Computer Networks 000 (2016) 1–11

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

TCP video streaming and mobile networks: Not a love story, but better with context

Florian Metzger[a,*], Eirini Liotou[b], Christian Moldovan[a], Tobias Hoßfeld[a]

[a] Chair of Modelling of Adaptive Systems, University of Duisburg-Essen, Germany
[b] Department of Informatics & Telecommunications, National & Kapodistrian University of Athens, Greece

A R T I C L E I N F O

Article history:
Received 22 October 2015
Revised 23 June 2016
Accepted 25 June 2016
Available online xxx

Keywords:
Mobile networks
TCP streaming
Context factors

A B S T R A C T

The rise in popularity of TCP-based video streaming in recent years is unbroken. These streaming services not just operate on wired access lines but more and more specifically target users of mobile networks as well. Yet it still remains difficult to evaluate the performance of such streaming approaches in mobile networks. This is especially critical as mobile networks exhibit much more potential for undesirable interactions between the network protocol layers and control plane properties on one side and the protocols and strategies of the application layer on the other side, ultimately resulting in scenarios with bad QoE for video streaming.

This paper aims to rectify this lack of knowledge and understanding in a multi-pronged approach as follows: The first contribution provides an easy means to investigate such interactions with a streaming simulation framework built on ns-3. In a second contribution three exemplary scenarios within this framework are investigated in order to uncover the nature of such interactions. The final undertaking attempts to unravel the issues of the mobility scenario using context information. Such information can be collected through crowdsensing and collaboratively processed in a Big Data approach. This results in a tailor-made analytical solution through the formulation of a Mixed Integer Linear Program (MILP) that can prevent video stalling in this particular scenario.

© 2016 Elsevier B.V. All rights reserved.

1. Introduction

While video streaming has always been a hot topic in computer networks, its prevalent form today, TCP-based streaming, has only shifted into focus in recent years. TCP-based streaming solutions exhibit a number of properties not present in the classic rtp-based approaches. These attributes helped make video streaming become popular in the Internet. As most variations of TCP streaming rest comfortably on top of HTTP they were an ideal fit for use in Web video services. But their differences in behavior also made them a worthwhile target for research focused on determining and optimizing their Quality of Service (QoS) and Quality of Experience (QoE), as has been conducted in many publications to date. A further recent development concerns the increasing importance of mobile networks on the last mile, with many people now using their mobile phones or tablets as their primary entry point to the Internet. Video streaming, with its high demands on network traffic, also has become prevalent on mobile devices.

But the transition from existing broadband access TCP streaming services to mobile network video streaming is not straightforward. It is immediately evident that conditions in the radio access are naturally different from a wired connection and thus can potentially be factored into designing such services. However, when looking past the obvious, some other effects pop up that are much harder to pin down. Most of these effects have to do with the fact that a mobile device is not just simply connected via a radio link to one base station, rather there is a whole mobile network behind it, with numerous specialized network nodes and functions managed through a very pronounced control plane with countless signaling procedures (cf., e.g., [1]).

All of these components and mechanisms can interact with end-to-end applications, such as TCP streaming, in unforeseen ways. Consider for example in a mobility scenario the brief service outage during a handover between two base stations, an effect that might occur quite frequently at higher velocities. Combined with a fading radio signal the throughput can vary greatly in short time scales, with the addition of high delay variations. Such kinds of "*interactions*" may also occur between other features of the lower

* Corresponding author.
E-mail addresses: florian.metzger@uni-due.de (F. Metzger), eliotou@di.uoa.gr (E. Liotou), christian.moldovan@uni-due.de (C. Moldovan), tobias.hossfeld@uni-due.de (T. Hoßfeld).

http://dx.doi.org/10.1016/j.comnet.2016.06.033
1389-1286/© 2016 Elsevier B.V. All rights reserved.

layers of the mobile network stack and unaware applications that may — unintentionally — be harmful to the user experience. In order to accurately evaluate and improve the quality of TCP streaming applications all of these circumstances beg a proper investigation, a feat which has previously not been undertaken.

The first of three main contributions of this work aims to rectify this problem by setting up a simple, yet flexible, streaming simulation model atop the popular NS-3 network simulator and its LTE-module. The streaming model is described along with the survey that preceded the selection of the mobile simulation and their caveats. As a second contribution, three scenarios are investigated within this framework that demonstrate some exemplary negative interactions of video streaming with some aspect of mobile networking. These basic scenarios make the influence and interaction potential immediately evident and serve as an incentive for further investigations in the future.

Finally, the last contribution takes a closer look at one of the interaction scenarios, namely mobility with periodical loss of radio coverage, which can for example occur in a road tunnel or during handover events. An analytical solution — using additional context information to predict these events in time — is presented that prevents any video stalling from occurring in this particular scenario. Context factors encompass various information sources not directly related to the actual property under scrutiny, including any sensor information from the device, but also preprocessed data from the cloud and other users, e.g., upcoming events derived from large user and social network datasets. The crucial point of this paper is to give a reason and provide the means to investigate such scenarios. If one understands such scenarios, solutions to cross-influences can be discovered more easily, e.g., using contextual information.

This work is an edited and extended version of a previously published publication in [2], with further input derived from [3]. While the original work solely focused on the basic streaming simulation framework aspects, this new publication centers on the breadth of possible interaction scenarios between the mobile network and TCP streaming. Special attention is given to mobility, which is investigated using the framework as well as by formulating an optimization problem, and the usefulness of context monitoring in this regard.

This paper is structured as follows. Section 2 briefly covers the background on TCP streaming and related work, while Section 3 provides a survey of existing network simulators and their suitability for mobile networks. Section 4 then moves on to describe the simulation setup including the employed model of the streaming strategy. Afterwards, Section 5 presents three scenarios of negative interactions between mobile network intrinsics and TCP streaming in the simulator. In a small intermission, Section 6 describes the basics of context factors and monitoring and their purpose for streaming in mobile networks. With this knowledge at hand, Section 7 conducts an analytical breakdown of the scenario derived from the simulations and presents an analytically optimal solution using context information. Finally, Section 8 concludes this paper.

2. Related work

TCP streaming approaches, including the most prominent HTTP-based ones like HTTP Adaptive Streaming (HAS), share certain commonalities related to their TCP-nature. Amongst those, the reliable transmission feature can be one of the most influential — or detrimental — attributes to video streaming. This reliability makes video streaming never miss any video frame or parts of it, any video will be played out in the exact same image quality as it was stored in. But if one intermittent packet is missing, and the video buffer subsequently runs out of data, playback must stop and wait until the buffer gets filled up again.

Therefore, the frequency and lengths of these video playback stalls is the defining quality metric of simple TCP streaming and has been the topic of several past publications (e.g., confer to [4] and [5]). The more flexible HAS variants offer multiple versions of the same video in different quality levels to choose from, either at will or at certain fixed positions during playback. But this also requires additional metrics to fully measure any quality level the player currently displays and has buffered. Previous publications, such as [6], have already recognized the complexity of accurately tackling the quality and behavior of adaptive streaming in a realistic network scenario. This especially includes effects originating from running multiple adaptive streams over the same network, with the control loops of their adaptation processes interacting in unintended ways. Additional feedback loop interactions may also occur with other network protocols, such as the TCP congestion control loop, applications, or network elements.

Next, looking at mobile network architectures, one notices the glaring differences to a typical wired network architecture. Mobile nets are much more vertically integrated and have a pronounced control plane not just at the radio interface, but throughout the whole network, including the Core Network (CN). An incomplete list of relevant specifications for the involved nodes, protocols, and signaling procedures can be found in the Third Generation Partnership Project (3GPP) specifications at [7–14]. Previous publications, e.g., [1,15,16], have already looked at some of the problematic aspects of the strong dependence on signaling, namely the load and overload of core components induced by specific, seemingly benign, application actions and transmission patterns. The authors of [17,18] verify a similar interaction known as a “signaling storm” on the radio interface, which strongly influences the radio network’s load but also equally the user device’s energy consumption and experienced subjective quality.

Similarly, such interactions of seemingly uncoupled network components and layers have already been observed for other fields in the past, for example in the interworking of Peer-to-Peer (P2P) content distribution and operator networks in [19]. Managing the QoE of applications in such ecosystems can therefore become quite the challenge, as [20] observes.

