



FoG and Clouds: Optimizing QoE for YouTube

Tobias Hoßfeld^{1,3}, Florian Liers², Thomas Volkert², Raimund Schatz³

¹University of Würzburg

²Technical University of Ilmenau

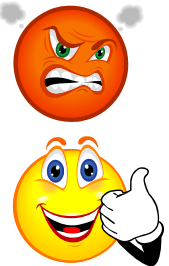
³Telecommunications Research Center Vienna FTW

Trend towards Quality of Experience

- ▶ Increasing competition among Telco's and ISPs, among application and service providers, among cloud providers
- ▶ Keep customers happy, attract new customers
- ➔ **Quality** as key differentiator, but only as **experienced by end user**

- ▶ Shift from Quality of service (QoS) to Quality of Experience (QoE)
 - QoS: packet loss, delay, jitter, ...
 - QoE: subjective experience/satisfaction of users of a service
- ▶ *Example:* web user interested in short page load times
VoIP user interested in speech quality
video user interested in video quality and smooth playout w/o interruptions

- ▶ What are key **QoE influence factors** and appropriate **QoE models**?
- ▶ How to monitor QoE? How to **optimize QoE**?

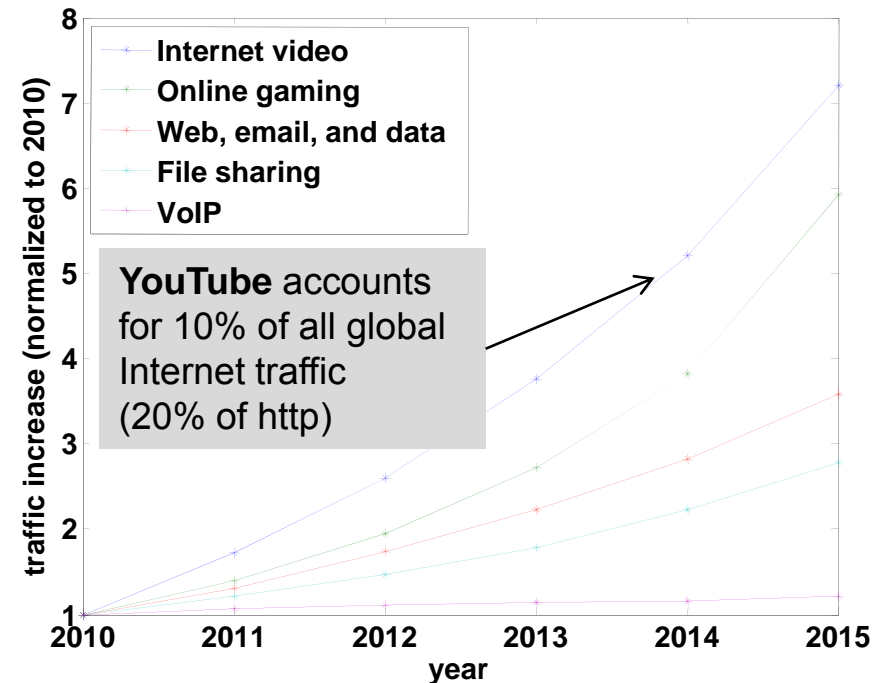


Observations

- ▶ Importance of Internet video streaming is increasing
 - Related work on video QoE mainly covers video quality degradation, e.g. artifacts
 - Missing **QoE model** wrt. **stalling** instead of quality degradation

- ▶ Application and network are not communicating

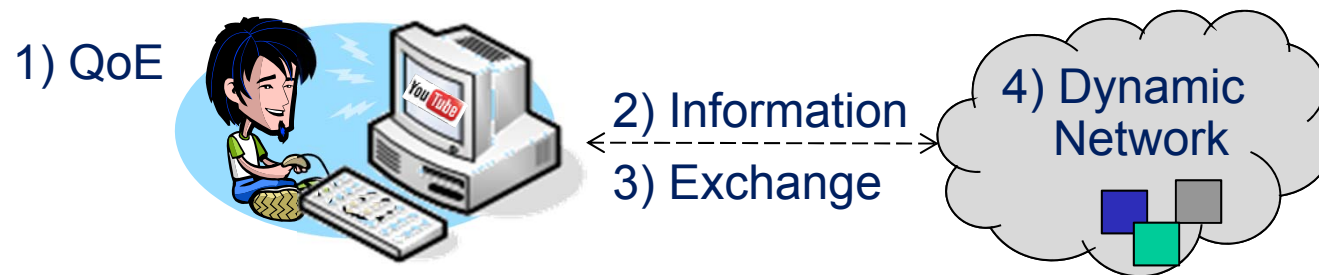
- **missing information** for optimized transmission
- missing interfaces and languages for network description and QoE/application requirements



Source: Cisco Visual Networking Index: Forecast and Methodology, 2010-2015

Problem Formulation

- 1) How to model YouTube Quality of Experience?
- 2) What are requirements for optimal video transmission with respect to QoE and network costs?
- 3) How to exchange information between network and application?
- 4) How to implement dynamic network stack based on requirements?



