QoE optimisation in Software-Defined Networks

Jérémie Leguay

Head of Traffic and Network Optimization Team

Mathematical and Algorithmic Sciences Lab
Huawei FRC, Paris

www.huawei.com

February 2018, IRT System X

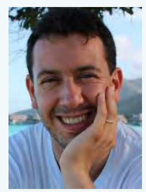
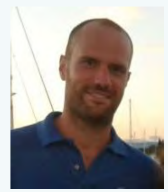
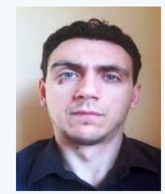
HUAWEI TECHNOLOGIES CO., LTD.

HUAWEI

Network and Traffic Optimization Team

Vision

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

Jeremie Leguay (TL)	Lei Shi (PM)	Stefano Paris	Paolo Medagliani	Lorenzo Maggi	Mathieu Leconte	Lazaros Gkatzikis	Zaid Allybokus (PhD)
							
Network optim. Protocols, Traffic eng.	routing, OAM, simulatio, evaluatio, optim.	Network optim. Protocols, Game theory, Traffic eng.	Network optim. Protocols, Traffic engineering	Network optim. Control Theory. Machine Learning	Netwok optim. Graph theory. Machine Learning	Network optim. Resource allocation, caching,	Game Theory, Optimization Operations Research

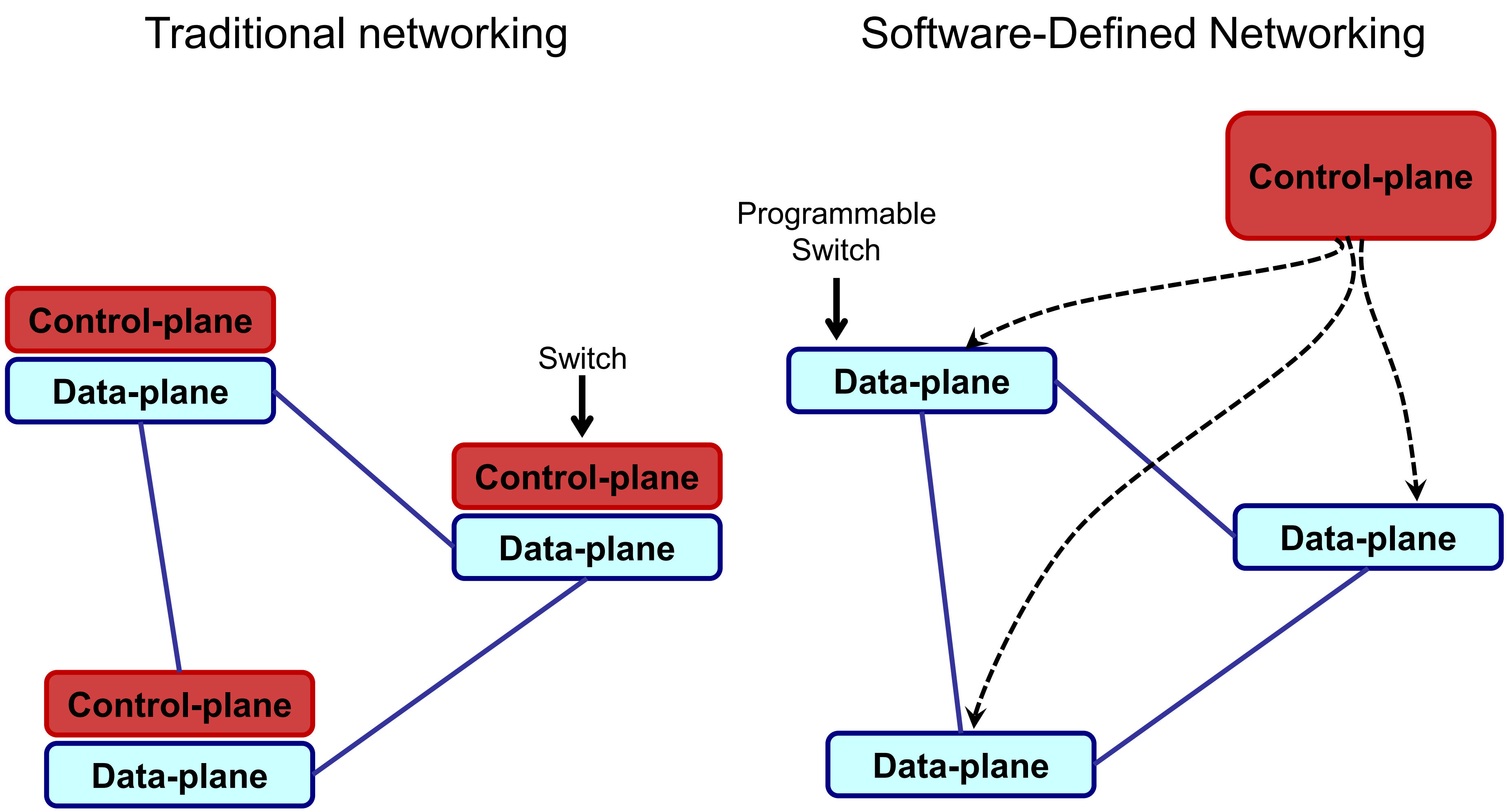
Tools and Skills

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

Topics

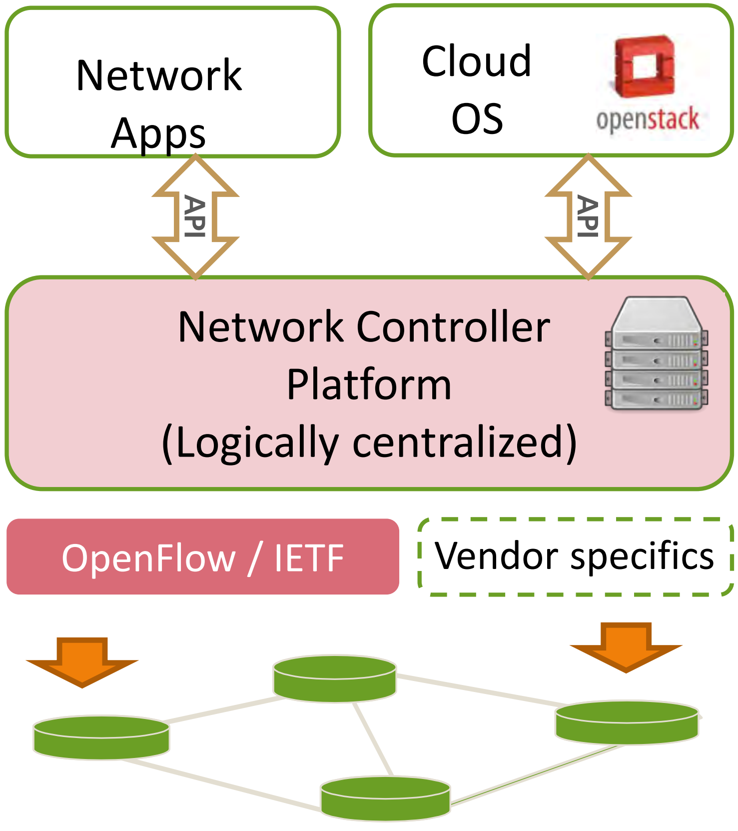
Routing, Resource allocation, Placement, Monitoring

The (new) paradigm: SDN



Network Automation and Optimization with SDN

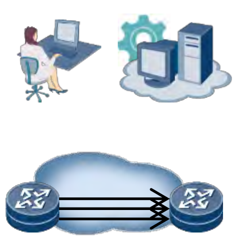
- Main properties of SDN / PCE
  - **Offload** the control plane to (powerful) external x86 servers
  - Provide **network programmability** through abstractions
- Operational benefits
  - Advanced **automation** O&M **60%**
  - Global **optimization and control**
    Network efficiency **10 times**



SDN architectures

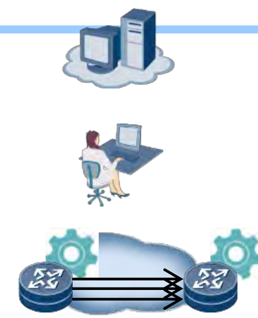
Intelligence (ML) Defined Network
- Towards Full Autonomic Network

Manually Defined Network



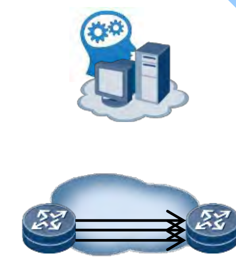
- **Basic approach**
  - Manually plan/ configure/ optimize/ diagnose/ debug network
- **Characteristics**
  - Fully depend on human's experience and knowledge
  - High cost, low efficiency
  - Resource waste
  - Configuration mistakes
  - Network throughput is empirical

Software Defined Network



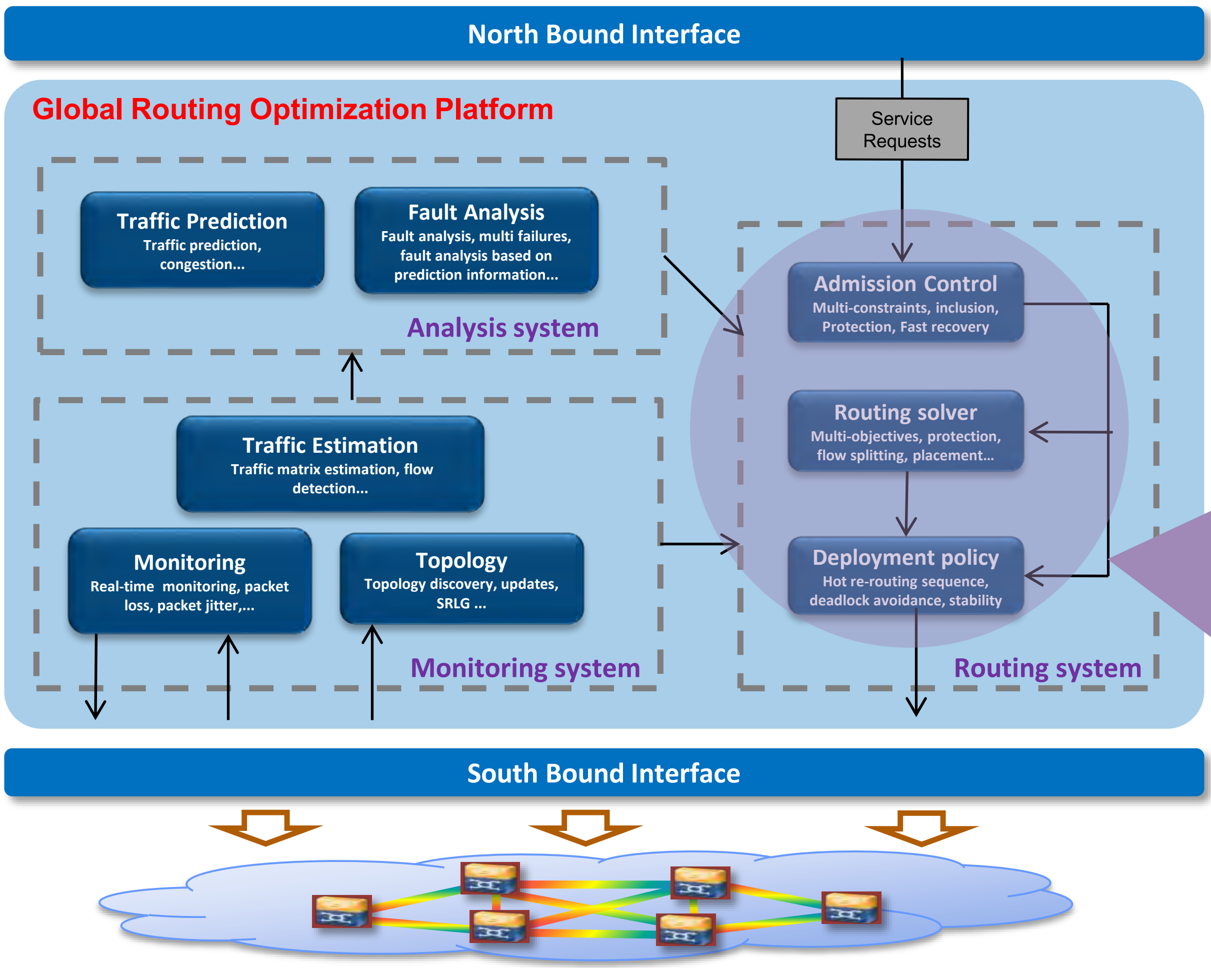
- **Basic approach**
  - Derive/abstract forwarding table from application requirements, use Netconf/YANG or OpenFlow to configure the network forwarding table
- **Characteristics**
  - Semi-autonomic
  - Cost reduced
  - Network throughput is improved, but not yet optimal