3. Mobile network simulator survey

In order to evaluate certain aspects of a communication network one can take a few different roads, including active measurements, passive measurements, testbeds, simulations, and analytical approaches. But especially when facing mobile networks all come with a certain set of issues. Take passive measurements as an example. To get a complete picture of the mobile network’s user and control plane one would have to deploy and coordinate wiretaps at multiple points, if one even gets the operator’s permission as this tackles both economic and user privacy issues. Similarly, deploying active measurements to many users is a time-consuming and costly task. A possible solution is to employ crowdsourcing as discussed in [21]. To generate meaningful results from mobile devices it is almost always necessary to include additional sensor data through which, e.g., their position and mobility patterns can be inferred, as they will have an influence on any transmission. This data can then be centrally aggregated and processed. Approaches like [22] can help with the collection while simultaneously preserving the participants’ privacy.

Simulative approaches are usually the most accessible and can be used to rapidly test changes to existing properties or entirely new approaches and protocols. Therefore, they are often the preferred means to conduct network evaluations, at the cost of realism and precision. Simulation frameworks are especially important for mobile networks due to the aforementioned issues. This holds equally true for TCP streaming across mobile networks. For a complete evaluation one needs to rapidly modify the parameters of the video player's buffering and transmission strategy in order to test their effects and optimize the resulting playback quality. There are relatively few mobile streaming studies tested through active measurements, e.g., [23], most are conducted in fixed networks.

Before implementing such a streaming framework a viable network simulator, that includes a framework for 3GPP-based mobile networks, has to be found. It would not make much sense to start implementing a new network simulator just for one publication. Creating and maintaining a network simulator requires large efforts in terms of verification of its behavior as well as keeping it up-to-date with the latest network protocols and standards. This is especially critical for mobile networks such as Long Term Evolution (LTE) with its large array of interdependent protocols and network architectures, that have to be mimicked very closely in a simulation to correctly resemble a real network's behavior.

Two of the oft neglected parts of a mobile network in simulation frameworks are the core network as well as the signaling and control plane. Those two can have a large influence on any user plane transmission, and not just the characteristics of the radio transmission as would be immediately evident. Therefore, to better evaluate reliable streaming, a survey on existing mobile network simulation frameworks was conducted first. The following list overviews current publicly available simulation frameworks with either 3G or LTE support:

- An external UMTS module[1] is available for the **ns-2** network simulator. A further, separate collection of patches dubbed E-UMTS system level simulator also extends ns-2 with UMTS radio link capabilities [24]. Both extensions are no longer being developed and not up-to-date to the latest 3GPP specifications. They also focus solely on the radio link's physical and link layer in the user domain of UMTS.
- Another third-party LTE radio link simulation model is available and implemented in MATLAB[2] [25].
- A standalone LTE simulation[3] [26] includes models for some LTE nodes, including the Evolved Node B (eNB) and Mobility Management Entity (MME) and implements a selection of protocols (PDCP, RLC, and RRC). However, the implementation of these nodes and protocols is rudimentary and does not adhere to the specifications very well. Additionally, the simulator lacks a coherent TCP/IP stack as IP is reduced to its basic functionality and TCP is completely absent.
- A framework dubbed SimuLTE[4] is available for OMNeT++.[5] Included are the user plane aspects of the radio link and some basic Serving Gateway (SGW) and Packet Gateway (PGW) functionality.
- The ns-3[6] simulator also contains an LTE/Evolved Packet Core (EPC) module called LENA[7] [27] with features similar to SimuLTE. Again, only user plane SGW/PGW functionality is present with an initial GTP-U implementation.
- Several other commercial simulators are also available, e.g., Riverbed Modeler, but were not further considered due to their commercial nature.

[1] http://net.infocom.uniroma1.it/reti_files/reti_downloads.htm.
[2] http://www.nt.tuwien.ac.at/research/mobile-communications/vienna-lte-a-simulators/.
[3] http://telematics.poliba.it/index.php/en/lte-sim.
[4] https://github.com/inet-framework/simulte.
[5] http://www.omnetpp.org/.
[6] http://www.nsnam.org.
[7] http://networks.cttc.es/mobile-networks/software-tools/lena/.

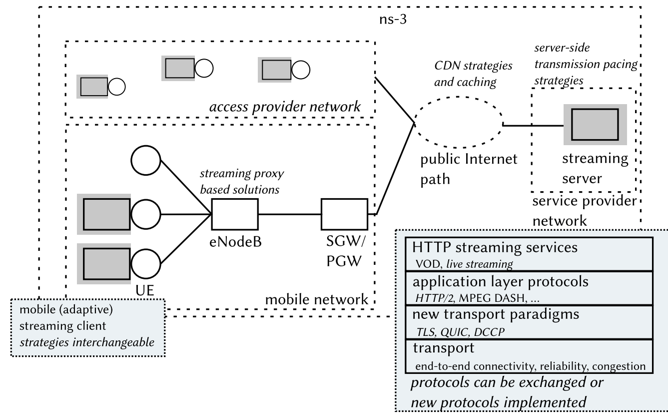


Fig. 1. LTE TCP streaming simulation testbed using a possible network topology. Items with a gray backdrop are contributions of this work. Suggestions for features the framework could be extended with are marked in italics.

The goal is to simulate reliable streaming in a realistic mobile environment. That would ideally include both a complete horizontal network path — the radio link, the access, and the core network — as well as the full vertical network stack — comprising both user plane and control plane. Unfortunately, none of the above feature a complete representation, especially in terms of the control plane and some off-path core network components. Nonetheless, the simulators can still provide a viable basis for a mobile streaming framework as long as these limitations are kept in mind. The choice as a foundation for reliable streaming simulations was made for ns-3. Alongside with SimuLTE it has the most complete LTE representation for the goals of this work. With the exception of OMNeT++, which has comparable capabilities, ns-3's TCP/IP implementation is much more complete and realistic than that of the other frameworks. Additionally, it can also incorporate the actual TCP/IP stack of certain Linux kernels using the Network Simulation Cradle (NSC) and Direct Code Execution (DCE) projects. This is especially important when considering the diversity of TCP implementations and other transport protocols running in current operating systems. That diversity could never be fully reflected in a simulator and would limit the realism of any result.

In the long run, to better represent actual mobile networks, the base radio framework in ns-3 would still need to be extended with more control plane aspects and continuously adapted to the latest 3GPP specifications.

4. TCP Streaming simulation framework

With ns-3 chosen and the core network model set, the task is now to define and implement TCP streaming on top of the LTE network. This testbed should allow arbitrary TCP streaming playback approaches to be tested and optimized for the various conditions and pitfalls in mobile networks at which the subsequent Section 5 will give an initial glance.

The initial network model is kept as simple as possible as depicted in Fig. 1. Modules for the User Equipments (UEs), eNB, and a combined SGW/PGW node are readily provided by the simulator. The full mobile network protocol stack is available for the user plane traffic on all these links. Our contribution to this setup is the implementation of a streaming server and client.[8]

4.1. Streaming server

The streaming server represents the node from which the client requests and pulls the video segments. It is connected through a

[8] Source code is available at https://github.com/mas-ude/lteNS3.

Please cite this article as: F. Metzger et al., TCP video streaming and mobile networks: Not a love story, but better with context, Computer Networks (2016), http://dx.doi.org/10.1016/j.comnet.2016.06.033

link, representing the Internet, to the mobile ISP's PGW. To set up more realistic scenarios, the simulated network can be easily expanded to include more streaming clients, or other nodes generating background traffic. Similarly, the typical CDN structures of today's streaming service providers can also be recreated more accurately. Upon receipt of a request for a specific segment over TCP an appropriately sized segment is immediately sent back to the streaming client. The TCP congestion avoidance mechanism implemented by NS-3 is New Reno, which has to be kept in mind when evaluating any simulation results.

Typically, TCP streaming is conducted by using HTTP as an application layer protocol. In this setup however, TCP will be directly used to transport the stream in order to further reduce additional sources of influence. Therefore, the simulation can be used to imitate any type of TCP streaming approach or content type. The streaming client and server could be adjusted accordingly, depending on the desired outcome, e.g., to either resemble video on demand or a live source. HTTP is just considered in the form of an appropriately sized data overhead. Newer protocols and protocol variations such as, e.g., using HTTP/2 or other transport protocols add some interesting features and behavioral patterns, that may be worth investigating separately. However, this would even further complicate the goal of finding interactions between the streaming and the mobile network stacks and is therefore neglected for this first investigation. Yet, it might be very worthwhile for follow-up examinations.