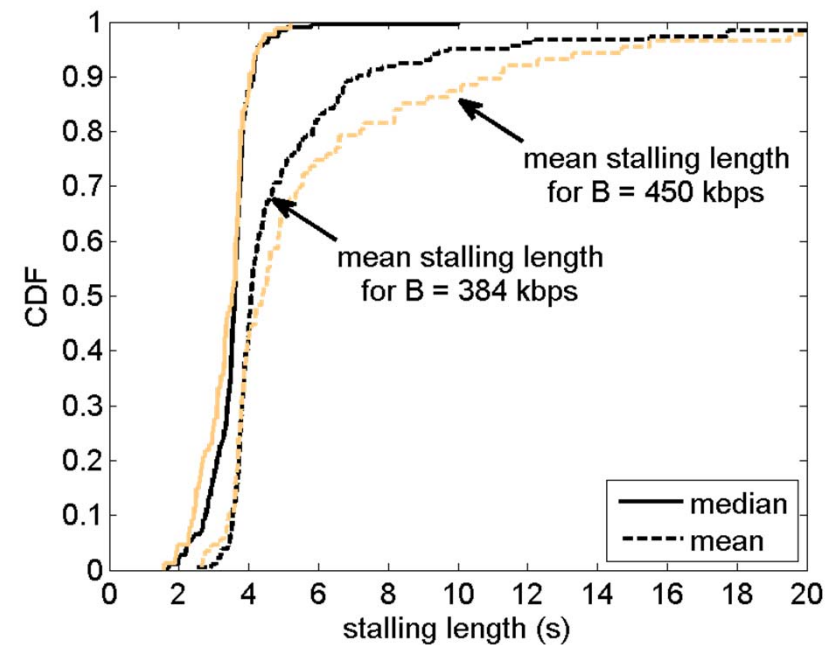
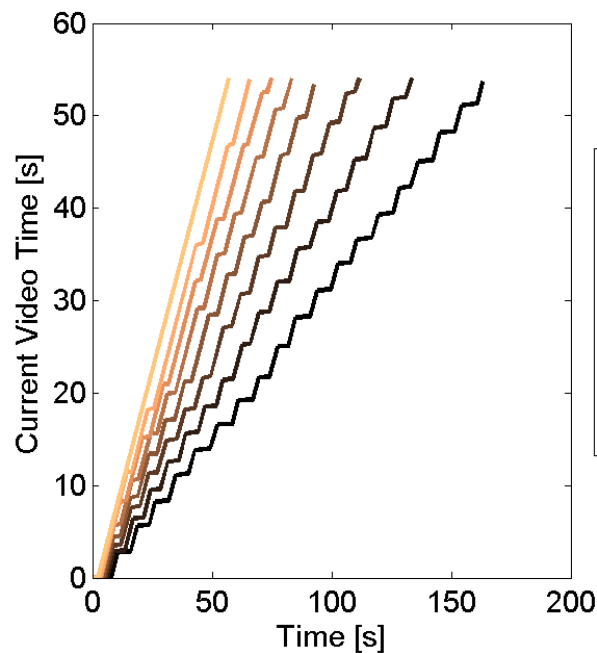
QoE Issue: Waiting, Waiting, Waiting...

You Tube



Stalling Pattern in case of Bottleneck Scenario

- ▶ A bottleneck link throttles the download capacity to a fixed bandwidth
- ▶ Stalling patterns of YouTube with bottleneck link scenario
- ▶ Large measurement campaign: Capturing on application layer



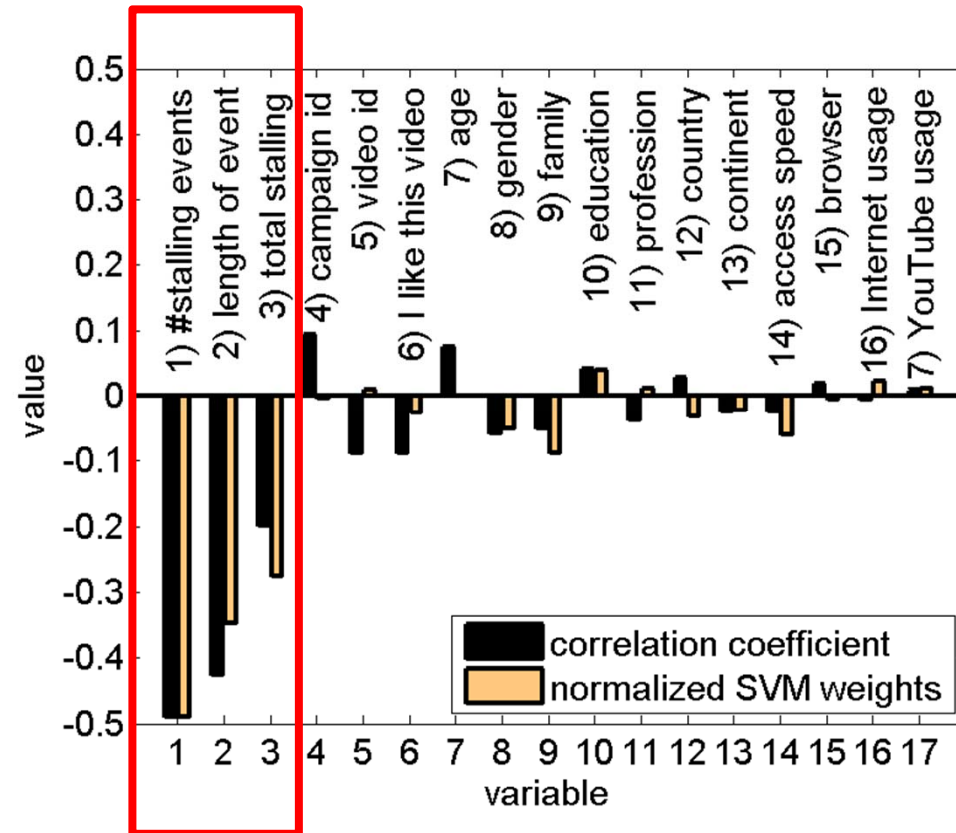
- ➔ **Stalling occurs at regular intervals of equal length**
- ➔ **Input parameter for subjective user studies**