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



Our focus

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

Ongoing transformations of network control planes:

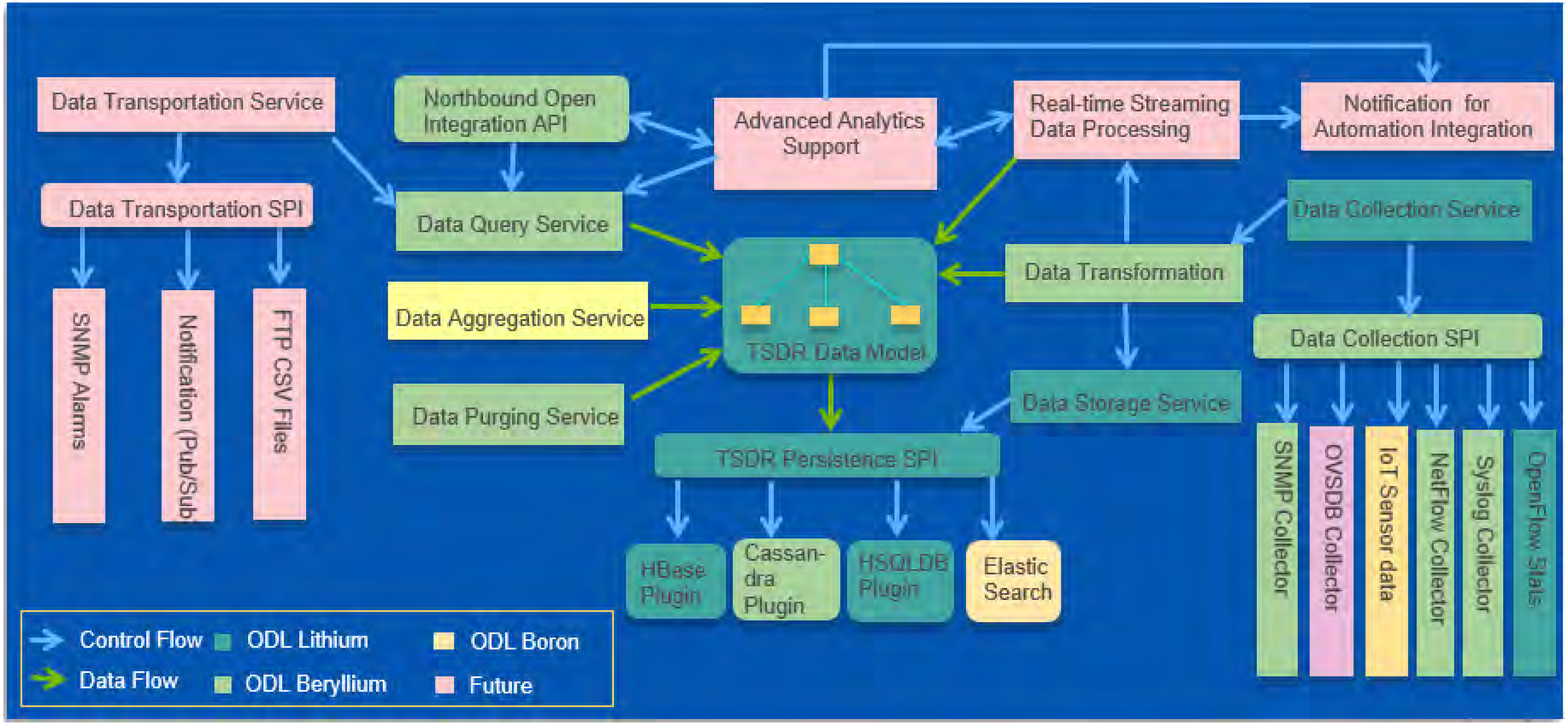
- Convergence of monitoring, traffic analysis and routing systems
- (Logically) centralized on powerful software platforms

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

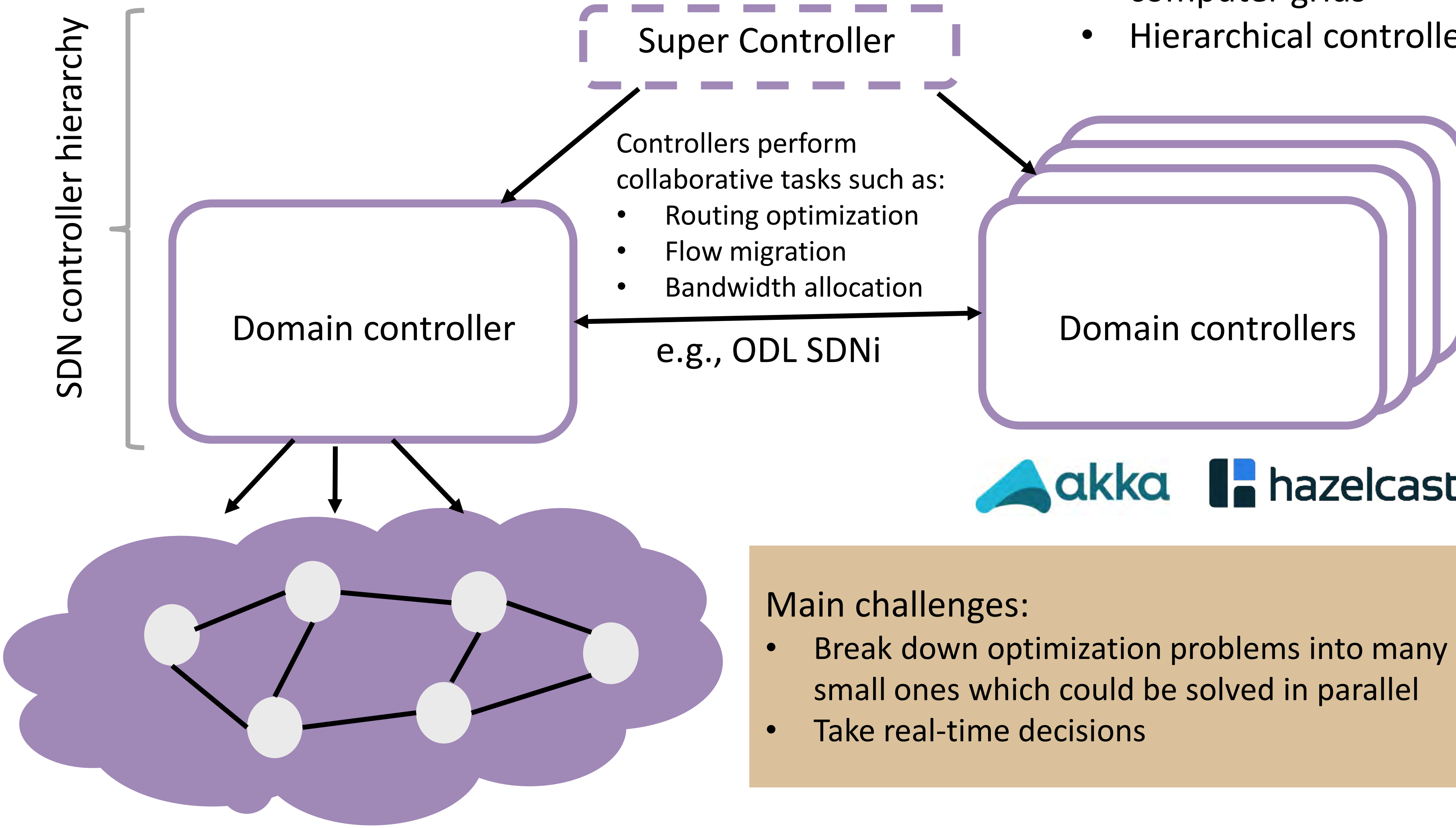
Embedded Machine Learning tools are already available in network controllers

Built-in distributed / parallel computing

From routing protocols to distributed routing platforms

Architectural features:

- Multi-domain networks
- Domain controllers are computer grids
- Hierarchical controllers



