Mobile Video Delivery: Challenges and Opportunities

Rajesh Krishna Panta • AT&T Labs Research

Demand for mobile videos is significantly on the rise, yet network infrastructure resources that deliver mobile videos haven’t increased proportionally. In exploring the challenges and solutions for mobile video delivery, the author also considers the impact of virtualizing network functions and services.

Video constitutes a large fraction of the Internet traffic today. Globally, IP video traffic will be 79 percent of all Internet traffic in 2018, excluding video shared through peer-to-peer file sharing. In the last few years, we’ve seen a tremendous increase in cellular data traffic, fueled primarily by the surge in mobile video traffic. We can mostly attribute the tremendous growth of mobile video traffic to the popularity of smart handheld devices like smartphones and tablets, and the ease with which we can generate and consume videos using mobile devices.

This ever-increasing demand has introduced several challenges in delivering videos with a good quality of experience to the user. Although these challenges exist in both wired and wireless environments, the problem is especially severe in cellular mobile contexts due to several factors: for example, the limited wireless spectrum and bandwidth, time- and location-dependent wireless link characteristics, radio congestion, potential handoff issues, heterogeneous device features and limitations, and so on. Although network operators have made huge investments in recent years to address this problem, the increase in network resources hasn’t kept pace with the surge in mobile traffic demand. As a result, the quality of the user’s experience for mobile video is of critical concern today.

With this in mind, here I explore several techniques for efficient mobile video delivery. I also consider why current systems might be inadequate in the long run. Finally, I discuss the recent trend of virtualizing mobility core (the network that connects the cellular network to the IP network), which can significantly impact mobile video delivery. The focus of this article is on mobile video streaming, not video downloads, because most of the popular video services like YouTube and Netflix use streaming.

Mobile Video Streaming Background

Existing popular video streaming services use an HTTP-based over-the-top method to deliver video content to users. Early versions of such services pushed video content to the client devices as quickly as possible using TCP at a constant bit rate, irrespective of network conditions and device capabilities. However, network conditions can vary greatly for different clients. Even for a single client, the available network bandwidth can vary over time. Furthermore, because of heterogeneous device capabilities, the quality of video a client can play varies significantly among client devices.

These problems are even more pronounced in wireless cellular environments because of several factors, including dynamic radio environment, device mobility, and a wide range of device capabilities. Thus, most of the streaming services today support streaming videos at variable bit rates using various adaptive bit-rate (ABR) technologies such as Dynamic Adaptive Streaming over HTTP (DASH), Apple HTTP Live Streaming (HLS), Microsoft Individualized-Integrated Book (IIB) Smooth Streaming, and Adobe HTTP Dynamic Streaming. In these ABR schemes, the source video is divided into smaller chunks and multiple versions of each chunk are pre-encoded at different bit rates on the streaming server. The video player running...
Challenges and Opportunities

Now that we have this background, let’s discuss some important challenges for efficient mobile video delivery. We consider some potential solutions as well.

Rate Adaptation and Scheduling

As previously mentioned, the fundamental problem in ABR-based mobile video streaming is the inaccurate estimation of network bandwidth. There are many factors that make network bandwidth estimation challenging in mobile networks, such as the time and location varying network link conditions, device mobility, congestion in the wireless medium, random device arrival/departure, and so on. The most commonly used methods for estimating network bandwidth are based on the observed TCP throughput for downloading previous video chunks. Some recent research studies propose the use of other parameters, such as the video player’s buffer occupancy at the client as an indicator of network conditions. The idea is based on the assumption that the rate at which the video buffer is filled represents approximately the current available network bandwidth.

The fundamental problem with these approaches is that in wireless environments, the available network capacity can vary so wildly that any prediction based on historical information might not always yield accurate bandwidth estimation.

Another fundamental problem of ABR-based video streaming is that when a cellular network link is shared by multiple ABR flows, the temporal overlap of the chunks of different flows might cause under- or over-estimation of the network bandwidth. In cellular networks, the base station schedules flows of different users to achieve some notion of fair resource allocation across end user devices in a cell. However, these schedulers are designed for single bit-rate videos and they can’t handle multiple bit rates efficiently. As a result, sharing a cellular network link among multiple ABR flows of varying bit rates results in inefficient use of radio resources, unfair resource allocation among concurrent video streaming flows, and instability of bit-rate selection. By separating the scheduling algorithm of ABR video flows from other non-ABR video flows (and non-video flows), and by designing the scheduler optimized for multiple ABR flows, we can achieve better resource utilization and fair scheduling among such flows.