Key Influence Factors on YouTube QoE

- ▶ Statistical analysis and sophisticated statistical tools, like support vector machines used to derive key influence factors on QoE

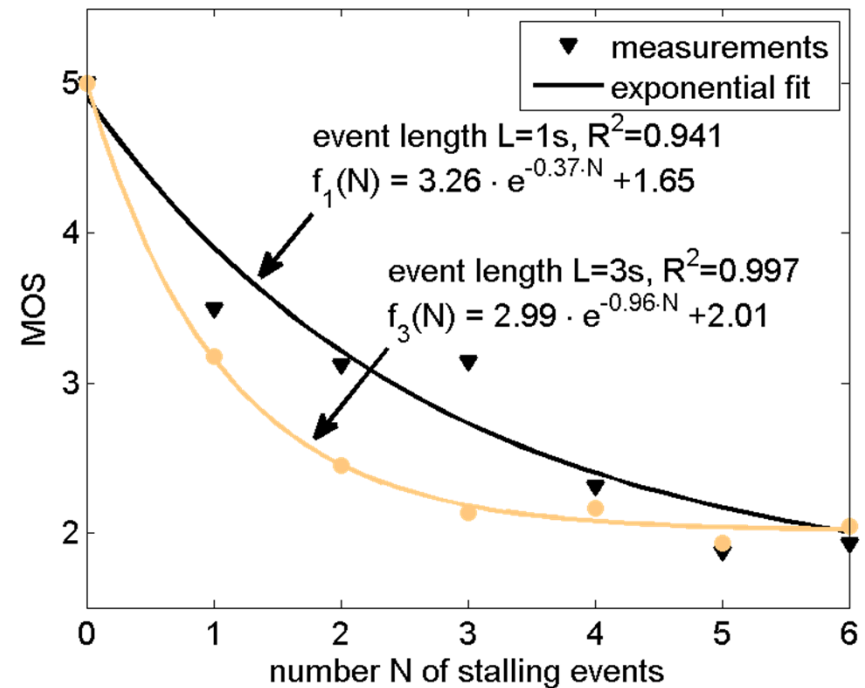
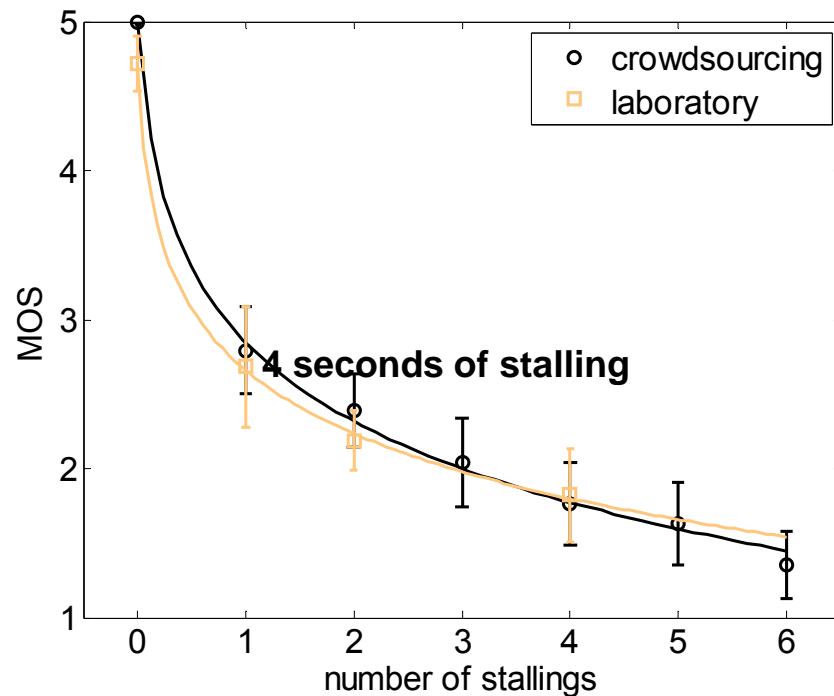
- ▶ Interesting: no correlation of QoE and
 - video characteristics like resolution, type of content ratio of audio/video, etc.
 - users preference, whether they liked video
 - demographical features