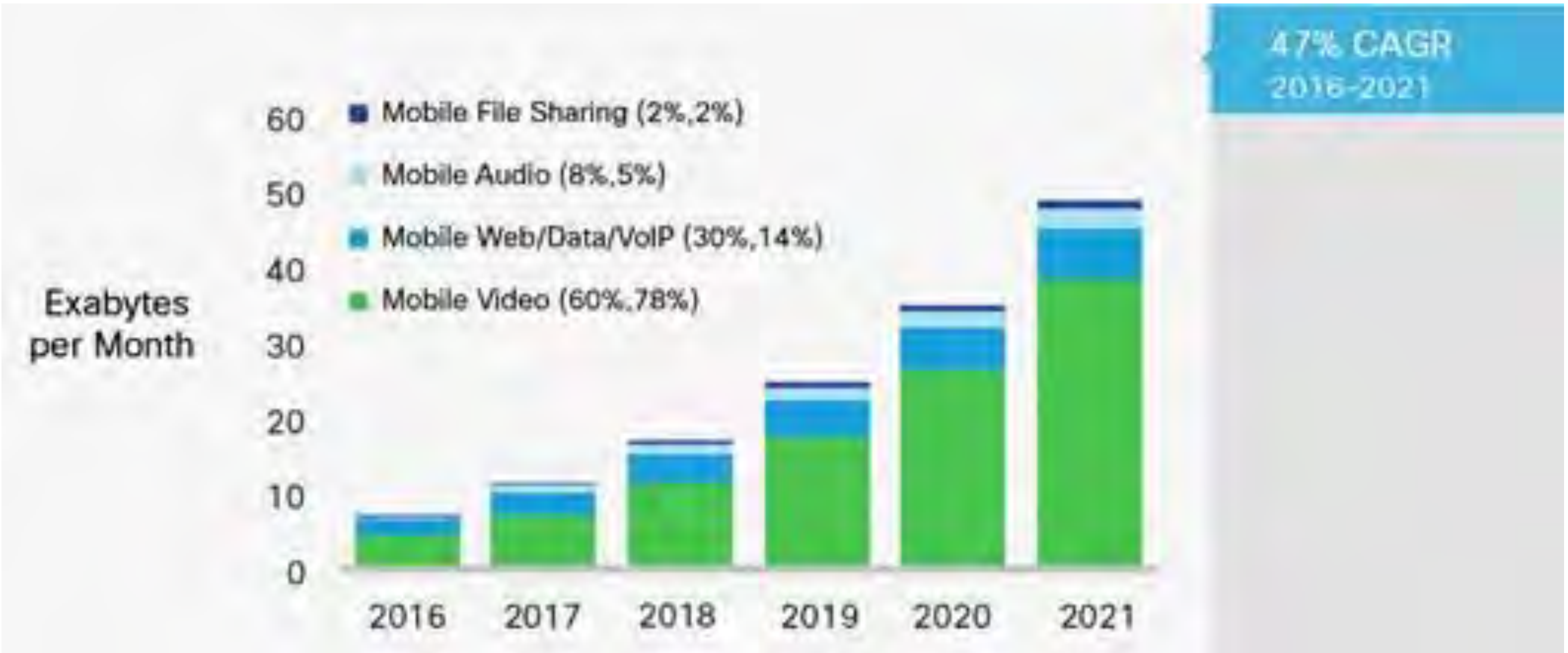
Main challenges:

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

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

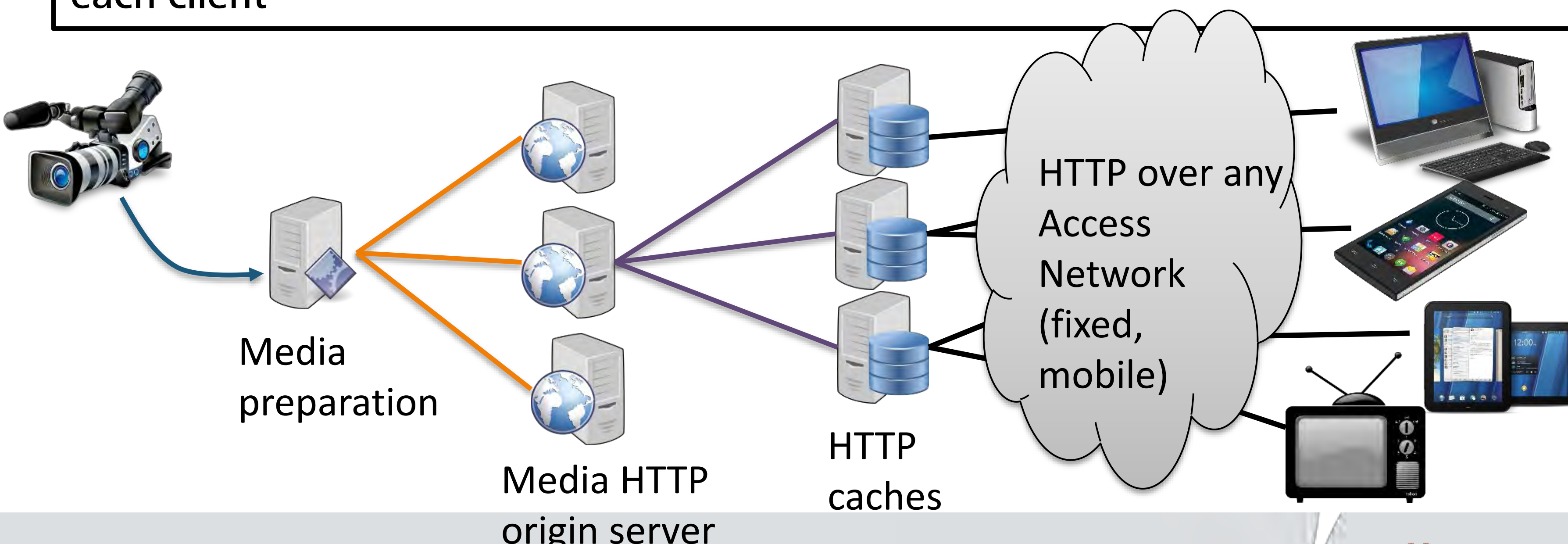
HTTP Adaptive Steaming (HAS)

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

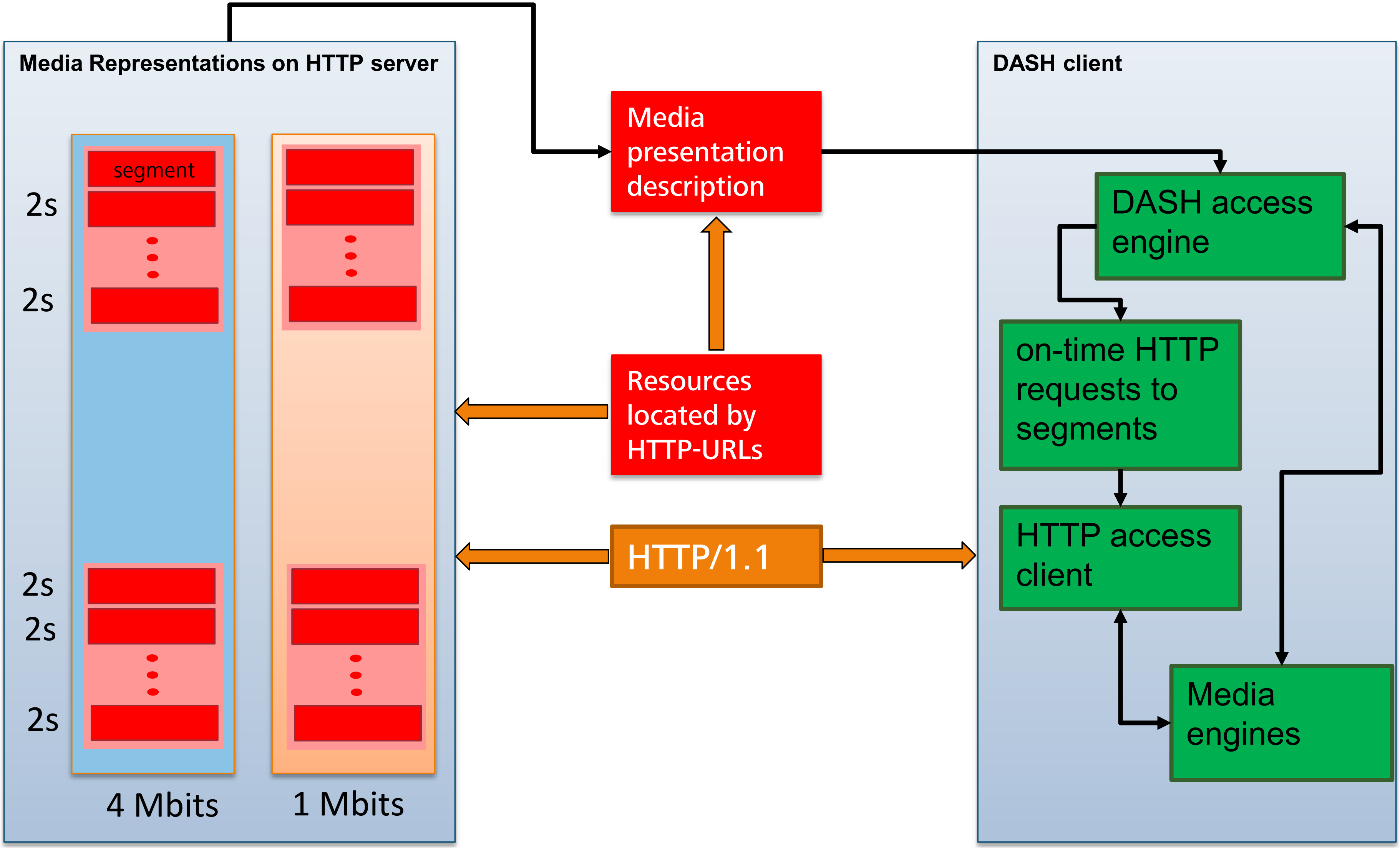
What: Video streaming solution where small pieces of video streams/files (*chunks*) are requested with HTTP and spliced together by the client. Client entirely controls delivery.

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.

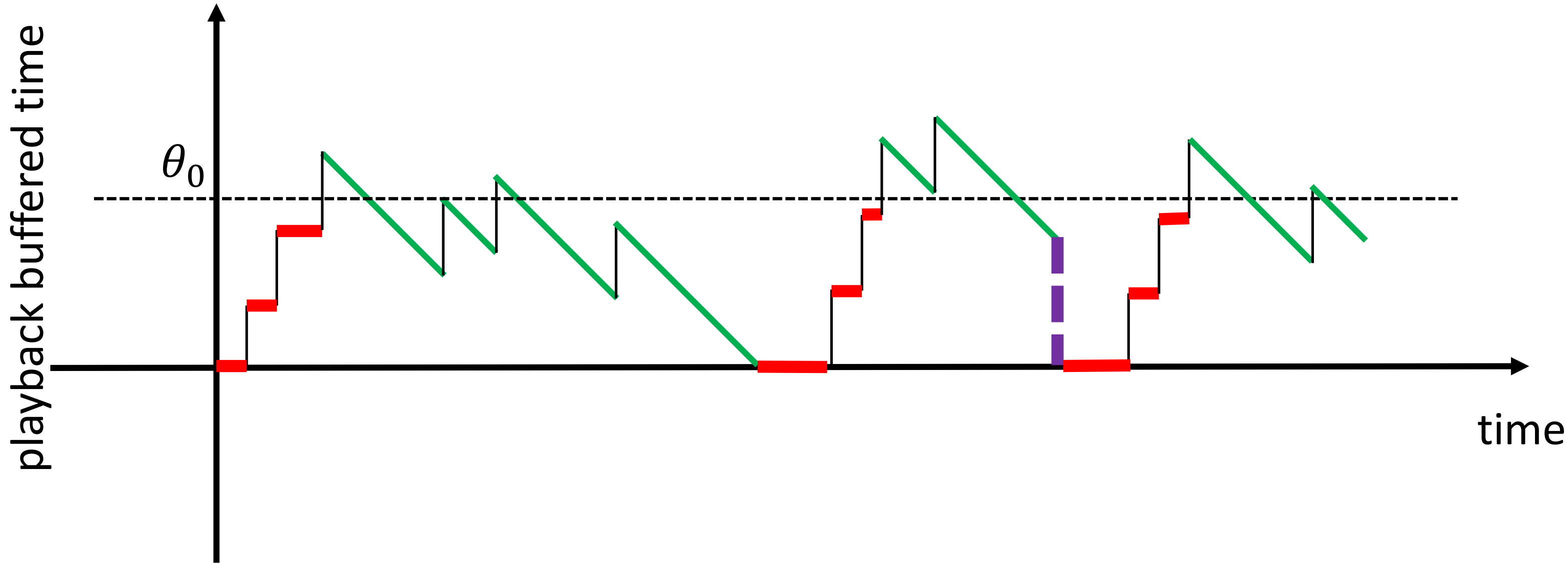
A complete description of the available chunks is provided into a Media Presentation Description (MPD) file exposed by the media server and consulted by each client



DASH - What is specified and what is not?