4.2. Streaming client

The client's playback buffering model is implemented as a module running atop the UE. The component initiates the streaming process and takes care of requesting each individual segment. The video playback simulator reads a list of frames and their sizes and synchronizes this information with the received amount of data to correctly calculate the video buffer and the number of frames contained within. The frame list is generated from actual video data using external tools such as `ffmpeg`. This information is then combined to simulate the playback process and calculate the resulting quality metrics, i.e. the stalling characteristics and — in case of adaptive streaming — the selected quality level. In addition, full network traces can be kept by NS-3 for each link for further evaluation.

One of the core elements of any TCP streaming client is its transmission and playback strategy. Much research has been conducted especially in the area of adaptive streaming strategies, suggesting paths to optimize adaptive strategies for example in terms of their average quality level and their number of layer switches [28] as well as implementing an improved QoE management for the streaming process [29]. Furthermore, subjective user studies have also been conducted that underline the impact of certain adaptation parameters [30]. This research shows that video quality should be maximized first, with a lesser importance on the number of quality switches. The information available at the streaming client allows for an implementation of a multitude of different playback strategies, i.e., buffer-based strategies, based on QoS monitoring and bandwidth predictions, see [6]. For reasons of simplicity the initial framework only contains one simple exemplary strategy that aims to demonstrate the interaction issues between streaming and mobile nets in the next section.

This strategy employs segmented video files and requests each segment at a time deemed appropriate. It further uses four thresholds that govern the buffering behavior, each representing a distinct buffer fill level at which the playback or the transmission will either be started or stopped respectively. Fig. 2 demonstrates this strategy in an example scenario. The threshold values were chosen arbitrarily and do not represent an optimal solution but rather serve as a showcase.

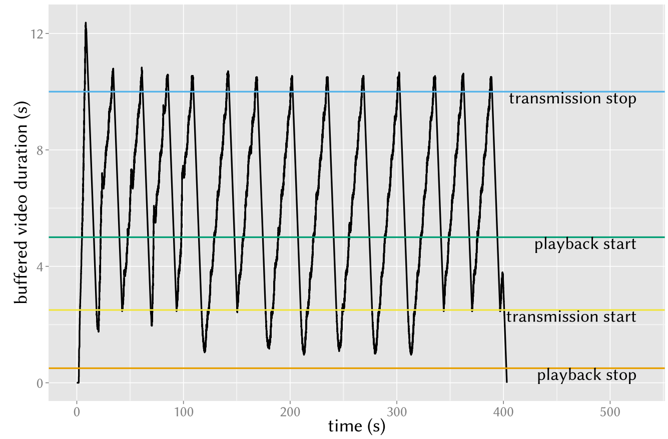


Fig. 2. Sample simulation run demonstrating the four threshold strategy.

4.3. Basic scenario and streaming process

The streaming framework also defines a basic mobile network scenario to be used as a foundation for future experiments. During transmission, the existing LTE framework sets up the lower layer protocols, including the radio stack and the GPRS Tunneling Protocol (GTP) core network bearer, accordingly. In this basic model, only one eNB and one UE are present in order to avoid interference with other devices. Additionally, a stationary mobility model was selected for the target UE. Furthermore, the link between the PGW and the streaming server acts as the Internet link. Here, any conditions that would be present during a transmission and are not intrinsic to the mobile network, should be situated. This can include adding a certain amount of delay and packet loss or limiting the throughput to account for the server's distance to the mobile network. By modeling this link or supplanting it with a more concrete network model, a Content Distribution Network (CDN) infrastructure or other use cases, such as edge caching, can be simulated and evaluated (also confer the options in Fig. 1 again). Apart from the described settings the LTE nodes are otherwise left at their default configuration, which should yield an overall net bandwidth of $80\,\frac{\text{Mbit}}{\text{s}}$ in the LTE radio cell. The validity of the LTE model is taken as-is and covered by, e.g., [27].

5. Mobile network and streaming interaction scenarios

During the course of this section three simple examples will demonstrate possible interaction scenarios of video streaming with mobile networks. These are not intended to uncover every mobile streaming interaction, but rather serve to showcase the range of problems that can occur. Therefore, the scenarios are specifically chosen to be well-known and easily explainable. The first two examples emulate issues on the Internet path. The third scenario extends the setup and takes a closer look at the effects of mobility and handovers. In every evaluated case the aforementioned four-threshold strategy is facilitated and tested with the three videos described in Table 1, each representing a different quality profile.

5.1. Scenario 1: segment retrieval during high latency

In this first scenario the latency of the streaming server's link was increased incrementally up to an additional latency of 1000ms. The values were set deterministically with no probability distribution. High latency is typically present, e.g., due to congestion, a

Please cite this article as: F. Metzger et al., TCP video streaming and mobile networks: Not a love story, but better with context, Computer Networks (2016), http://dx.doi.org/10.1016/j.comnet.2016.06.033

Table 1
Parameters of the video used in the streaming simulation scenarios.

	Low quality	Standard quality	High quality
Duration	318s	602s	596s
Size	19.2MiB	258MiB	853MiB
Frame rate	29.97Hz	29.97Hz	24Hz
Avg. bitrate	504 $\frac{\text{kbit}}{\text{s}}$	3596 $\frac{\text{kbit}}{\text{s}}$	12 $\frac{\text{Mbit}}{\text{s}}$

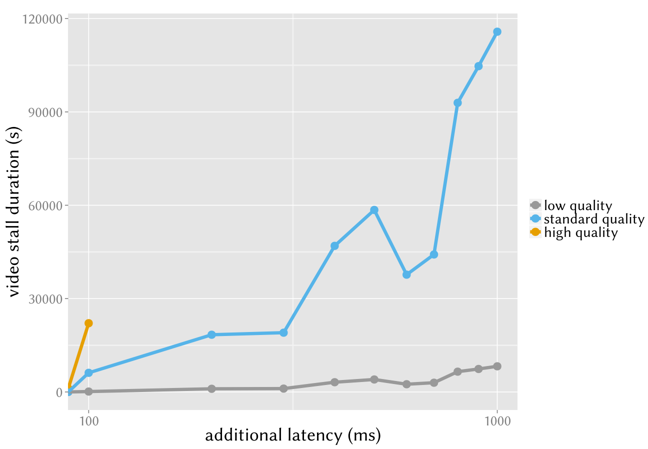


Fig. 3. Relative stalling duration of the simulated streaming player under increased Internet latency.

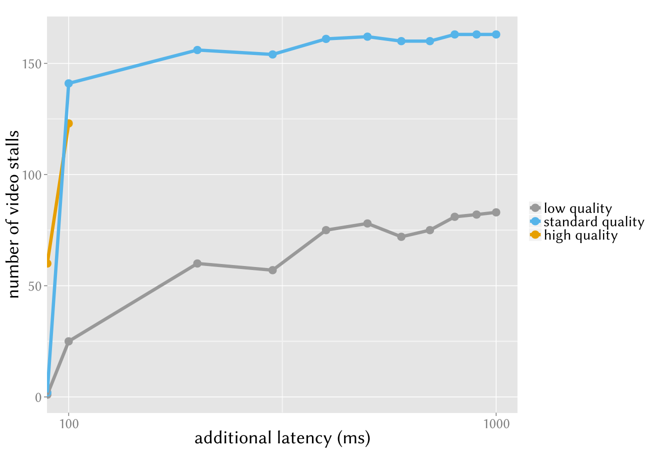


Fig. 4. Number of stalling events of the simulated streaming player under increased Internet latency.

large distance between the video caches and the streaming client, or, in the case of mobile networks, due to a combination of radio propagation effects, mobility, and buffering as well as queuing by the mobile network nodes. While the impact of a high delay variation, another typical property of mobility, is not tested here, even a constant delay can have interesting repercussions.

Figs. 3 and 4 show the results in terms of the relative stalling duration as well as the number of stalling phases. The playback simulation seems to be excessively sensitive to latency increases. Even a rather small 100 ms increase makes the high and standard quality videos completely unwatchable as the stalling duration is more than ten times longer than the actual video. With latencies beyond 100ms the high quality video stream did not successfully finish during the allotted simulation time, thus no results were produced.