- ▶ **Stalling frequency and stalling duration determine the user perceived quality**



YouTube QoE Model

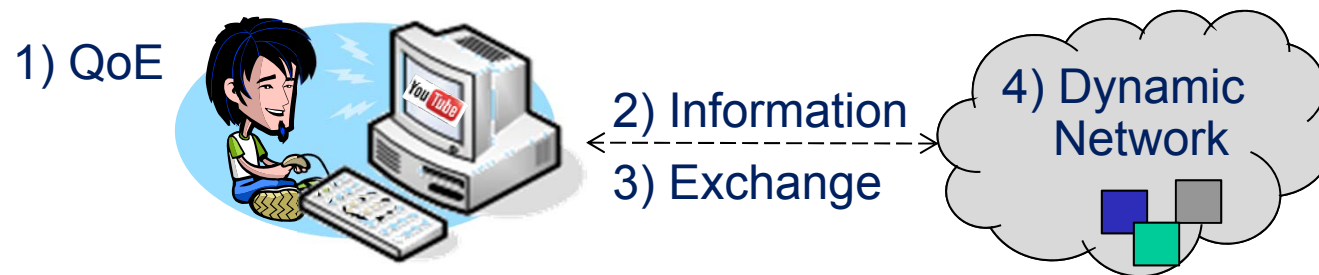
- ▶ Lab studies with UniWue's CrowdSourcing and at FTW's i:Lab
- ▶ Mapping functions between MOS and stalling pattern, i.e. event length and the number of stalling events, are provided



➔ Users only accept almost no stalling or only short stalling

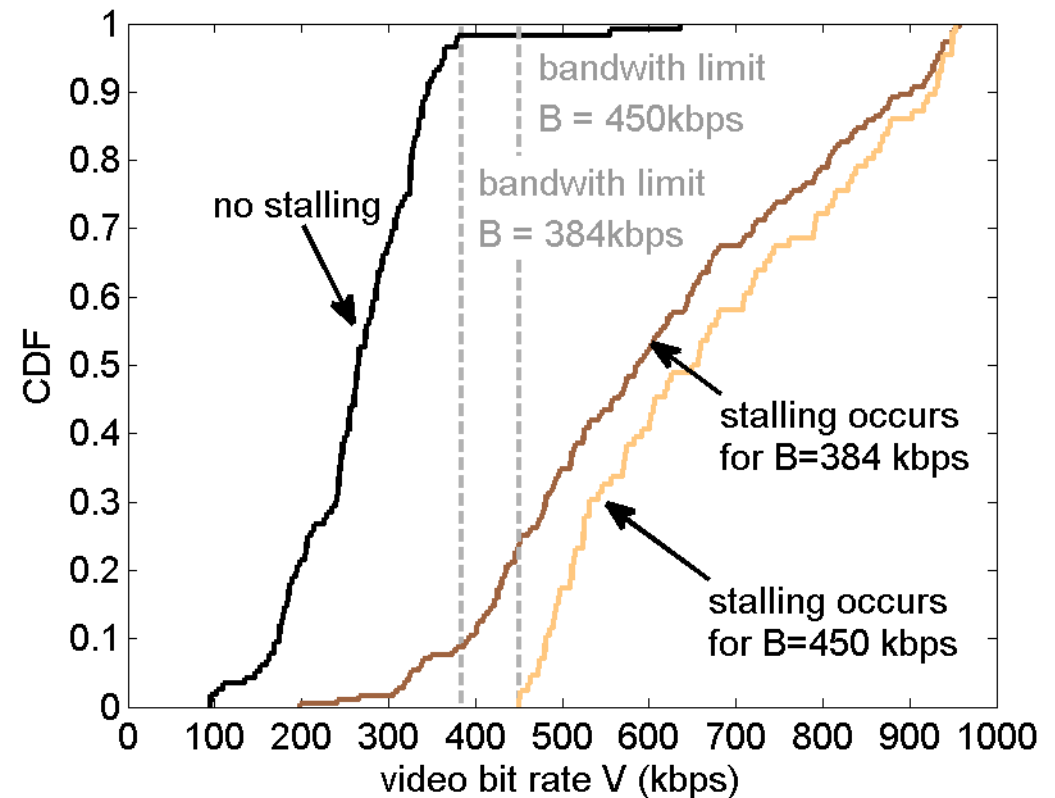
Problem Formulation

- 1) How to model YouTube Quality of Experience?
- 2) What are requirements for optimal video transmission with respect to QoE and network costs?
- 3) How to exchange information between network and application?
- 4) How to implement dynamic network stack based on requirements?