This discussion suggests that different schedulers customized for different types of data flows could be necessary to ensure better performance for different classes of data traffic. It’s also reasonable to assume that various types of information available at the network might help solve (or mitigate) several of the aforementioned problems for mobile video streaming. For example, in current approaches, mobile clients use local information (such as TCP throughput and buffer occupancy) alone to make future bandwidth estimation. However, if the information available in the cellular network (for example, congestion level in a cell or the total number of active ABR flows in cell) is somehow communicated to mobile devices, they might be able to make better bandwidth estimates. If the network provides some APIs to share this information with video streaming services (and other applications), then those services are equipped better to tackle many of mobile video streaming’s fundamental problems. Furthermore, each video service might decide to implement its own customized scheduler, policy engines, and data-processing algorithms, based on their requirements and capabilities. Later, we discuss why this is difficult in today’s environment and how the trend towards virtualizing mobility core can help in achieving these goals.

Demand-Supply Mismatch

Another critical issue in mobile video delivery is the mismatch between

on the client device adaptively selects the appropriate bit rate based on current network conditions and device capabilities. The client uses a prediction algorithm to estimate the network bandwidth that will be available to download the next video chunk, based on the observed TCP throughput for previous video chunks. Each video chunk is typically stored as a regular file on the video server and downloaded using standard HTTP GET requests. In short, the ABR mechanism’s goal is to maximize the video quality by choosing the highest bit rate the network can support without causing video pauses for the client.

If network conditions are stable, the device can make fairly accurate estimates of future capacity based on past TCP throughput observations. However, in wireless cellular environments, network conditions can vary widely over time. Mobility complicates the problem further, since available network capacity can change significantly as a function of the user location— if a user moves away from the base station, the available bandwidth can degrade significantly. Several studies show that the throughput observed during a single video streaming session can vary wildly. For example, even within a single Netflix session, the measured throughput varies from 500 Kbits/s to 17 Mbits/s. Thus, estimating future network capacity is challenging in mobile video streaming. Inaccurate estimates can lead to degraded quality of experience for the end users. If network capacity is underestimated, the user will receive the video with lower quality, even though the current network condition allows a higher quality of video to be delivered to the user. If the ABR mechanism overestimates the network capacity, the video player picks a video bit rate greater than network capacity. As a result, the video plays back faster than the rate at which it’s downloaded. This ultimately leads to video buffer depletion, causing video pauses.
Mobile video demand and the network resources available to meet such demands. Although mobile video demand has been increasing tremendously in recent years, the capacity of cellular networks, especially the wireless spectrum, hasn’t increased proportionally. One way to tackle this problem is to increase network resources by, for example, using small cells to augment the capacity of traditional macro cells, adding WiFi hotspots to offload cellular traffic to WiFi, and using portable base stations (such as Cell On Wheels or COWs) in crowded areas for some time periods. These approaches are expensive, though, and they might be insufficient to handle the explosive growth of video traffic.

My colleagues and I came up with a recent proposal, CoAST, that attempts to solve this problem by using two key insights. First, the mobile data traffic exhibits high burstiness over small time scales (tens of seconds). Thus, to ensure adequate quality of service at all times, it’s important to reduce instantaneous peak traffic, not just aggregate traffic. Second, although it might not seem intuitive at first glance, the video streaming clients can tolerate delays of tens of seconds (as long as the playback buffer isn’t empty) without affecting the quality of video playback. These two insights suggest that if the right video flow (from the set of all video flows in a cell) is delayed at the right time for the right time duration, it’s possible to reduce peak traffic in a cell without affecting the user experience on any mobile device. However, this requires both device-level information (for example, delay tolerance values at the given time instant) and cell-level information (for instance, the total traffic demand in the cell at a given time instant). CoAST provides an efficient collaboration framework for mobile devices and a cellular network to exchange such information and make proper decisions about delaying video streaming traffic. CoAST reduces traffic peaks in a cell by up to 50 percent and eliminates some pauses in video streaming.