One of the reasons for this unexpectedly strong negative influence can be easily explained by looking at the request and retrieval strategy used in this experiment: New video segments are

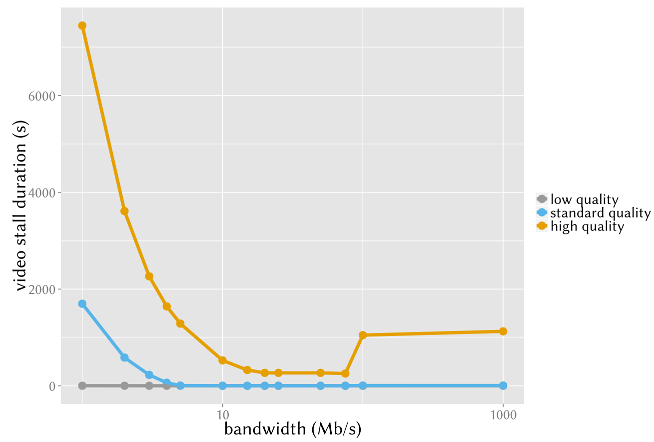


Fig. 5. Relative stalling duration of the simulated streaming player under limited Internet bandwidth.

only requested when the current one has finished transmitting. As this takes a full round trip between client and server, the latency has a significant influence on the arrival time of subsequent segments. This behavior is known as *stop-and-wait* in the literature and can never saturate any kind of link. One usually expects to find this behavior, if at all, at the transport layer, but it may be rather unexpected to also notice it in the usage pattern of an application layer protocol. But due the continuous requests of deadline-sensitive video segments, this phenomenon emerges here as well.

To avoid the stop-and-wait behavior, new segments need to be requested sufficiently in advance while the previous segment is still being transmitted, so that the full bandwidth is always utilized. With these improvements to the retrieval strategy this specific issue can be easily eliminated. But this shows just one of the many pitfalls and influences of various layers. A simple implementation detail may have large implications on the streaming quality the user experience.

5.2. Scenario 2: high bandwidths and buffering

In the second scenario instead of increasing the delay of the Internet path its bandwidth gets constricted, thus emulating a congested bottleneck link. The link is configured to bandwidths between 1 $\frac{\text{Mbit}}{\text{s}}$ and 1 $\frac{\text{Gbit}}{\text{s}}$. With the default LTE network configuration the radio link reaches a maximum net bandwidth of only about 80 $\frac{\text{Gbit}}{\text{s}}$ as previously mentioned. It should therefore be interesting to see if the Internet link can oversaturate the radio connection.

The results are depicted in the Figs. 5 and 6. As soon as the link's bandwidth exceeds the video's bit rate, the number and duration of stalls are reduced. Stall events and duration in both the low and standard quality videos drop to zero, only the high quality video shows slightly differing results. As its bit rate of 12 $\frac{\text{Mbit}}{\text{s}}$ should be perfectly manageable for the radio link some other explanation is required. An additional effect can be observed at link speeds of 100 $\frac{\text{Mbit}}{\text{s}}$ and above, where the stalling phases suddenly experience an unexpected increase. As this happens exactly when the transmission speed exceeds the radio links capacity it might be an indication of the negative interaction of both LTE's reliable link layer transmission, i.e., the Hybrid Automatic Repeat Request (HARQ) mechanism, with TCP's congestion avoidance mechanisms.

This is likely related to bufferbloat issues, discussed, e.g., in [31]. TCP can not properly detect the bottleneck capacity in a timely manner as LTE attempts to buffer and retransmit any packet exceeding the radio capacity below the user Internet Protocol (IP) layer. As the buffers fill up, and the latency induced by the buffer increases ever more, the video segments will arrive past their

ARTICLE IN PRESS
JID: COMPNW [m5G;July 14, 2016;10:42]

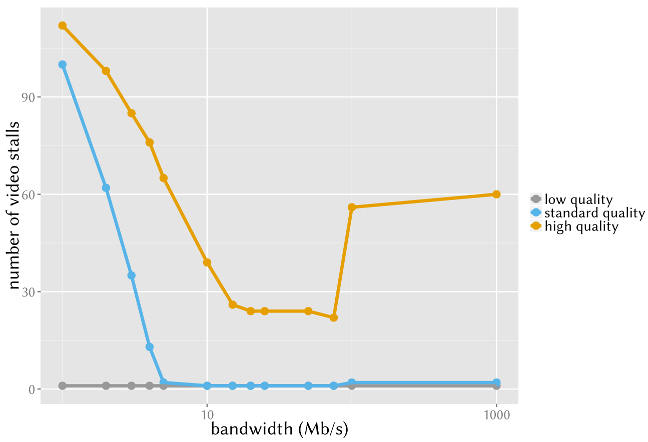


Fig. 6. Number of stalling events of the simulated streaming player under limited Internet bandwidth.

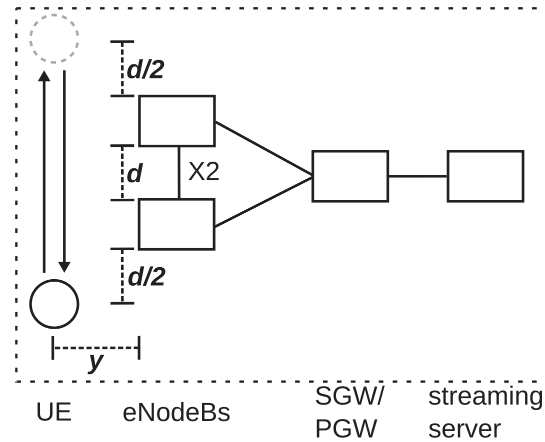


Fig. 7. Simulated handover mobility scenario using waypoints.

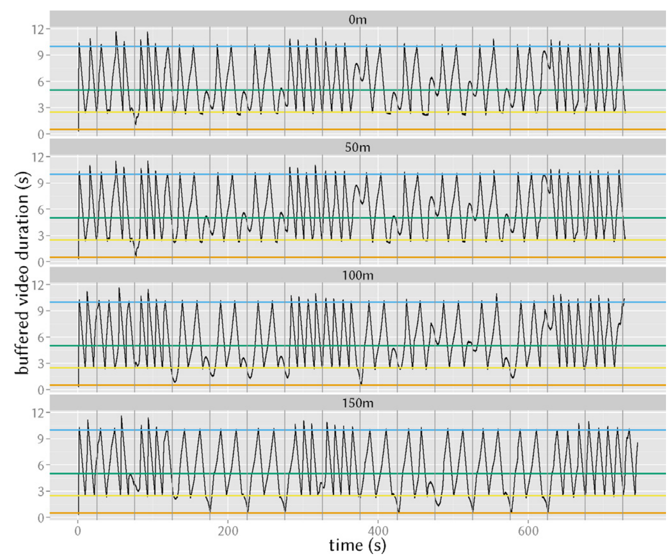


Fig. 8. Playback buffer time series of the simulated mobility experiments with increasing distance between device and pleNB.

deadline, leading to stalling. The effective goodput seems to get reduced to about 5$\frac{\text{Mbit}}{\text{s}}$, which is below the required bandwidth for the high quality video. This detrimental interaction of TCP and the link layer features of mobile networks has been known to cause problems for some time (cf., e.g., [32]), but is especially influential on time-sensitive applications like video streaming and needs to be better researched in this context to find viable solutions.

5.3. Scenario 3: device mobility

The final simulation scenario aims to investigate effects of mobility. Mobility is one of the most crucial features of mobile networks, but it also brings along a series of issues that can occur in certain situations. Today's mobile users expect that streamed videos also work when they are on the move, which can be a demanding task for a streaming strategy to achieve as the radio link's properties and capacity can rapidly change over time. But even the loss of radio coverage for a certain period — e.g., due to fading or during handover — could become a foreseeable and manageable event with the right amount of information at hand.

For this scenario to be reflected in the simulator, a secondary eNB is added to the network. Both are connected through the X2 interface, which is facilitated by the eNB-anchored mobility provided by NS-3. Instead of having a constant position relative to the eNB, the UE will now move back and forth between two waypoints on a path running in parallel to the two base stations. A handover is triggered each time the device leaves the range of one station and enters the other one's coverage area. Additionally, at the coverage edges of each base station, the radio bandwidth might drop down to levels insufficient for a stable video stream. Fig. 7 depicts this scenario and the positioning of the waypoints. For the purpose of this experiment, the two eNBs were placed at a vertical distance of $d = 500$ m while the horizontal distance to the device was varied between $y = 0$ m and $y = 150$ m. The device is moving at a constant velocity of 20$\frac{\text{m}}{\text{s}}$. During the movement the standard quality video was streamed to the device and played back.