Video Bit Rate as only Information for Transmission

- ▶ No stalling although the video bit rate $V >$ network bandwidth B
→ reason: initially buffered video data (and actual video duration)
- ▶ Stalling occurs, if $V > B$
- ▶ Stalling sometimes occurs, if $V < B$ → reason is variability of video codec
- ▶ Consequences for
 - monitoring
 - QoS provisioning and resource management
 - Dimensioning



→ **Transmission with video bit rate not sufficient to avoid stalling**

Initial Delay

- ▶ Avoid stalling by increasing
 - network bandwidth
→ at cost of network provider

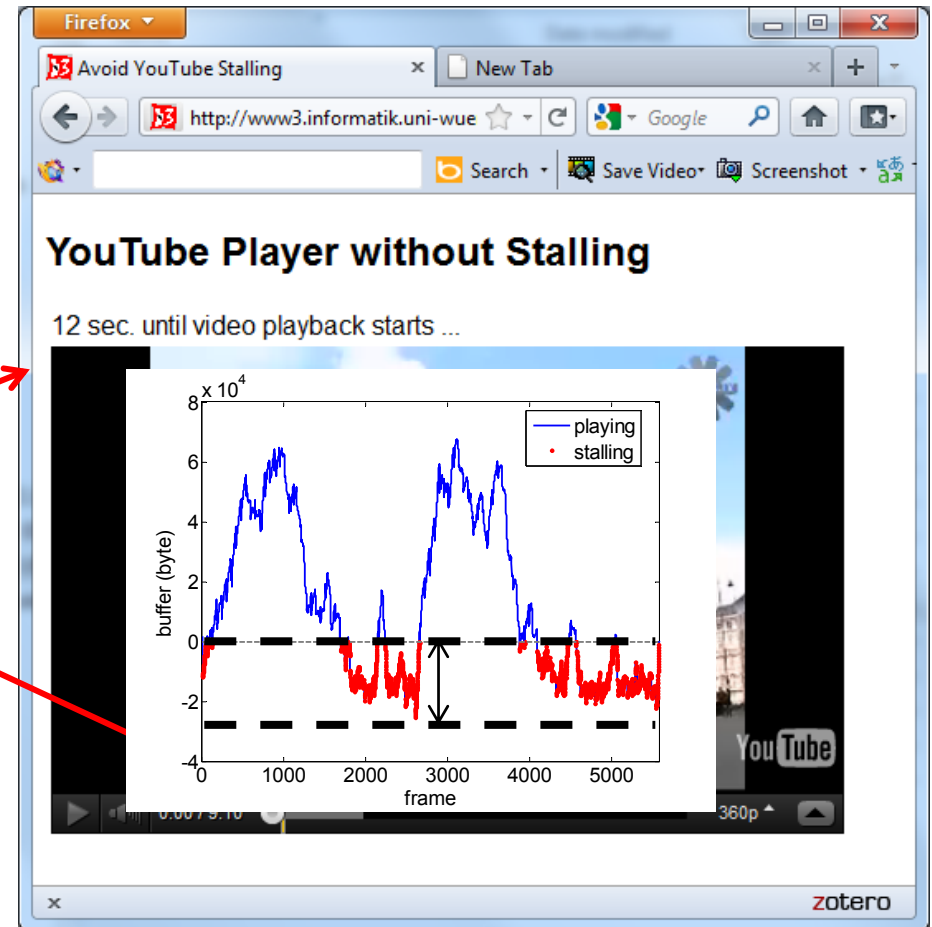
OR

- initial delay →
at cost of customer's
experience **initial delay
announced to user**

**initial delay allows
buffering of video**

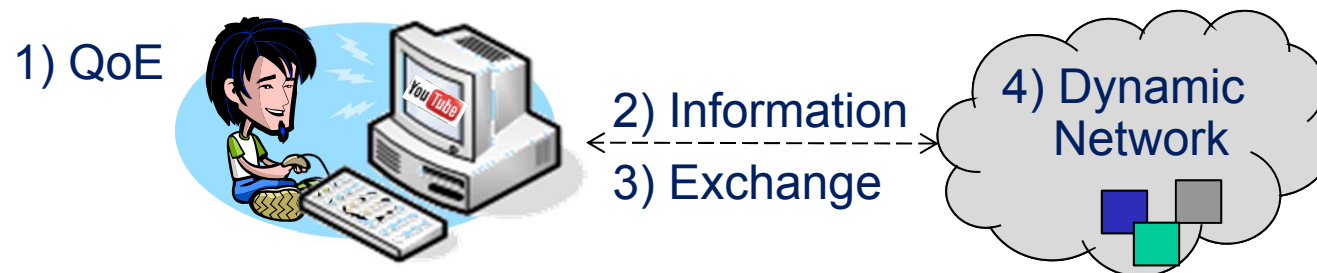
- ➔ For given video and bandwidth B ,
initial delay can be computed
to avoid stalling

- ▶ Implementation easy for embedded videos, see
<http://www3.informatik.uni-wuerzburg.de/staff/hossfeld/yt/hosstube.shtml?s=15>



