall class is practically unacceptable.

[Louppe], [Dimopoulos].
StallLabel Prediction: RF Analysis

There are 2 most commonly occurring problems with RF.

- The RF greedy split procedure might result in low quality local optimum.
- The RF's rectangular decision regions have boundaries parallel to the basis of the dimensions, which could fail to capture some linear dependencies.

Confirming, that the RF decision regions are not adequate for our data.

- There are many correlations between the input variables.
- If we change the data basis to a new basis specified by the parameters of a Logistic Regression (LR) classifier we get an improvement on the RF.

[Bishop],[Ng]
StallLabel Prediction: BN definition

Define the BN in Fig. 2.1. If the two LRxs are trained separately the performance is degraded. Also note that $\theta_1$ and $\theta_2$ are not d-separated [Koller] only when mild or severe stall is observed.

\[
Y = (\text{NoStall} = 0, \text{SomeStall} \neq 0)
\]

1st LR: $P_x(Y_i|\theta_1) \xleftarrow{} \theta_1$

If $P_x(Y_i|\theta_1) \geq \alpha \in [0, 1]$

No Stall \quad 2nd LR: $P_x(Y_i|\theta_2, P_x(Y_i|\theta_1) > \alpha) \xrightarrow{} \theta_2$

Mild Stall \quad Severe Stall

\[
P_x(Y, \theta_1, \theta_2) = [P_x(Y_i|\theta_1)P(\theta_1)] \cdot [P_x(Y_i|\theta_2, P_x(Y_i|\theta_1) > \alpha)P(\theta_2)]
\]

Table 2.2: Performance of the proposed BN model

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.8681</td>
<td>0.8684</td>
</tr>
<tr>
<td>0.05</td>
<td>0.7929</td>
<td>0.8048</td>
</tr>
<tr>
<td>0.1</td>
<td>0.9338</td>
<td>0.9368</td>
</tr>
</tbody>
</table>

Figure 2.1: Bayesian Network using LR models.
### StallLabel Prediction: Results

<table>
<thead>
<tr>
<th>Cross.val. BN</th>
<th>Tr.acc. 0</th>
<th>Tr.acc. 0.05</th>
<th>Tr.acc. 0.1</th>
<th>Val.acc. 0</th>
<th>Val.acc. 0.05</th>
<th>Val.acc. 0.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input only</td>
<td>Mean 0.8685</td>
<td>0.7946</td>
<td>0.9376</td>
<td>0.8665</td>
<td>0.7982</td>
<td>0.9267</td>
</tr>
<tr>
<td></td>
<td>STD 0.0004</td>
<td>0.0014</td>
<td>0.0031</td>
<td>0.0018</td>
<td>0.0045</td>
<td>0.0122</td>
</tr>
<tr>
<td>Added predicted simulation</td>
<td>Mean 0.8691</td>
<td>0.7963</td>
<td>0.9409</td>
<td>0.8691</td>
<td>0.7996</td>
<td>0.9316</td>
</tr>
<tr>
<td></td>
<td>STD 0.0004</td>
<td>0.0008</td>
<td>0.0031</td>
<td>0.0021</td>
<td>0.0036</td>
<td>0.0017</td>
</tr>
<tr>
<td>Added actual simulation</td>
<td>Mean 0.8689</td>
<td>0.7975</td>
<td>0.9396</td>
<td>0.8676</td>
<td>0.8023</td>
<td>0.9304</td>
</tr>
<tr>
<td></td>
<td>STD 0.0001</td>
<td>0.0010</td>
<td>0.0027</td>
<td>0.0023</td>
<td>0.0033</td>
<td>0.0018</td>
</tr>
<tr>
<td>Added predicted hidden</td>
<td>Mean 0.8758</td>
<td>0.8012</td>
<td>0.9434</td>
<td>0.8735</td>
<td>0.7983</td>
<td>0.9366</td>
</tr>
<tr>
<td></td>
<td>STD 0.0018</td>
<td>0.0033</td>
<td>0.0019</td>
<td>0.0034</td>
<td>0.0101</td>
<td>0.0079</td>
</tr>
<tr>
<td>Added actual hidden</td>
<td>Mean 0.9000</td>
<td>0.8399</td>
<td>0.9543</td>
<td>0.8990</td>
<td>0.8466</td>
<td>0.9530</td>
</tr>
<tr>
<td></td>
<td>STD 0.0012</td>
<td>0.0023</td>
<td>0.0026</td>
<td>0.0020</td>
<td>0.0091</td>
<td>0.0113</td>
</tr>
</tbody>
</table>