The resulting buffer fill levels are displayed in Fig. 8. The color-coded horizontal lines once again demark the same thresholds from the previously described four-threshold segmented streaming strategy. The handover events between the two eNB are denoted by the gray vertical lines and occur as expected in regular intervals. The actual control transfer usually last for only about 40 ms in this setup. But looking at the figures it is immediately evident that the video buffer, and therefore the throughput of the video segments, already suffers seconds before this handover event when the edge of the radio range of the currently active eNB is reached.

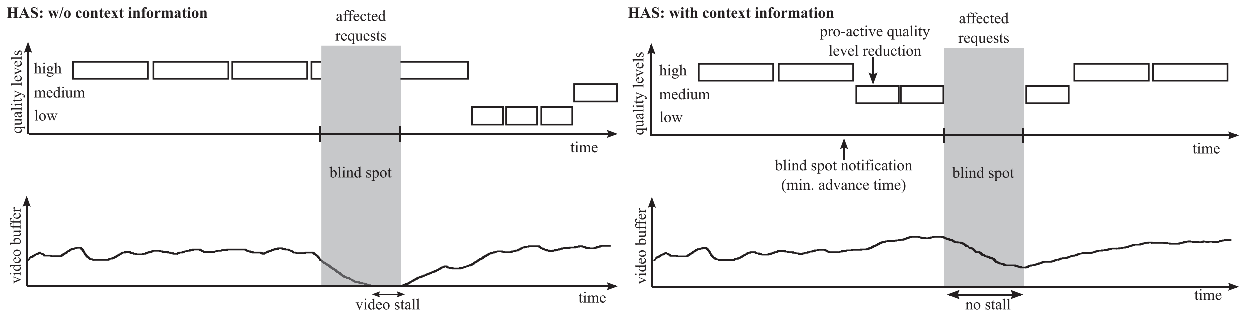
The playback stop threshold is undercut several times right at the handover mark resulting in a stalling phase.

It seems, that the throughput at the radio edge drops below the video's average bitrate of 3596$\frac{\text{kbit}}{\text{s}}$. Therefore, the buffer cannot be maintained for this quality level. If the buffer was at a critically low level beforehand, it will run empty and lead to stalling due to the mobility. Additionally, there is a dependency between the stalling probability and the distance between device and base station. A farther distance decreases the achievable throughput leading to an increased chance of the buffer running dry.

This scenario shows some of the many pitfalls that can occur in the interaction of mobility and TCP streaming. While some optimizations can probably be achieved by improving existing buffering strategies, such sudden loss events can still be detrimental to the playback quality. The next two sections therefore aim to present an in-depth analytical view and a possible solution to such a generalized scenario using additional context information.

6. Context factors and context awareness

Context may refer to "*anything that can be used to specify or clarify the meaning of an event*" [33]. In that sense, at least in a mobile network ecosystem, context constitutes a pertinent concept of the Big Data Era. More specifically, data acquisition is a tool



(a) Stalling occurs without context hinting.

(b) Stalling can be prevented through context hinting and filling the playback buffer in anticipation.

Fig. 9. Adaptive video streaming scenario with and without context information and cross-layer hinting.

towards context awareness, or, to put it otherwise, context awareness is the ultimate objective of collecting data in a mobile network environment. Context awareness, which therefore becomes possible through Big Data, may induce the transition of a network from pure packet-level management to a more sophisticated “scenario-based” management. Knowing or being able to learn the context factors relevant to a specific setting may facilitate a higher QoE for the end-users but also a more efficient resource provisioning in the network. This potential has been recently identified in the literature and research on context awareness and context-aware network mechanisms is ongoing. In [34], a context aware handover management scheme for proper load distribution in a Wi-Fi network is proposed, while in [35], the impact of social context on compressed video QoE is investigated. Moreover, in [36] a novel decision-theoretic approach for QoE modeling, measurement, and prediction is presented, whereas in [37] a case study of flash crowds demonstrates the benefits of context monitoring to QoE.

The first requirement towards exploiting context is the identification of the impact factors per scenario, as well as understanding their influence on the user’s QoE. Taking a mobile user as an example, it is valuable to define the context factors that need to be monitored, as well as those factors that can be managed to improve the user’s experience. For example, one could monitor a user’s trajectory and speed and subsequently manage network load through the user device’s association to specific base stations. One way to identify the most important context factors per case is through the monitoring and aggregation of data from many users, helping to derive context and predictions from this dataset. For instance, the acquired data may be used to enhance the users’ QoE, or predict imminent network problems and bottlenecks. Some of the possible context data that can be monitored in a mobile network are: *(a)* the current link state (e.g., the active Radio Access Technology (RAT) or information on neighboring cells), *(b)* spatial information (e.g., location and trajectory, terrain characteristics, or the presence of blind spots such as areas of insufficient radio coverage or limited capacity), *(c)* the current and upcoming network load, *(d)* the device’s state and information on applications (e.g., current CPU usage or energy consumption).

It needs to be noted that context awareness does not necessarily rely on predictions, e.g., future traffic demands. Having just access to current information, like for example the connectivity type, already enables many QoE optimizations. Coming back to video streaming, context monitoring can therefore be used either as a direct source of information for streaming strategies, or alternatively for building sufficiently precise prediction models, for example through machine learning techniques.

Context can be easily exploited to manage and improve mobile networks. This can occur either on an end-to-end basis or from the perspective of the network operator, depending on where the necessary context information is present or will be signaled to. With context-awareness, a network operator can take more sophisticated network-wide management decisions aimed at an increase in efficiency in terms of, for example, spectrum, energy, or other resources, with the ultimate goal often being a reduction in operational expenses. For instance, an operator can exploit information about flash crowds to drive an effective CDN load balancing strategy [37]. Such network-wide centralized monitoring solution might give more insights in certain cases but also come at the cost of privacy.

While solutions to this issue may exist — options like opt-in, anonymization and blurring techniques come to mind — the most privacy-friendly picture of context factors related to specific user applications presents itself only at the mobile device itself, and can also only be used there in the most optimal way. Specifically, for video streaming, the segment and quality selection strategy in an HAS player is in an ideal position to influence and optimize the streaming quality, even without any network assistance. Context-awareness therefore seems like a perfect solution to the issues that occurred in the mobility scenario in Section 5.3. If the streaming strategy would have knowledge of an upcoming loss of radio connectivity the video buffer could be filled in anticipation of this event in order to avoid any video stalling. This scenario is closely investigated in the following section.

7. Improving mobile streaming with context

As should be evident by now, TCP streaming in a mobile environment is a challenging issue. Properties of the wireless medium — namely the effects of path loss, shadowing, fading, or penetration loss — combined with user mobility can lead to significant fluctuations of the channel quality over time and location. Due to these phenomena, a mobile user may find herself in a *blind spot*: an area of insufficient radio coverage, where an application is not able to satisfy its traffic demands anymore. In video streaming applications this will drain the video buffer and cause stalling. Context-awareness presents itself as a possible solution to these scenarios, by anticipating upcoming blind spots and implementing a proactive streaming strategy. Such a strategy could be taking measures like temporarily increasing the maximum video buffer size, requesting segments more rapidly, or by requesting lower quality video segments sufficiently in advance and thereby filling the buffer. The effects of context-based streaming are illustrated in Fig. 9. Adapting the streaming strategy to specific conditions known in advance has been conducted in the literature before, e.g., in [38].

7.1. Scenario definition

Consider a user traveling in a vehicle, following a specific route at a certain velocity. In today’s context of Big Data it can easily

Please cite this article as: F. Metzger et al., TCP video streaming and mobile networks: Not a love story, but better with context, Computer Networks (2016), http://dx.doi.org/10.1016/j.comnet.2016.06.033

be assumed that historical data about this user is available (either locally or with a Cloud service), for instance regarding the user's usual routes and traveling behavior. Therefore, a prediction of the user's current trajectory can be made, which, combined with the vehicle's velocity, can reveal her future position. Likewise, regions with, e.g., bad signal reception can also be mapped through continuous measurements. Such data can be collected and provided to the users either directly by the operators or through crowdsensing [22]. Now assume that the mobile user is moving towards such a known blind spot. As long as the user's video streaming application triggers a proactive streaming strategy sufficiently in advance, video stalling could be avoided. In the demonstrative approach proposed here the strategy deliberately begins requesting segments from lower quality layers in advance of such blind spots to prevent stalling. Next, an optimization problem is formulated that estimates the selected HAS layers in such a way that the user's QoE is kept as high as possible while the number of switches between different HAS layers are kept as low as possible.