Buffer playback time evolution



chunk arrival at times $\boldsymbol{t}$

stall

video skip at times $\boldsymbol{s}$

Quality of Experience measures

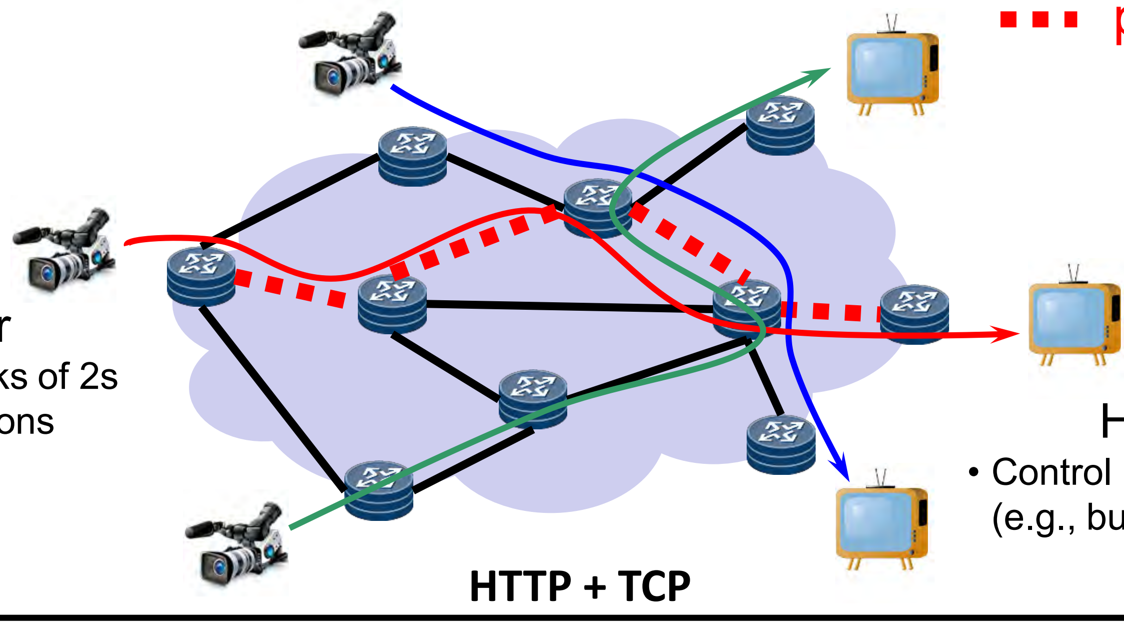
- **QoE factors (quantitative measurements)**
  - **Average video bitrate** of the downloaded chunks.
  - **Average bitrate variation:** the average of bitrate variations between consecutive chunks
  - **Re-buffering ratio:** freezing (or stalling) time over the duration of the video session.
- **QoE 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 chucks of 2s
- Multiple representations (qualities) of chunks

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

$$QoE_n\left(\underbrace{x_p}_{\text{Available bandwidth on p}}, \underbrace{\vec{\theta}_p}_{\text{QoS parameters of p}}, \underbrace{\vec{\theta}^n_{HAS}}_{\text{Client parameters of p}}\right) = f^n_{HAS}\left(\underbrace{g^n_{TCP}\left(x_p, \vec{\theta}_p\right)}_{\text{HTTP bandwidth}}, \vec{\theta}^n_{HAS}\right)$$

Available bandwidth on p

QoS parameters of p (e.g., delay, hops, loss)

Client parameters of p (e.g., HAS policy, content, resolution)

- 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

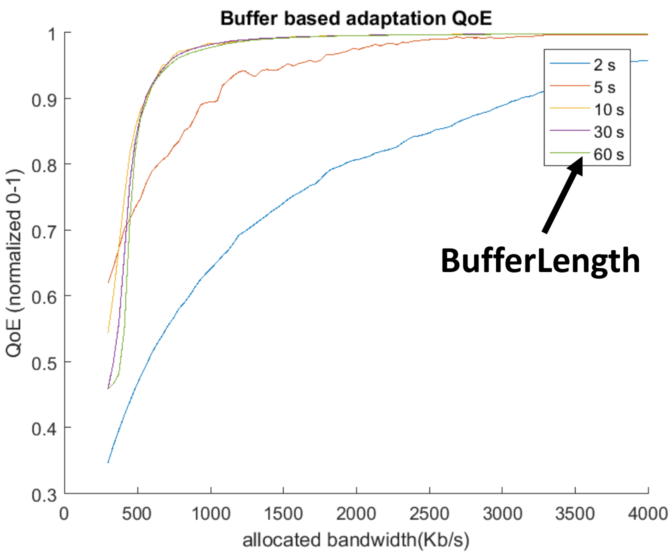
QoE Modeling – Linearization of $f_{HAS}^{n}(b,\theta_{HAS}^{n})$

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> ➔ θ_{HAS}^{n}
  - Available bandwidth ➔ b (bandwidth seen by HTTPS)
- Piecewise linearization of the measured $QoE_n = f_{HAS}^{n}(b,\theta_{HAS}^{n})$

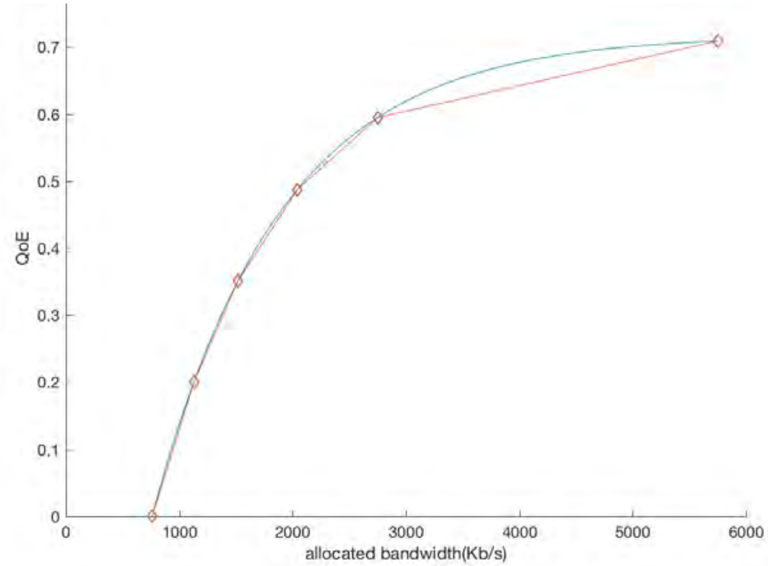
$$QoE_n = \frac{1}{1+\alpha\left(d'_{c_n} z_{c_n}^{b'_{c_1}} \underbrace{x_{c_n}^{b'_{c_0}}}_{\text{Compression}} + d'_{f_n} z_{f_n}^{b'_{f_1}} \underbrace{x_{f_n}^{b'_{f_0}}}_{\text{Freezing}}\right)^{\beta}}, \forall n \le N$$

f_{HAS}^{n}(b, <Buffer-Based-BufferLength) ,'Big Buck Bunny', 1080p>)



Linearization

f_{HAS}^{n}(b, <Buffer-Based-2s) ,'Big Buck Bunny', 1080p>)



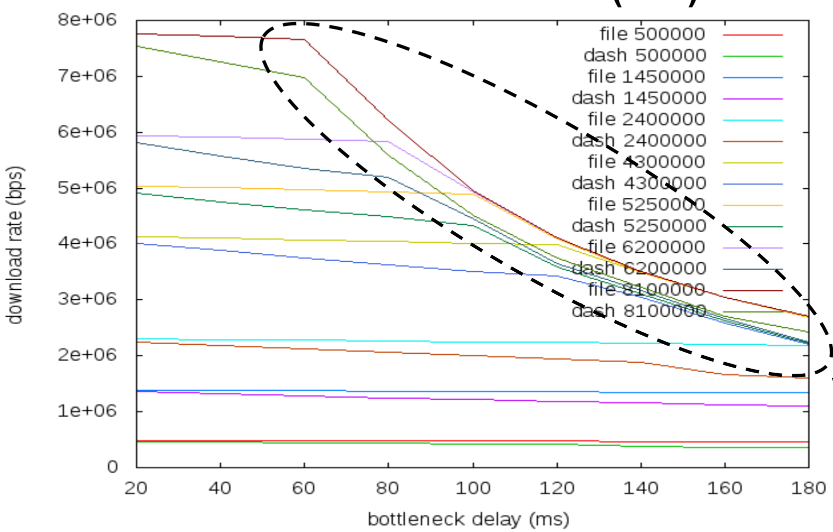
TCP–HAS Interplay

- 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

Congestion window very often decreases!

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

HAS with TCP cubic (ns3)



Round Trip Delay (path p)

$$\alpha_p = -1.279\cdot 10^{-3} + 1.0011\cdot \mathrm{RTD}_p$$

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

[1] J. Esteban, S. A. Benno, A. Beck, Y. Guo, V. Hilt, and I. Rimac, "Interactions between HTTP adaptive streaming and TCP," in ACM NOSSDAV, Jun. 2012.