### VIDEO QUALITY

<table>
<thead>
<tr>
<th>Case</th>
<th>TR.</th>
<th>Val.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input only</td>
<td>Mean 5.8139</td>
<td>6.8786</td>
</tr>
<tr>
<td></td>
<td>STD 0.0599</td>
<td>0.0172</td>
</tr>
<tr>
<td>Simul. Predicted</td>
<td>Mean 5.5994</td>
<td>6.6195</td>
</tr>
<tr>
<td></td>
<td>STD 0.0524</td>
<td>0.0105</td>
</tr>
<tr>
<td>Simul. Actual</td>
<td>Mean 4.7183</td>
<td>6.3470</td>
</tr>
<tr>
<td></td>
<td>STD 0.0757</td>
<td>0.0234</td>
</tr>
<tr>
<td>Hidden/Simul. Predicted</td>
<td>Mean 5.3013</td>
<td>6.4341</td>
</tr>
<tr>
<td></td>
<td>STD 0.0756</td>
<td>0.0164</td>
</tr>
<tr>
<td>Hidden/Simul. Actual</td>
<td>Mean 3.4870</td>
<td>4.6005</td>
</tr>
<tr>
<td></td>
<td>STD 0.0604</td>
<td>0.0127</td>
</tr>
<tr>
<td>Normalization for all values</td>
<td>1.0e+04</td>
<td>*</td>
</tr>
</tbody>
</table>

Video Quality. Using variables [38:42,44,45,51] from Tab. I.

The figure on the right shows the AvgVideoBitRate absolute difference in the prediction as a distribution.
VIDEO QUALITY VARIATION

<table>
<thead>
<tr>
<th>Case</th>
<th>TR. Mean</th>
<th>STD</th>
<th>Val. Mean</th>
<th>STD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input only</td>
<td>4.3335</td>
<td>0.0233</td>
<td>5.1738</td>
<td>0.0697</td>
</tr>
<tr>
<td>Simul. predicted</td>
<td>3.7510</td>
<td>0.0253</td>
<td>4.6414</td>
<td>0.0597</td>
</tr>
<tr>
<td>Simul. Actual</td>
<td>2.8882</td>
<td>0.0179</td>
<td>4.1241</td>
<td>0.0568</td>
</tr>
<tr>
<td>Hidden/Simul. Predicted</td>
<td>3.6783</td>
<td>0.0265</td>
<td>4.6099</td>
<td>0.0618</td>
</tr>
<tr>
<td>Hidden/Simul. Actual</td>
<td>2.6231</td>
<td>0.0134</td>
<td>3.7784</td>
<td>0.0384</td>
</tr>
<tr>
<td>Normalization for all</td>
<td>1.0e+04</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Video Quality Variation. Using variables [38:42,44,45,51] from Tab. I.

The figure on the right shows the AvgVideoBitRate absolute difference in the prediction as a distribution.
Conclusion

- Focused on the StallLabel variable: Proposition of a Bayesian network to well balance class accuracies.

- Proposition to make intermediate prediction for additional (hidden) variables to improve accuracy.

- We show that adding network-level information (overall load or bottleneck characteristics) improves QoE prediction.
REFERENCES


Andrew Ng, CS229 Lecture notes, pp: 1–3, 2000.


Future directions on QoE-aware networking

- Extend ML models to improve QoE predictions
- Use ML models for QoE-based routing
- Control bandwidth allocation or AQM parameters
- Network assisted DASH policies
Selected publications (network optimization for video traffic)

- **Quality of Experience-based Routing of Video Traffic for Overlay and ISP Networks.** Giacomo Calvigioni, Ramon Aparicio-Pardo, Lucile Sassatelli, Jeremie Leguay, Stefano Paris, Paolo Medagliani. *IEEE INFOCOM 2018*.