7.2. Optimization problem

This strategy can also be described as a Mixed Integer Linear Program (MILP). Even though an optimal solution to this optimization problem is *NP*-complete, MILP heuristic solutions are close enough to the optimum for all practical purposes. In HAS a video consists of n segments that each have the same playback length τ. Each segment i belongs to one of the $r_{\max}$ quality layers and needs to be downloaded until the segment's deadline D_i. Since there is an initial delay T_0 before the video is being presented to the user, the deadline is according to [30]

$$D_i = T_0 + i\tau, \quad \forall i = 1, \dots, n.$$

The size of a segment in layer j in bytes is denoted as S_{ij}. Moreover, each segment is associated with a quality value w_{ij}, which can be considered equal to the quality layer (or $w_{ij} = j$). The total data that has been downloaded at a point in time t is $V(t)$. Thus, the optimization problem can be formulated as follows: minimize the number of switches (scaled by α) and the quality value (scaled by β where $\beta < 0$) for a single user without stalling, or

$$\text{minimize} \sum_{i=1}^{n-1}\sum_{j=1}^{r_{\max}} \alpha (x_{ij} - x_{i+1,j})^2 + \sum_{i=1}^{n}\sum_{j=1}^{r_{\max}} \beta w_{ij} x_{ij}$$

subject to

$$x_{ij} \in \{0, 1\}$$

$$\sum_{j=1}^{r_{\max}} x_{ij} = 1, \forall i = 1, \dots, n$$

$$\sum_{i=1}^{k}\sum_{j=1}^{r_{\max}} S_{ij} x_{ij} \le V(D_k), \forall k = 1, \dots, n.$$

The first constraint in this optimization problem implies that the unknown variable is binary, while the second constraint ensures that only one quality layer can be downloaded per segment. Finally, the last constraint guarantees that the segments are downloaded before their respective deadlines. This approach merges two of the optimization problems formulated in [30] into one by considering the sum of the optimization measures and scaling them. This has the advantage that switches can be attributed with a certain scaling factor in relation to the quality. In contrast, the original formulation has the disadvantage that any minor improvement in quality always comes before minimizing the number of quality switches.

Next, a set of coverage zones Q_{zone} is added, where the bandwidth is set to 0. A blind spot (l, m) starts at segment l and ends at m. In order to avoid video stalling in the blind spot, the video needs to be downloaded up until segment m. This can be expressed by adding an additional constraint, thus

$$\sum_{i=1}^{m} \Delta S_i x_i \le \Delta V(D_l), \forall (l, m) \in Q_{\text{zone}}.$$

It should be noted that stalling events are not considered in our model. Instead, the model works under the assumption that stalling can always be prevented by switching to a lower layer. For the sake of simplification, the initial delay is also ignored in our model. However, it would be easy to integrate simply by adding the first optimization problem from [30] into the above MILP.

7.3. Evaluation

The objective is now to evaluate the proposed proactive HAS strategy and to compare it with a state-of-the-art strategy that is not context-aware. Additionally, a comparison with the optimal solution is given in terms of QoE and the number of quality switches. For this simulation, a blind spot with a duration of 150s is assumed, during which there is no radio connectivity. To simulate the available link capacity, a "bandwidth factor" is used, which is a metric of the network congestion [30]. Real network traces are used as input information about the data rates in the network. Furthermore, it is possible to indirectly enforce a congested network by multiplying with the bandwidth factor, ranging between 0 and 1 (the higher this factor the lower the congestion). The collected results are presented in Fig. 10.

Fig. 10a depicts the requirement to the advance time window, i.e., the minimum time required for the proactive HAS strategy to overcome the imminent blind spot. For very low data rates (e.g., a bandwidth factor of 0.2), the minimum required advance time is higher, because the user would need longer to fill the buffer in advance of the blind spot. However, the uncertainty (depicted as a shaded area in the figure) when taking accurate proactive measures is very high, especially when these measures need to be taken far in advance.

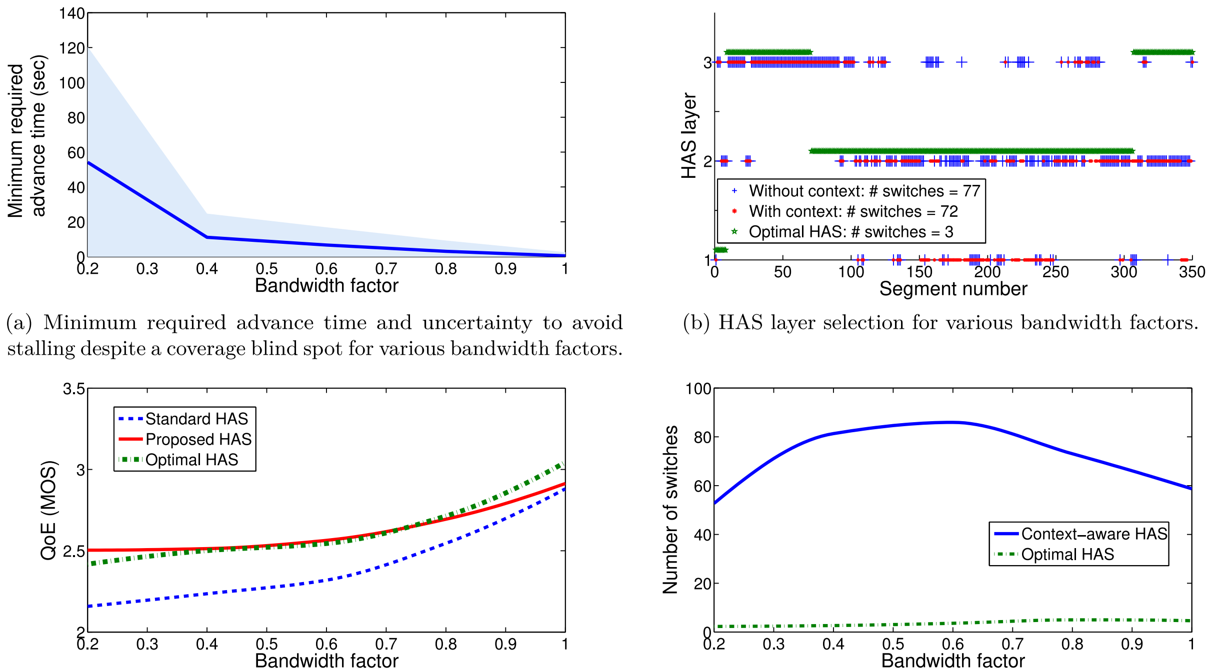
In Fig. 10b the selection of HAS layers for *(a)* the optimal solution, *(b)* the proactive HAS strategy, and *(c)* the state-of-the-art context-unaware strategy is presented. The optimal solution achieves a very low number of switches among the three available HAS layers, while it starts requesting segments at the medium quality layer earlier than the proposed strategy. In terms of QoE, the proposed strategy performs as well as the optimal one, albeit with considerably more quality switches (confer to Figs. 10c and 10 d). It is interesting to observe that the resulting QoE from the proactive HAS strategy seems to be higher than for the optimal case. However, this is simply because the employed QoE metric only takes the time on the highest layer into account and does not factor in the switching frequency. It should additionally be noted that the impact of switches on QoE is of much lower significance compared to the time on the highest layer according to [39].

Nevertheless, an enhanced proactive strategy that prevents overly frequent switches could be also devised, together with a QoE model that accounts for more than one quality influence factor (e.g., the frequency and amplitude between subsequent quality switches, among others). More extensive work on this topic has been conducted in [3]. For this work the task was solely to successfully provide a solution that can avoid stalling in the mobility scenario from Section 5.3.

7.4. Big data requirements and implications

So, in order for the proposed HAS strategy to work, exactly what kind of data is necessary and how can it be collected?

ARTICLE IN PRESS



(a) Minimum required advance time and uncertainty to avoid stalling despite a coverage blind spot for various bandwidth factors.

(b) HAS layer selection for various bandwidth factors.