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

Parameter	Meaning
$a_k^d \in \mathbb{R}_{\geq 0}$	Slope of the straight-line section $k \in K_d$ for the utility of demand $d \in D$
$b_k^d \in \mathbb{R}_{\geq 0}$	y-intercept of the straight-line section $k \in K_d$ for the utility of demand $d \in D$
$r_d^{\min} \in \mathbb{R}_{\geq 0}$	Required average bandwidth to get worst HAS representation allowed for demand $d \in D$
$r_d^{\max} \in \mathbb{R}_{\geq 0}$	Required average bandwidth to get best HAS representation allowed for demand $d \in D$
$c_e \in \mathbb{R}_{\geq 0}$	Capacity of link $e \in E$

Variable	Meaning
$u^d \in \mathbb{R}_{\geq 0}$	Utility value (QoE) of demand $d \in D$
$x_p^d \in \mathbb{R}_{\geq 0}$	Available bandwidth on path $p \in P$ to serve demand $d \in D$
$z_p^d \in \{0,1\}$	Whether path p is used to serve demand d
$z_{dpi} \in \{0,1\}$	Whether $i \in \mathbb{Z}_{\geq 0}$ competitive videos shares the bottleneck of path $p \in P$ used to serve demand $d \in D$
$n_e \in \mathbb{Z}_{\geq 0}$	Number of competitive videos sharing link $e \in E$

Max-HQR – Mathematical Formulation

Assumption:

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

$$x_p^d = \frac{c_e}{i} \Rightarrow U_{dpi} = \mathrm{U}_d\left(\alpha_p \cdot \frac{\min\{c_e : e \in E\}}{i}\right)$$

Number of competing flows

$$\max_{\{z,n\}} \sum_{d \in D} \sum_{p \in P_d} \sum_{i>0} U_{dpi} \cdot z_{dpi}$$

Utility (QoE) maximization

$$s.t. \quad \sum_{p \in P_d} z_p^d = 1 \qquad \forall d \in D$$

Single path

$$\sum_{d \in D} \sum_{p \in P_d} z_p^d = n_e \qquad \forall e \in E$$

Number of competing videos flows

$$z_p^d \cdot n_e \le \frac{c_e}{C} \sum_{i>0} i \cdot z_{dpi} \qquad \forall d \in D, \forall p \in P_d, \forall e \in E_p$$

$$\sum_{i>0} z_{dpi} = 1 \qquad \forall d \in D, \forall p \in P_d$$

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)

$$\max_{\{x,z,u\}} \sum_{d \in D} u^d$$

Utility (QoE) maximization

$$s.t. \quad \sum_{d \in D} \sum_{p \in P_d} x_p^d \le c_e \qquad \forall e \in E$$

Capacity constraint

$$a_k^d \left(\sum_{p \in P_d} \alpha_p \cdot x_p^d \right) + b_k^d \ge u^d \qquad \forall d \in D, k \in K_d$$

Rate allocation linearization

$$\sum_{p \in P_d} \alpha_p \cdot x_p^d \ge r_d^{\min} \cdot z_p^d \qquad \forall d \in D$$

$$\sum_{p \in P_d} \alpha_p \cdot x_p^d \le r_d^{\max} \cdot z_p^d \qquad \forall d \in D$$

Range of allocated bandwidth (between min and max bitrate)

$$\sum_{p \in P_d} z_p^d = 1 \qquad \forall d \in D$$

Single path

Max-QoE – Lagrangian Relaxation

$$\lambda_p = \sum_{e \in E} \lambda_e$$

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

$$s.t. \quad a_k^d \left(\sum_{p \in P_d} \alpha_p \cdot x_p^d \right) + b_k^d \geq u^d \qquad \forall d \in D, k \in K_d$$

$$\sum_{p \in P_d} \alpha_p \cdot x_p^d \geq r_d^{\min} \cdot z_p^d \qquad \forall d \in D$$

$$\sum_{p \in P_d} \alpha_p \cdot x_p^d \leq r_d^{\max} \cdot z_p^d \qquad \forall d \in D$$

$$\sum_{p \in P_d} z_p^d = 1 \qquad \forall d \in D$$

Coupling constraints have been eliminated ⇨ Subproblems can be solved independently

Max-QoE – Lagrangian Relaxation

Primal subproblem

$$\max_{\{x,z,u\}} \quad u^d - \sum_{p \in P_d} \lambda_p \cdot x_p^d$$

$$s.t. \quad a_k^d \left(\sum_{p \in P_d} \alpha_p \cdot x_p^d \right) + b_k^d \geq u^d \qquad k \in K_d$$

$$\sum_{p \in P_d} \alpha_p \cdot x_p^d \geq r_d^{\min} \cdot z_p^d$$

$$\sum_{p \in P_d} \alpha_p \cdot x_p^d \leq r_d^{\max} \cdot z_p^d$$

$$\sum_{p \in P_d} z_p^d = 1$$

Dual subproblem

$$\min_{\{\lambda,\pi,\mu,\upsilon\}} \quad \sum_{k \in K} \pi_k^d \cdot b_k^d - \mu_d \cdot r_d^{\min} + \upsilon_d \cdot r_d^{\max}$$

$$s.t. \quad \sum_{k \in K} \pi_k^d \geq 1$$

$$\frac{\lambda_p}{\alpha_p} \geq \left(\sum_{k \in K} \pi_k^d \cdot a_k^d - \mu_d + \upsilon_d \right) \qquad \forall p \in P_d$$

$$\pi_k^d \in \mathbb{R}_{\geq 0} \qquad \forall k \in K$$

$$\mu_d \in \mathbb{R}_{\geq 0}$$

$$\upsilon_d \in \mathbb{R}_{\geq 0}$$

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

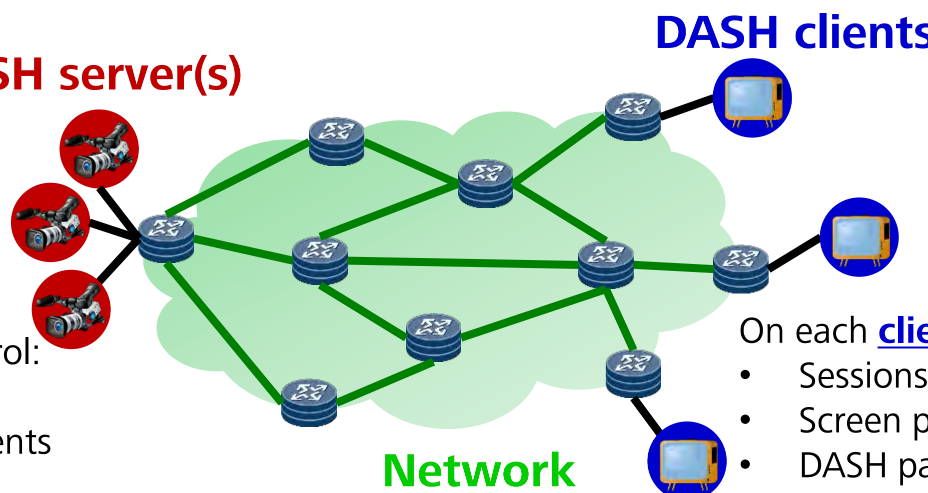
Dual Subgradient based on Lagrangian Relaxation (DGLR)

- Initialization:
  - $\mathrm{LB} = -\infty;\ \lambda_e = 0, \forall e \in E$
  - Compute K shortest paths for each demand $\forall d \in D$ (delay as link metric)
- While ($t \leq N_{max}$) 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 $r_p(t) = {}^{\lambda_p(t)}/_{\alpha_p(t)} = {}^{\sum_{e \epsilon p} \lambda_e(t)}/_{\alpha_p(t)}$
     - 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: $LB = \begin{cases} L^{(t)}(x, \lambda) & if L^{(t)}(x, \lambda) > LB \\ LB & if L^{(t)}(x, \lambda) \leq LB \end{cases}$
  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)



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

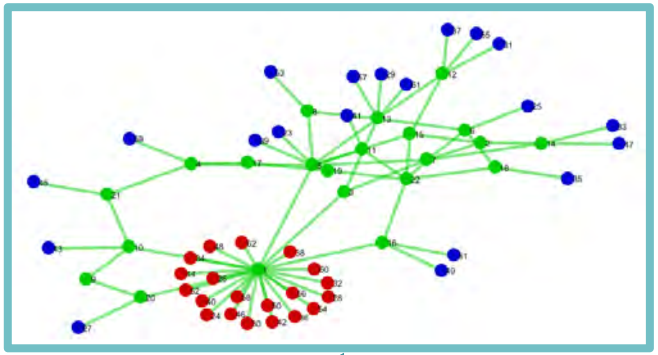
In the **network**, we can control:
- The topology and link properties (latency, jitter, loss rate, capacity)
- TCP parameters (e.g., congestion control algorithm, segment size)
- Drop policy (e.g., Red)
- …

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

Numerical Results

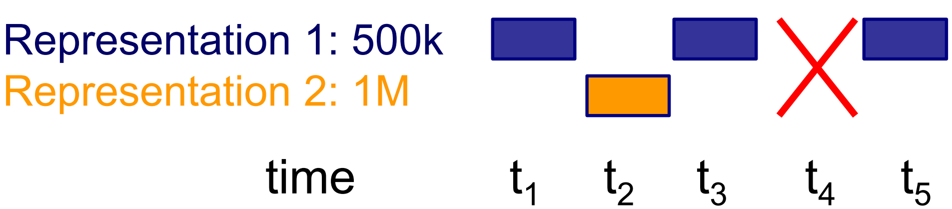


- **Scenario**
  - 3 representative movies:
    - Big Buck Bunny (BBB) – cartoon with a mix of low and high motion scenes
    - The Swiss Account (TSA) – sport video with high motion
    - Red Bull Play Street (RBPS) - sport documentary with regular motion
  - GEANT network (22 nodes and 36 links)
    - 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

[1] A. Juttner, B. Szviatovski, I. Mecs, and Z. Rajko, “Lagrange relaxation based method for the QoS routing problem,” in IEEE INFOCOM, 2001.