Our previous work shows another example of how the collaboration between mobile devices and cellular infrastructure can help improve the performance of video streaming. Cellular networks possess rich information about mobile devices’ location, mobility patterns, and possible geographic regions where network link qualities might be degraded. By analyzing this information, the network can predict and inform a mobile device running a video streaming application that it’s likely to enter a congested region in the near future. Upon receiving such notification, the mobile device can aggressively download the video content. As a result, when the mobile device actually enters the congested region, it will have sufficient video content in its playback buffer to avoid pauses.

These two examples highlight the importance of cooperation between mobile devices and network infrastructure to improve the performance of video streaming in cellular networks. And of course, there are many other examples that could benefit from such collaboration. As we discuss later, the trend of virtualizing network functions can facilitate the development of such collaborative frameworks.

**Multicast Videos**
Multicast is another way to improve the performance of mobile video delivery in some cases. If a group of users in a given area are interested in the same video at the same time, it’s more efficient in terms of resource usage to broadcast/multicast a single (or a limited number of) video flow to all the users, instead of transmitting individual unicast streams to each mobile device. Enhanced Multimedia Broadcast Multicast Service (eMBMS) is the 3GPP (Third Generation Partnership Project) standard specification for multicast over Long-Term Evolution (LTE) networks. The multicast feature is especially useful in scenarios such as live video streaming for broadcasting sporting events, concerts, or other special events. There are several technical challenges for proper implementation of the multicast service. For example, how should we allocate network resources among unicast and multicast services? Because different users in a cell can experience different radio link conditions, how should we organize users in multicast groups so that mobile devices can receive video with quality commensurate with their wireless link bandwidth? These issues suggest that we might need customized schedulers and distribution algorithms for multicast videos, further strengthening the case for virtualized mobility core.

**The Effect of Network Virtualization**
In recent years, there’s a trend in the telecommunication industry towards an important transformation — virtualizing network functions and services. The idea is to replace the dedicated network appliances that perform specific network functions with general-purpose commodity hardware (such as x86-based machines), where the network functions are implemented in software. The advantages of such a transformation include a reduction in capital and operational costs, shortening the time required to deploy new networking services to support changing business requirements, and the ability to scale up or down services to address changing demands.

This trend has important implications for mobile video streaming as well. From the discussion in previous sections, it’s clear that improving video delivery over cellular networks might require customized solutions for different scenarios. We might need highly customized schedulers and policy decisions for different video services. The collaboration mechanism between mobile devices and the
cellular network might be different for different video streaming services (and other non-video applications). Multicast solutions may require yet another set of resource allocation modules. It’s difficult to incorporate these highly individualized solutions into today’s appliance-based network elements, which afford little flexibility.

Virtualizing network functions enables the development of a software platform that multiple video streaming (or non-video) applications can share, and then the applications can implement their own high-level network functions according to their requirements and capabilities. In addition to the ability to deploy customized solutions for individual streaming services, the virtualized mobility core also allows allocation of resources for each application based on their current load. Efficient resource utilization is important for network service providers because it helps to tackle the mismatch between high video demand and limited resources available to satisfy such demands. For example, a video streaming service can independently scale out its network resources when it experiences a burst in the mobile video demand. Although this subsection focuses on how virtualization can help video delivery in mobile networks, we should note that many other applications could also benefit from the virtualized platform.

One important question regarding network function virtualization is: Can network functions implemented in software meet critical performance requirements, such as latency and throughput? It turns out that recent technological developments like Intel’s Data Plane Development Kit (DPDK) help narrow the performance gap between hardware and software implementations.

Even though we’ve come to expect high-quality videos on mobile devices, the increased demand for mobile videos means that creating such an experience for users is challenging. Hopefully some of the techniques mentioned here (customized scheduling at the network edge, and collaboration between the network and mobile devices) can diminish these challenges. The important features of network function virtualization also show potential in providing a software platform to build resource-aware and customized solutions for mobile video streaming services.

References


Rajesh Krishna Panta is a member of the technical staff at AT&T Labs Research. His research interests include cloud computing, Big Data, storage systems, mobile systems, wireless networks, and wireless sensor networks. Panta has a PhD in electrical and computer engineering from Purdue University. Contact him at rpanta@research.att.com.

Selected CS articles and columns are also available for free at http://ComputingNow.computer.org.