(c) Comparison of three HAS strategies in terms of QoE.

(d) Number of switches for the optimal and proactive strategy.

Fig. 10. Evaluation of the proactive HAS strategy.

Besides the usual information necessary for the operation of a streaming strategy, i.e. the video file's metadata, buffer characteristics, and throughput estimations, some further context data is required, namely information on the device's location and trajectory as well as the location of areas with coverage issues. The device's own position is easily acquirable through the usual built-in sensors. From this predictions about the future position and trajectory can be deduced in order to find out whether a user is moving towards a blind spot. This can be conducted, for instance, using path prediction techniques in combination with an accurate map of the area, or simply using historical routes of the user. Lastly, the strategy needs to calculate the minimum required advance time to start a proactive action, but this can be directly derived from both one's own location and trajectory as well as the blind spot's location.

Regarding the availability of the geographical location of a blind spot, this data can either be collected locally at the device itself, or through a centralized context service that aggregates crowdsourcing data. In the first case, the device will recognize radio problems on frequent routes. But the dataset will be incomplete for deviations to daily patterns. This is where a network-wide service could help out, as it would aggregate information from many devices over long periods of time to eventually create a coverage map that covers larger areas. Such a service could even facilitate the creation of spatio-temporal coverage profiles which would help uncover radio connectivity issues through predictable albeit time-limited events such as heavy traffic. As previously discussed in Section 6 any context information service comes with certain costs in terms of privacy. A careful balance between those two objectives — preserving privacy and increasing the user's QoE — would need to be found, but this is no task for this work.

8. Conclusions and future work

TCP-based video streaming and mobile networks can interact in the most interesting ways as this work aimed to demonstrate. These kinds of interactions are not easy to uncover due to the sheer complexity of mobile networks and the lack of tools for proper investigations in the past. While the former issue can only be unraveled through strenuous, time-intensive research, work conducted in this paper can help improve the situation around the latter issue. The mobile streaming examination framework compiled here can provide an initial foundation for further interaction evaluations. The LTE models currently implemented in free network simulators are finally mature enough to provide some insights into the performance of mobile streaming applications, albeit with the limitation of the missing control plane signaling interactions and load.

The three initial experimental scenarios, that were set up using this framework, already showed some pitfalls in the way streaming transmission strategies can be set up, and a surprising interaction of the TCP/IP stack with LTE's mobile network design. It can therefore be concluded that appropriate measures need to be taken to ensure a consistent streaming quality. Amongst others, this involves, for example, a careful choice of the playback strategy and its parameters as well as thresholds. The additional analytical investigation of scenarios that can occur during device mobility, namely the intermittent loss of the radio connection and the subsequent chance of stalling, also demonstrated the positive impact the monitoring of context factors and context-aware streaming strategies can have. Both the acquisition of context information as well as the successive prediction of blind spot events can be tackled by a mixture of cross-layer information exchange, crowdsourcing, and Big Data and are an endeavor for future investigations.

All in all, it can be said that the knowledge of interactions between mobile nets and video streaming is still in its infancy. Further research on this topic and on remedies for specific interactions, like context awareness, shows much prospect.

References

[1] F. Metzger, A. Rafetseder, P. Romirer-Maierhofer, K. Tutschku, Exploratory Analysis of a GGSN's PDP Context Signaling Load, J. Comput. Netw. Commun. (2014), doi:10.1155/2014/526231. http://www.hindawi.com/journals/jcnc/2014/526231/.

Please cite this article as: F. Metzger et al., TCP video streaming and mobile networks: Not a love story, but better with context, Computer Networks (2016), http://dx.doi.org/10.1016/j.comnet.2016.06.033

[2] F. Metzger, C. Steindl, T. Hoßfeld, A simulation framework for evaluating the QoS and QoE of TCP-based streaming in an LTE network, in: Teletraffic Congress (ITC 27), 2015 27th International, 2015, pp. 168–176, doi:10.1109/ITC.2015.27.

[3] E. Liotou, T. Hoßfeld, C. Moldovan, F. Metzger, D. Tsolkas, N. Passas, Enriching HTTP adaptive streaming with context awareness: a tunnel case study, in: Communications (ICC), 2016 IEEE International Conference on, 2016.

[4] F. Metzger, A. Rafetseder, K. Tutschku, A performance evaluation framework for video streaming, in: Packet Video Workshop (PV), 2012 19th International, 2012, pp. 19–24, doi:10.1109/PV.2012.6229739.

[5] T. Hoßfeld, R. Schatz, E. Biersack, L. Plissonneau, Internet video delivery in YouTube: from traffic measurements to quality of experience, in: E. Biersack, C. Callegari, M. Matijasevic (Eds.), Data Traffic Monitoring and Analysis, Lecture Notes in Computer Science, 7754, Springer Berlin Heidelberg, 2013, pp. 264–301, doi:10.1007/978-3-642-36784-7_11. http://dx.doi.org/10.1007/978-3-642-36784-7_11.

[6] M. Seufert, S. Egger, M. Slanina, T. Zinner, T. Hobfeld, P. Tran-Gia, A survey on quality of experience of http adaptive streaming, Commun. Surv. Tutorials, IEEE 17 (1) (2015) 469–492, doi:10.1109/COMST.2014.2360940.

[7] 3GPP, Evolved Packet System (EPS); 3GPP EPS AAA interfaces, TS 29.273, 3rd Generation Partnership Project (3GPP), http://www.3gpp.org/DynaReport/29273.htm.

[8] 3GPP, Network architecture, TS 23.002, 3rd Generation Partnership Project (3GPP), http://www.3gpp.org/DynaReport/23002.htm.

[9] 3GPP, Handover procedures, TS 23.009, 3rd Generation Partnership Project (3GPP), http://www.3gpp.org/DynaReport/23009.htm.

[10] 3GPP, Mobile radio interface Layer 3 specification; Core network protocols; Stage 3, TS 24.008, 3rd Generation Partnership Project (3GPP), http://www.3gpp.org/DynaReport/24008.htm.

[11] 3GPP, GPRS Tunnelling Protocol for User Plane (GTPv1-U), TS 29.281, 3rd Generation Partnership Project (3GPP), http://www.3gpp.org/DynaReport/29281.htm.

[12] 3GPP, 3GPP Evolved Packet System (EPS); Evolved General Packet Radio Service (GPRS) Tunnelling Protocol for Control plane (GTPv2-C); Stage 3, TS 29.274, 3rd Generation Partnership Project (3GPP), http://www.3gpp.org/DynaReport/29274.htm.

[13] 3GPP, General Packet Radio Service (GPRS) enhancements for Evolved Universal Terrestrial Radio Access Network (E-UTRAN) access, TS 23.401, 3rd Generation Partnership Project (3GPP), http://www.3gpp.org/DynaReport/23401.htm.

[14] 3GPP, General Packet Radio Service (GPRS); Service description; Stage 2, TS 23.060, 3rd Generation Partnership Project (3GPP), http://www.3gpp.org/DynaReport/23060.htm.

[15] F. Metzger, Evaluating reliable streaming in mobile networks, 2015 Ph.D. thesis. http://eprints.cs.univie.ac.at/4376/.

[16] 3GPP, Study on Core Network (CN) overload solutions, TR 23.843, 3rd Generation Partnership Project (3GPP), http://www.3gpp.org/DynaReport/23843.htm.

[17] C. Schwartz, T. Hoßfeld, F. Lehrieder, P. Tran-Gia, Angry Apps: the impact of network timer selection on power consumption, signalling load, and web QoE, J. Comput. Netw. Commun. (2013), doi:10.1155/2013/176217.

[18] E. Gelenbe, O.H. Abdelrahman, Countering mobile signaling storms with counters, in: Proc. Int'l Conf. on Cyber Physical Systems, IoT and Sensors Networks (Cyclone), Rome, Italy, 2015.

[19] G. Dan, T. Hoßfeld, S. Oechsner, P. Cholda, R. Stankiewicz, I. Papafili, G. Stamoulis, Interaction patterns between P2P content distribution systems and ISPs, Commun. Mag. IEEE 49 (5) (2011) 222–230, doi:10.1109/MCOM.2011.5762821.

[20] T. Hoßfeld, R. Schatz, M. Varela, C. Timmerer, Challenges of QoE management for cloud applications, Commun. Mag. IEEE 50 (4) (2012) 28–36, doi:10.1109/MCOM.2012.6178831.