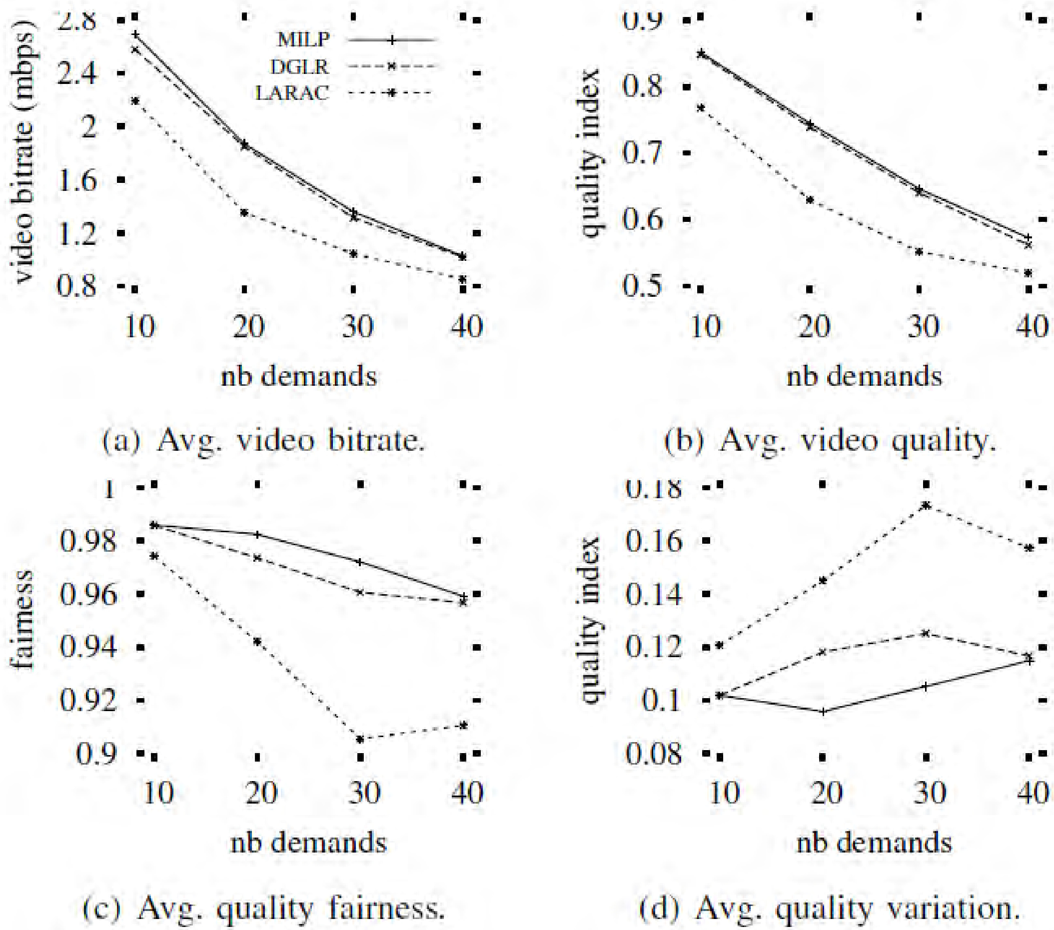
Numerical Results

- **Performance indicators**
  - **Average video bitrate** of the downloaded chunks. **(500k + 1M + 500k + 500k)/4**
  - **Average quality variation:** the standard deviation of the quality index which quantifies quality changes over the different downloaded chunks. **(500k + 500k)/2**
  - **Average video quality [0,1]:** Average on all downloaded chunks of a normalized quality index indicating to which representation they belong. **(0.5 + 1 + 0.5 + 0.5)/4**
  - **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. **20%**



Numerical Results

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



(a) Avg. video bitrate.

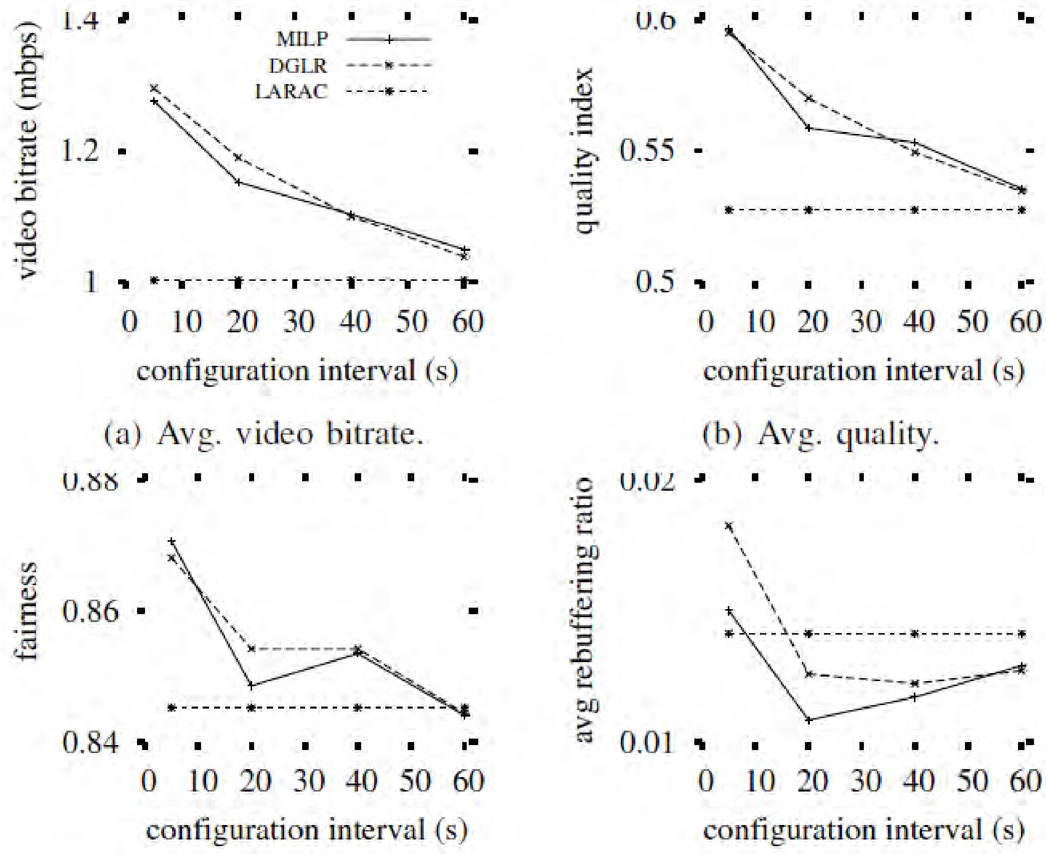
(b) Avg. video quality.

(c) Avg. quality fairness.

(d) Avg. quality variation.

Numerical Results

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



(a) Avg. video bitrate. (b) Avg. quality. (c) Avg. quality fairness. (d) Avg. re-buffering ratio.

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

$$QoE_n\left(\hat{x}_p, \overrightarrow{\theta_p}, \overrightarrow{\theta_{HAS}^n}\right)$$

- Bandwidth **x_p is fixed**
- We measure QoE_n
- We get linear coefficients of QoE_n

Optimization

$$\max_{\{x,z\}} \; QoE_n\left(x_p, \overrightarrow{\theta_p}, \overrightarrow{\theta_{HAS}^n}\right)$$
$$\text{s.t.} \quad \text{some constraints}$$

- We get the **optimal x^*_p**

Observation

$$QoE_n\left(x^*_p, \overrightarrow{\theta_p}, \overrightarrow{\theta_{HAS}^n}\right)$$

- We measure the real QoE_n
- **We want:**

$$\boxed{QoE_n\left(\hat{x}_p, \overrightarrow{\theta_p}, \overrightarrow{\theta_{HAS}^n}\right) = QoE_n\left(x^*_p, \overrightarrow{\theta_p}, \overrightarrow{\theta_{HAS}^n}\right)}$$

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?

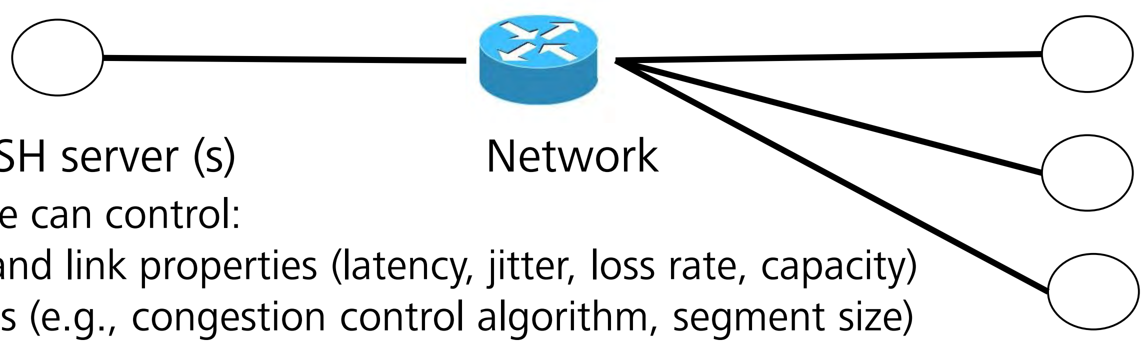
[1] H. Nam, K.-H. Kim and H. Schulzrinne. QoE Matters More Than QoS: Why People Stop Watching Cat Videos. In Proc. Infocom, 2016

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

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



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 caracteristics:** DASH policy, client resolution, max video buffer.

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

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

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

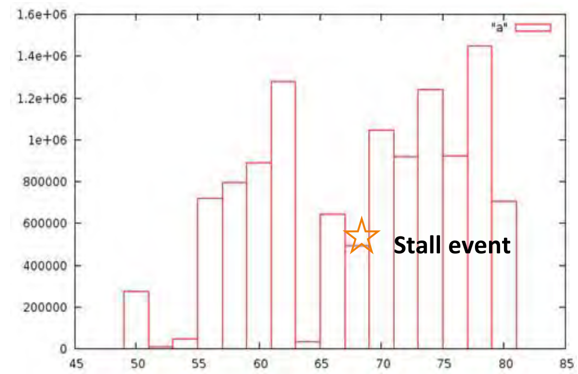
Well correlated with the network rate

The difficult one

Preliminaries: Data Characteristics

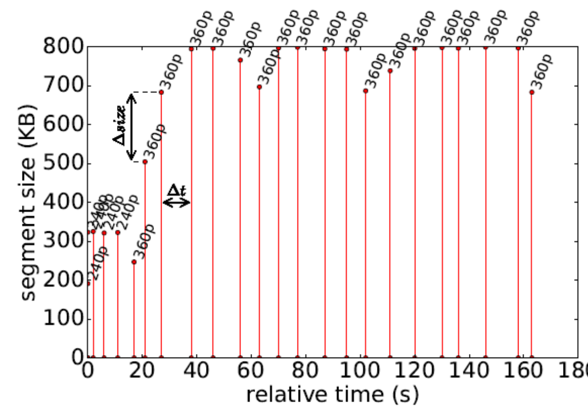
Index	Name	Type
2	NbClients	Simulation
3	BottleneckBW	Simulation
4	BottleneckDelay	Simulation
5	BottleneckLoss	Simulation
6	DASHPolicy	Simulation
7	ClientResolution	Simulation
8	RequestDuration	Input
9	TCPOutputPacket	Input
10	TCPOutputDelay	Input
11	TCPOutputJitter	Input
12	TCPOutputPloss	Input
13	TCPInputPacket	Input
14	TCPInputDelay	Input
15	TCPInputJitter	Input
16	TCPInputPloss	Input
17	TCPInputRetrans	Input
18	StdInputRate	Input
[19:27]	[0,5,10,25,50,75,90,95,100] InputRateVariation	Input
28	StdInterATimesReq	Input
[29:37]	[0,5,10,25,50,75,90,95,100] InterATimesReq	Input
38	StartUpDelay	Hidden
39	AvgDownloadRate	Hidden
40	StdDownloadRate	Hidden
41	AvgBufferLevel	Hidden
42	StdBufferLevel	Hidden
43	StallEvents	Hidden
44	RebufferingRatio	Target
45	StallLabel	Target
46	TotalStallingTime	Hidden
47	AvgTimeStallingEvents	Hidden
48	AvgQualityIndex	Hidden
49	AvgVideoBitRate	Target
50	AvgVideoQualityVariation	Target
51	AvgDownloadBitRate	Hidden