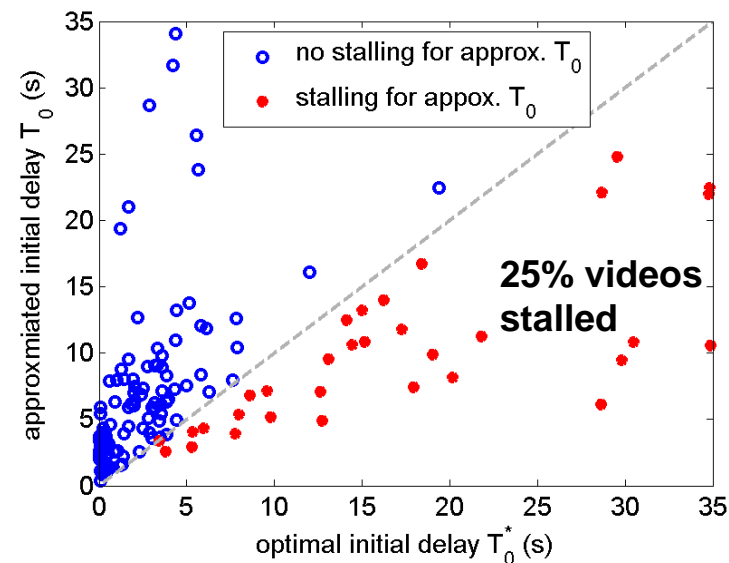
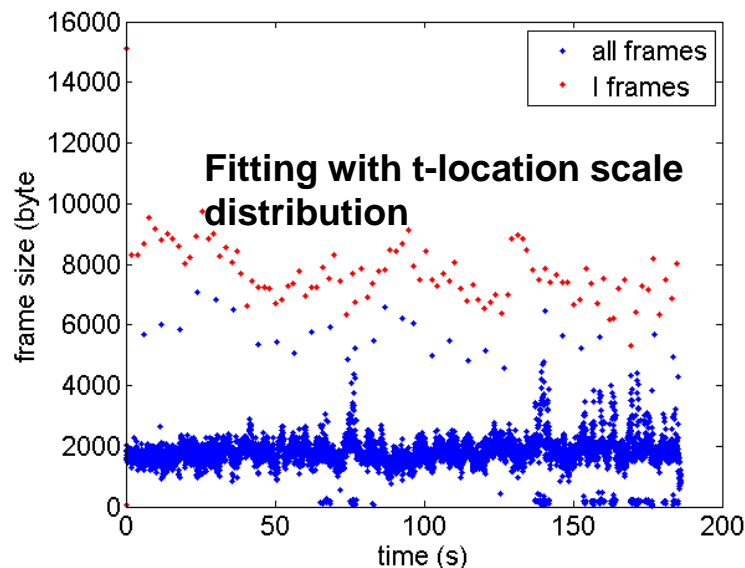
Problem Formulation

- 1) How to model YouTube Quality of Experience?
- 2) What are requirements for optimal video transmission with respect to QoE and network costs?
- 3) How to exchange information between network and application?
- 4) How to implement dynamic network stack based on requirements?



Exchange of Information: Approximation of Video Characteristics

- ▶ Initial delay depends on individual video and available bandwidth
 - Optimal initial delay requires size for each frame → overhead
 - Sizes of key and inter-frames described statistically → overhead--
 - Approximation of initial delay with simple (or more sophisticated) equations $B \sim \text{Norm}(0, N_G (\sigma_k^2 + N_i \sigma_i^2))$

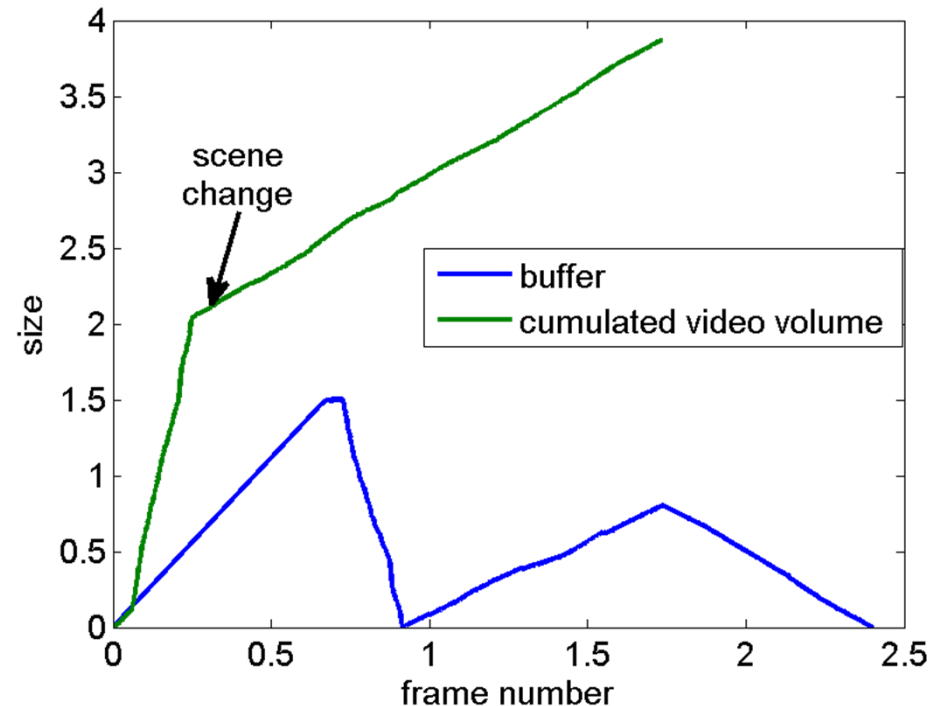


➔ Approximation works only for 3/4 of the considered videos

Scene changes have to be considered!