[21] M. Hirth, T. Hoßfeld, M. Mellia, C. Schwartz, F. Lehrieder, Crowdsourced network measurements: benefits and best practices, Comput. Netw. (2015). http://dx.doi.org/10.1016/j.comnet.2015.07.003. http://www.sciencedirect.com/science/article/pii/S1389128615002236.

[22] A. Rafetseder, F. Metzger, L. Pühringer, Y. Zhuang, J. Cappos, Sensorium a generic sensor framework, PIK - Praxis der Informationsverarbeitung und Kommunikation 36 (1) (2013) 46, doi:10.1515/pik-2012-0061.

[23] C. Müller, S. Lederer, C. Timmerer, An evaluation of dynamic adaptive streaming over HTTP in vehicular environments, in: Proceedings of the 4th Workshop on Mobile Video, in: MoVid '12, ACM, New York, NY, USA, 2012, pp. 37–42, doi:10.1145/2151677.2151686. http://doi.acm.org/10.1145/2151677.2151686.

[24] M. Vranješ, T. Švedek, S. Rimac-Drlje, The use of ns-2 simulator in studying umts performances, Int. J. Electr. Comput. Eng. Syst. 1 (2) (2011) 63–71.

[25] C. Mehlführer, J. Colom Colom Ikuno, M. Å imko, S. Schwarz, M. Wrulich, M. Rupp, The Vienna LTE simulators - enabling reproducibility in wireless communications research, EURASIP J. Adv. Signal Process. 2011 (1) (2011), doi:10.1186/1687-6180-2011-29. http://dx.doi.org/10.1186/1687-6180-2011-29.

[26] G. Piro, L. Grieco, G. Boggia, F. Capozzi, P. Camarda, Simulating LTE Cellular Systems: An Open-Source Framework, IEEE Trans. Veh. Technol. 60 (2) (2011) 498–513.

[27] N. Baldo, M. Requena-Esteso, M. Miozzo, R. Kwan, An open source model for the simulation of LTE handover scenarios and algorithms in ns-3, in: Proceedings of the 16th ACM International Conference on Modeling, Analysis: Simulation of Wireless and Mobile Systems, in: MSWiM '13, ACM, New York, NY, USA, 2013, pp. 289–298, doi:10.1145/2507924.2507940. http://doi.acm.org/10.1145/2507924.2507940.

[28] T. Hoßfeld, M. Seufert, C. Sieber, T. Zinner, P. Tran-Gia, Close to optimum? PIK - Praxis der Informationsverarbeitung und Kommunikation 37 (4) (2014) 275–285. http://dx.doi.org/10.1515/pik-2014-0029. http://www.degruyter.com/view/j/piko.2014.37.issue-4/pik-2014-0029/pik-2014-0029.xml.

[29] C. Sieber, T. Hoßfeld, T. Zinner, P. Tran-Gia, C. Timmerer, Implementation and user-centric comparison of a novel adaptation logic for DASH with SVC, in: IEEE International Symposium on Integrated Network Management (IM 2013), 2013 IFIP, 2013, pp. 1318–1323.

[30] T. Hoßfeld, M. Seufert, C. Sieber, T. Zinner, P. Tran-Gia, Identifying QoE optimal adaptation of HTTP adaptive streaming based on subjective studies, Comput. Netw. (2015). http://dx.doi.org/10.1016/j.comnet.2015.02.015.

[31] J. Gettys, K. Nichols, Bufferbloat: Dark buffers in the internet, Queue 9 (11) (2011) 40. http://www.sciencedirect.com/science/article/pii/S1389128615000626.

[32] H.-S. Park, J.-Y. Lee, B.-C. Kim, TCP performance degradation of in-sequence delivery in LTE link layer, Int. J. Adv. Sci. Technol. 37 (2011) 27–36.

[33] P. Reichl, S. Egger, S. Möller, K. Kilkki, M. Fiedler, T. Hoßfeld, C. Tsiaras, A. Asrese, Towards a comprehensive framework for QoE and user behavior modelling, in: Proc. 7th International Workshop on Quality of Multimedia Experience (QoMEX 15), Greece, 2015.

[34] A. Sarma, S. Chakraborty, S. Nandi, Context aware handover management: Sustaining QoS and QoE in a public IEEE 802.11e hotspot, IEEE Trans. Netw. Serv. Manage. 11 (4) (2014) 530–543, doi:10.1109/TNSM.2014.2352811.

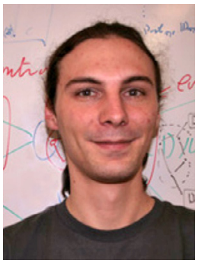
[35] Y. Zhu, I. Heynderickx, J.A. Redi, Understanding the role of social context and user factors in video quality of experience, Comput. Hum. Behav. 49 (2015) 412–426. http://dx.doi.org/10.1016/j.chb.2015.02.054.

[36] K. Mitra, A. Zaslavsky, C. Åhlund, Context-aware QoE modelling, measurement and prediction in mobile computing systems, IEEE Trans. Mobile Comput. 14 (5) (2015) 920–936, doi:10.1109/TMC.2013.155.

[37] T. Hoßfeld, L. Skorin-Kapov, Y. Haddad, P. Pocta, V. Siris, A. Zgank, H. Melvin, Can context monitoring improve QoE? a case study of video flash crowds in the internet of services, in: IEEE International Symposium on Integrated Network Management (IM), 2015 IFIP, 2015, pp. 1274–1277, doi:10.1109/INM.2015.7140480.

[38] T. Hosfeld, C. Moldovan, C. Schwartz, To each according to his needs: dimensioning video buffer for specific user profiles and behavior, in: IEEE International Symposium on Integrated Network Management (IM), 2015 IFIP, 2015, pp. 1249–1254, doi:10.1109/INM.2015.7140476.

[39] T. Hoßfeld, M. Seufert, C. Sieber, T. Zinner, Assessing effect sizes of influence factors towards a QoE model for HTTP adaptive streaming, in: Sixth International Workshop on Quality of Multimedia Experience (QoMEX), 2014, 2014, pp. 111–116, doi:10.1109/QoMEX.2014.6982305.



Florian Metzger received his doctorate degree in computer science at the University of Vienna, Austria, in 2015. Between 2010 and 2015 he was a researcher in the Future Communication group at the University of Vienna, Austria. Since 2015 he is a postdoctoral researcher at the Chair "Modeling of Adaptive Systems" at the University of Duisburg-Essen, Germany. His research interests include video streaming in mobile networks, video game modeling, as well as computer network transport protocols.

Eirini Liotou received the Diploma in Electrical & Computer Engineering from the National Technical University of Athens and the MSc in Communications & Signal Processing from the Imperial College of London. She has worked as a Senior Software Engineer in Siemens Enterprise Communications within the R&D department. Since 2013 she is a PhD candidate in the Department of Informatics & Telecommunications in the National & Kapodistrian University of Athens, working on QoE provisioning in 4G/5G cellular networks.

Christian Moldovan received his Master's degree in Computer Science at the University of Würzburg, Germany, in 2014. He is currently a PhD candidate at the Chair "Modeling of Adaptive Systems" at the University of Duisburg-Essen, Germany. He is working on analytical models and user engagement metrics for video streaming services.

Tobias Hoßfeld is professor and head of the Chair "Modeling of Adaptive Systems" at the University of Duisburg-Essen, Germany, since 2014. He finished his PhD in 2009 and his professorial thesis (habilitation) "Modeling and Analysis of Internet Applications and Services" in 2013 at the University of Würzburg, Chair of Communication Networks, where he was also heading the "Future Internet Applications & Overlays" research group. He has published more than 100 research papers in major conferences and journals, receiving 5 best conference paper awards, 3 awards for his PhD thesis, and the Fred W. Ellersick Prize 2013 (IEEE Communications Society) for one of his articles on QoE. Tobias is organizer of several Dagstuhl seminars and TPC Co-Chair of several conferences including ITC 2016. He is member of the advisory board of the ITC conference and the editorial board of International Journal of Electronics and Communications as well as IEEE Communications Surveys & Tutorials. Homepage: http://www.mas.wiwi.uni-due.de/

Please cite this article as: F. Metzger et al., TCP video streaming and mobile networks: Not a love story, but better with context, Computer Networks (2016), http://dx.doi.org/10.1016/j.comnet.2016.06.033