Network rate
Sampled over bins of 2s



Time

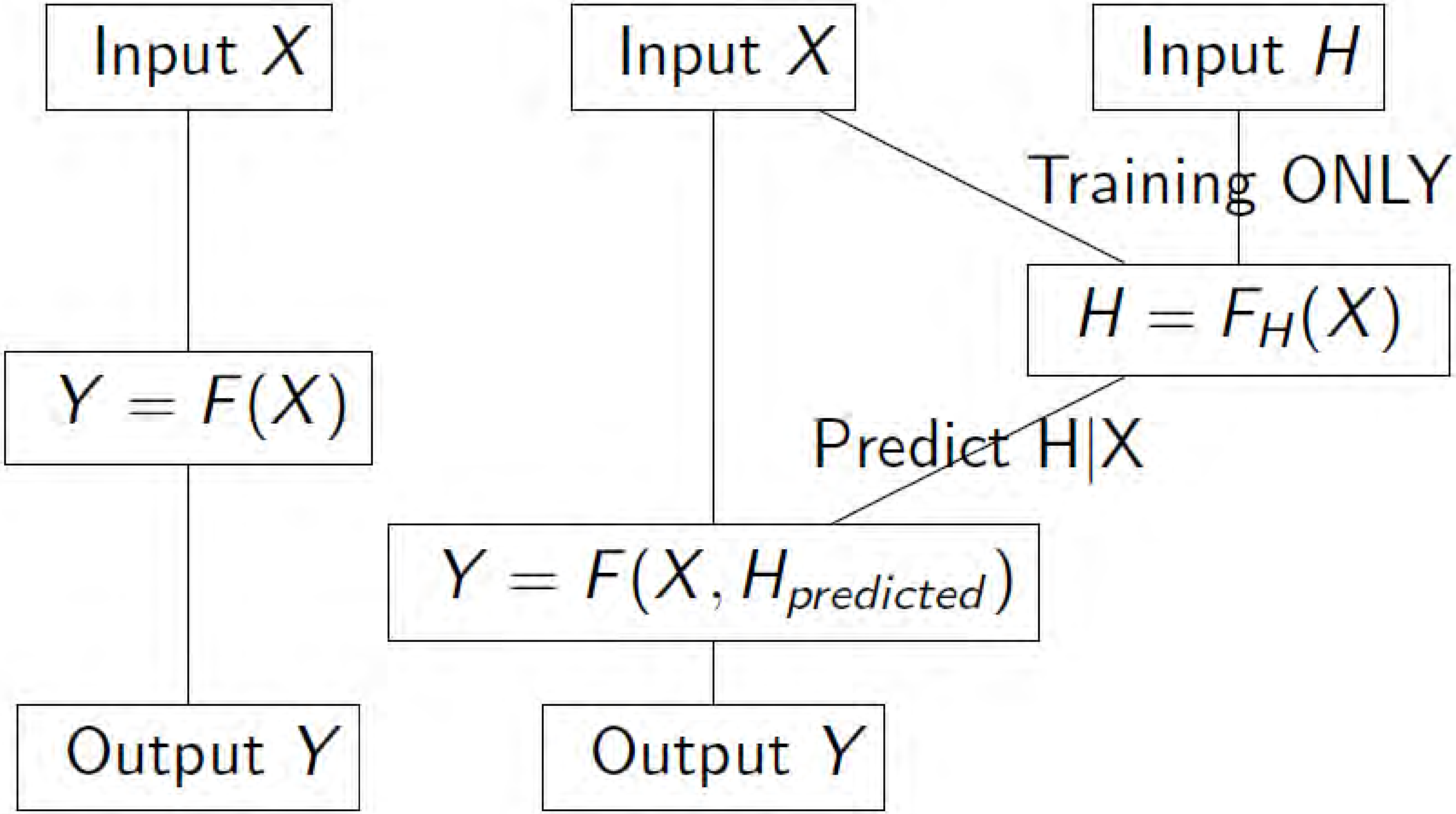
Inter-arrival times of HTTP requests
In mobile networks, all HTTP(s) requests go though HTTP proxies



- RebufferingRatio=0 → No staling
- 0 <RebufferingRatio< 0.1 → Mild staling
- RebufferingRatio≥ 0.1 → Severe staling

Preliminaries: Using Hidden Variables

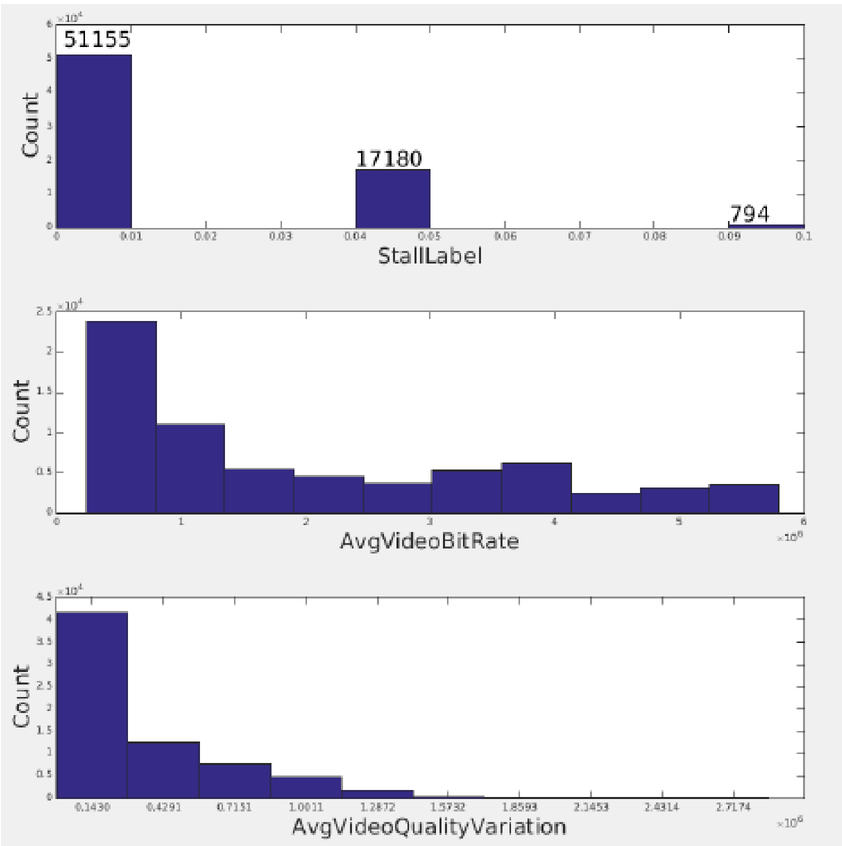
i.e. variables we may access only at training phase



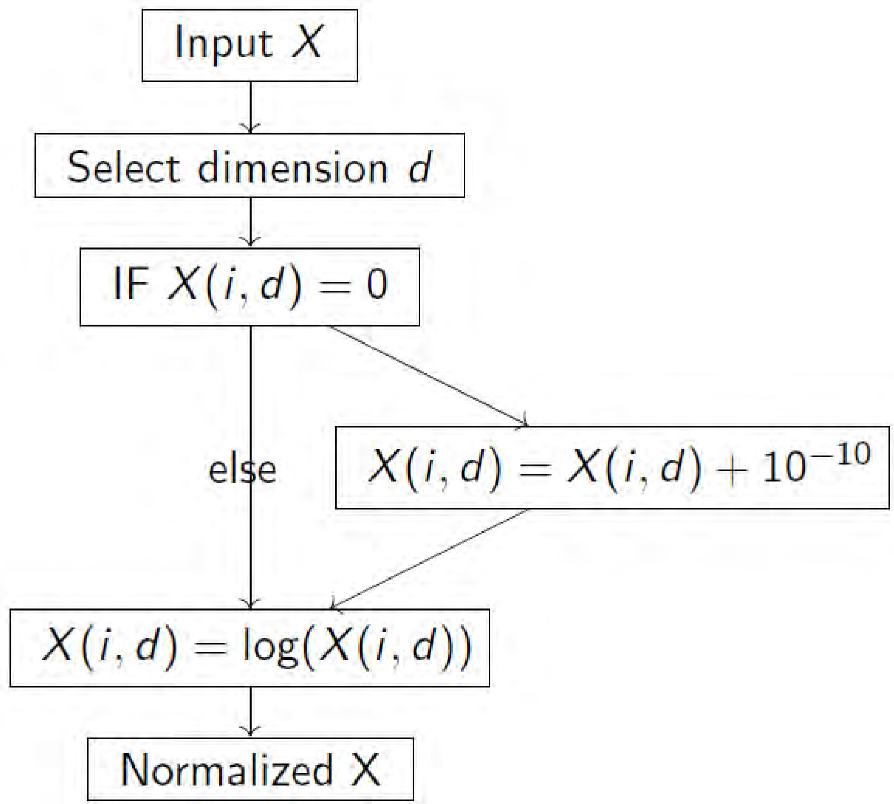
Bagged Regression Tree and Bagged Random Forest are used for the additional variable prediction based on whether they are continuous or discrete.

Preliminaries: Data Characteristics

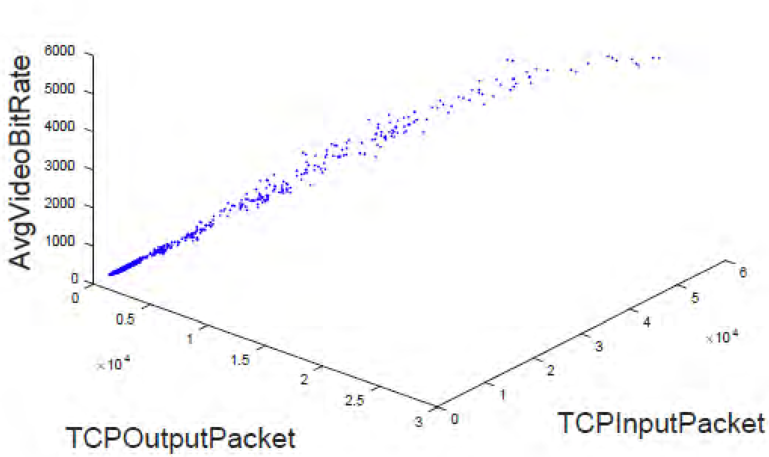
The distribution of the target variables is somewhat exponential.

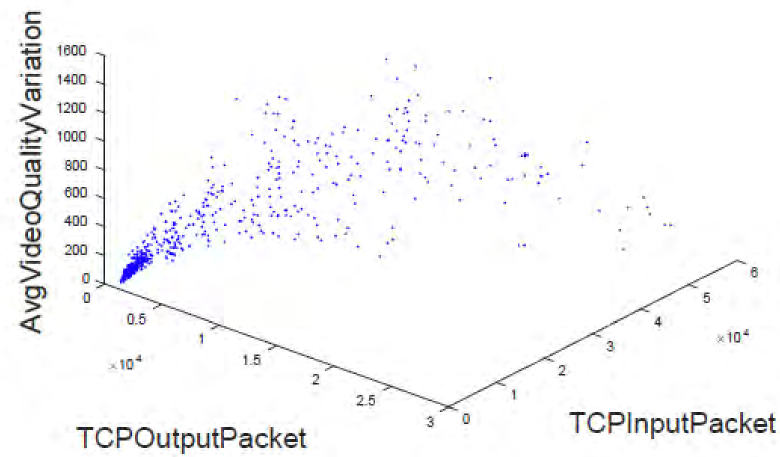


This exponential pattern occurred also for the input, hidden and simulation variables.