- ▶ Scene changes may lead to significant changes of video bitrate

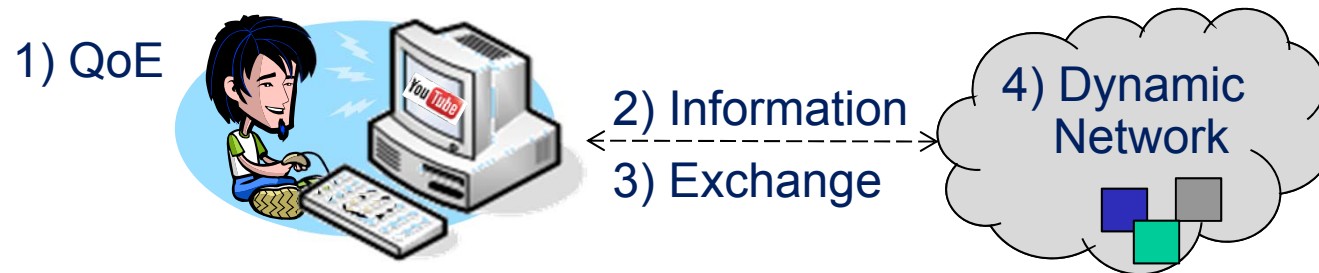
- ▶ Options for improved approximation
 - Statistical description of frame sizes per scene
 - Complex frame size models taking into account correlations across scenes



- ▶ But: YouTube videos are short!
 - ➔ **Specify sizes of all frames in meta information of video file**
 - ➔ **Define Application-Network Interface**

Problem Formulation

- 1) How to model YouTube Quality of Experience?
- 2) What are requirements for optimal video transmission with respect to QoE and network costs?
- 3) How to exchange information between network and application?
- 4) How to implement dynamic network stack based on requirements?

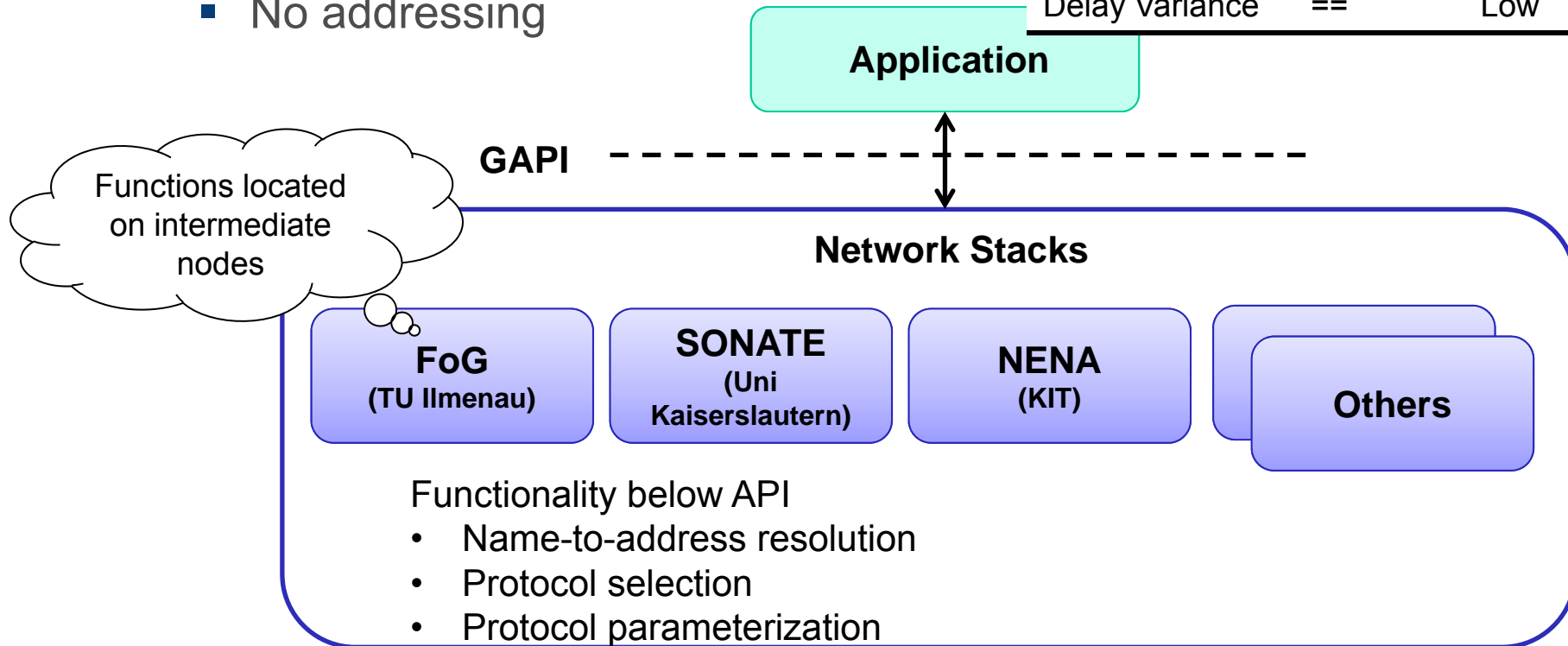


GAPI: G-Lab App-to-Network Interface



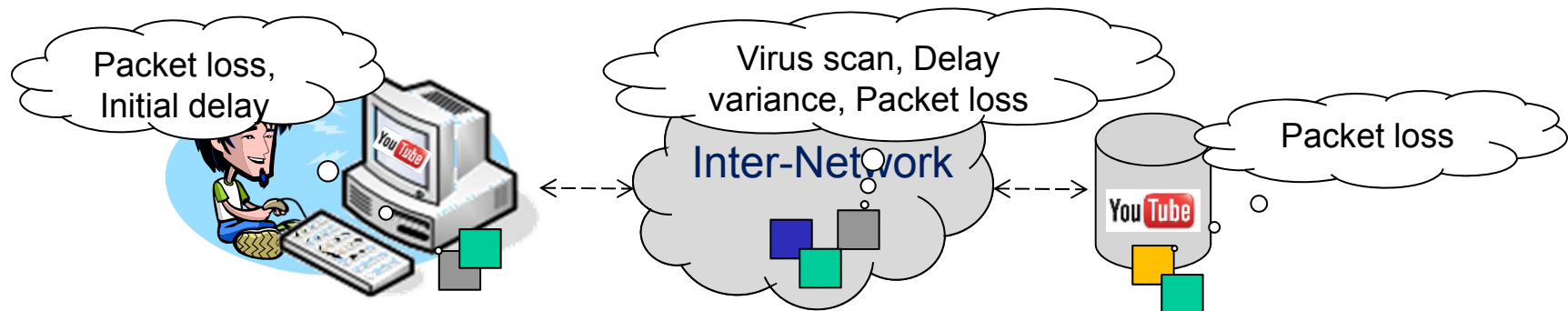
- ▶ Move network knowledge from app to network
- ▶ GAPI: Requirement-based API
 - No protocol knowledge
 - No addressing

Effect	Operator	Attribute
Packet Loss	<	5%
Virus Scan	==	True
Initial Delay	<	10s
Delay Variance	==	Low



Forwarding on Gates (FoG)

- ▶ Dynamic composition of functional blocks (called “Gates”)
 - Based on application requirements
 - End-to-end and on intermediate nodes
 - Reuse of existing gates for improved scalability
- ▶ Routing
 - Bases on gates and not on network structure
 - Supports functional and non-functional requirements
- ▶ Transfer independent from naming/addressing
 - Routing is free to choose any addressing scheme



FoG Demo Screenshot

The screenshot displays the Eclipse IDE interface for a simulation project named 'bus_bc(2)'. The main workspace is divided into several panes:

- Simulation:** Shows the overall simulation status and a list of running simulations: AS:as2 and AS:as1.
- AS:as1:** A network diagram showing nodes A, bus_ab(0), B, and bus_bc(2) connected by links.
- bus_bc(2):** A configuration panel for the bus component with the following settings:
 - Loss probability: 10 %
 - Bit error probability: 10 %
 - Link delay: 5 msec
- AS:as2:** A network diagram showing nodes C, bus_cd(1), and D connected by links.
- Properties:** A table showing the properties of the selected component (AS:as1):

Property	Value
AS	
Capabilities	
Gates	n.a.
Identity	B={Simple}
Is broken	false
Lower layer	2
Name	B
Registered servers	
Routing service	
Running apps	
- Event handler status:** Shows a diagram of the event handler for the selected component, including a '51 NumberingGate (11629)' and a '50 OrderAndCheckGate (39099)'. A green box highlights the function 'FN(host//_6)@B'.
- Frogger Video Viewer:** Two video windows showing a 3D rendered scene of a grey rabbit-like creature in a field. The left window is labeled 'best effort' and the right window is labeled 'QoS'.
- Console:** Shows the selected logger 'B' and a list of log messages:

Nu...	Packet	Object
94328	(((23,47,17,10,12,13],[41,42,43]))[[21,6,7,1,28,54]], 'NumberingHeader(no=11615, data=[B@19fca3b)' (A,B)	FN(host//_6)@B
94328	(((23,47,17,10,12,13],[41,42,43]))[[21,6,7,1,28,54]], 'NumberingHeader(no=11615, data=[B@19fca3b)' (A,B)	FN(host//B)@B
94328	(((23,47,17,10,12,13],[41,42,43]))[[21,6,7,1,28,54]], 'NumberingHeader(no=11615, data=[B@19fca3b)' (A,B)	FN(host//_2)@B
94328	(((23,47,17,10,12,13],[41,42,43]))[[21,6,7,1,28,54]], 'NumberingHeader(no=11615, data=[B@19fca3b)' (A,B)	LowerLayerReceiveGate@B

Conclusions

- 1) YouTube QoE model is available.
- 2) Trade-off between user's QoE and ISP's costs adjustable by means of initial delays.
- 3) Additional video information has to be specified
 - to monitor QoE within the network (*or monitor at end user*),
 - to optimally dynamically allocate network resources (*or feedback information from end user*);
 - per frame information is recommended in practice.
- 4) Implementation of GAPI and FOG: work in progress.
 - Clouds and open interfaces may be utilized
 - New cloud applications and cloud QoE