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)

RF	Tr. Acc.	Val. Acc.
0	0.96178	0.95525
0.05	0.7585	0.73587
0.1	0.43874	0.34211

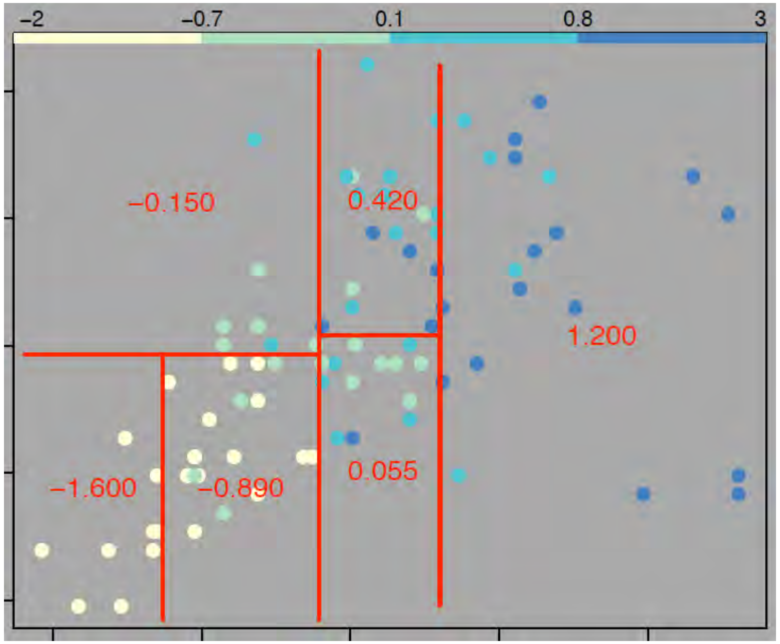
Random Forest (RF) pruned to minimum leaf size of 50.

The performance on the SevereStall class is practically unacceptable.

- 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 cause by the greedy split procedure.

[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.

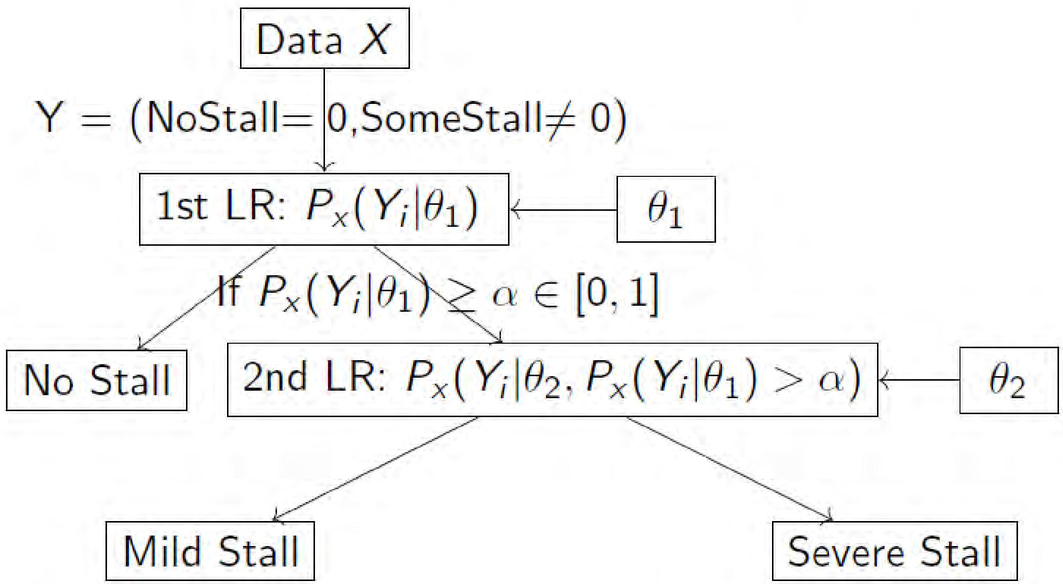
Confirming, that the RF decision regions are not adequate for our data.

- 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) classier we get an improvement on the RF.

[Bishop],[Ng]

StallLabel Prediction: BN definition

Define the BN in Fig. 2.1. If the two LRs are trained separately the performance is degraded. Also note that θ_1 and θ_2 are not d-separated [Koller] only when mild or severe stall is observed.



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

Figure 2.1: Bayesian Network using LR models.

BN	Tr. Acc.	Val. Acc.
0	0.8681	0.8684
0.05	0.7929	0.8048
0.1	0.9338	0.9368

Table 2.2: Performance of the proposed BN model

StallLabel Prediction: Results

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

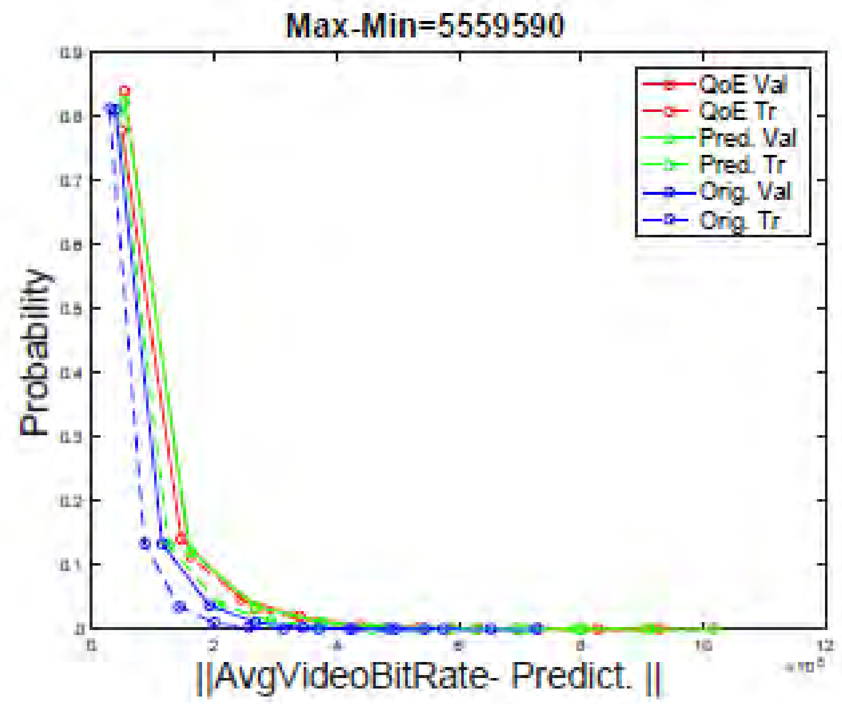
BN Stall Label. Using variables [38:42,48:51] from Tab. I.

VIDEO QUALITY

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

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.

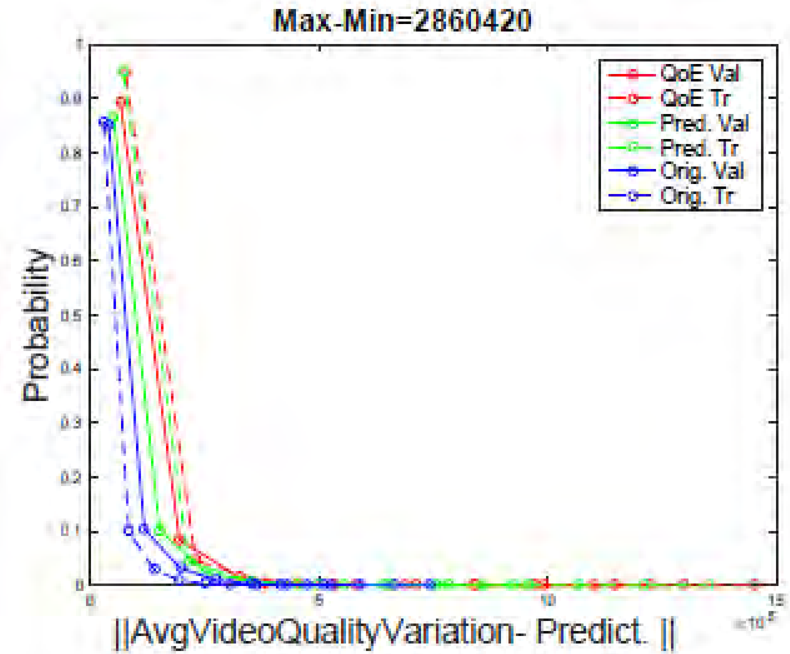


VIDEO QUALITY VARIATION

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

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

- C. M. Bishop, Pattern recognition and machine learning. Springer, 2006.
- D. Koller and N. Friedman, Probabilistic graphical models: principles and techniques. MIT press, 2009.
- V. Voloshin, Introduction to Graph and Hypergraph Theory, Nova Science Publishers, Inc.,2009
- G. Dimopoulos, I. Leontiadis, P. Barlet-Ros, K. Papagiannaki, Measuring Video QoE from Encrypted Traffic, ACM, 513–526, 2016
- G. Louppe, Understanding random forests: From theory to practice, arXiv:1407.7502, 2014
- Andrew Ng, CS229 Lecture notes,pp:1–3,2000
- V. Vasilev, G. Iliev, V. Poulkov A. Mihovska, A Latent Variable Clustering Method for Wireless Sensor Networks, Asilomar , 2016
- V. Vasilev, Chromatic Polynomial Heuristics for Connectivity Prediction in Wireless Sensor Networks, ICEST 2016, Ohrid, Macedonia, 28-30 June, 2016
- V. Vasilev, Algorithms and Heuristics for Data Mining in Sensor Networks, LAP LAMBERT Academic Publishing, 22 December 2016, ISBN:987-3330025219

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.* .
- **Predicting QoE Factors with Machine Learning.** Vladislav Vasilev, Jeremie Leguay, Stefano Paris, Lorenzo Maggi, Merouane Debbah. *IEEE ICC 2018.*
- **Overlay Routing for Fast Video Transfers in CDN**. Paolo Medagliani, Stefano Paris, Jérémie Leguay, Lorenzo Maggi, Xue Chuangsong, Haojun Zhou. IEEE IM 2017.
- **Scalable Request Routing for VR-ready CDNs.** Pierre-Louis Poirion, Jérémie Leguay, Ruosi Liu. ICIN 2018.
- **Adapting Caching to Audience Retention Rate: Which Video Chunk to Store?** Lorenzo Maggi, Lazaros Gkatzikis, Georgios Paschos, Jeremie Leguay. Elsevier Computer Communications. January 2018.