Performance of DASH and WebRTC Video Services for Mobile Users

Fraida Fund*, Cong Wang†, Yong Liu*, Thanasis Korakis*, Michael Zink†, Shivendra S. Panwar*
*Department of Electrical and Computer Engineering
Polytechnic Institute of New York University
ffund@nyu.edu, {yongliu,korakis}@poly.edu, panwar@catt.poly.edu
†Department of Electrical and Computer Engineering
University of Massachusetts Amherst
{cwang, zink}@ecs.umass.edu

***Abstract*—With the confluence of the growing market for mobile Internet devices, and users' expectations of instant access to high-quality multimedia content, the delivery of video over wireless networks has become the challenge of the decade. Dynamic Adaptive Streaming over HTTP (DASH) and WebRTC are new and evolving standards that have been developed specifically to meet this demand and enable a high-quality experience for mobile users of video on demand and real time communication services, respectively. However, there has been no systematic study of how these services are experienced by users in a realistic mobile setting. In this work, we describe measurements collected from DASH and WebRTC implementations while moving at walking speeds through an 802.16e WiMAX network. Using data from the application, network, and physical layers, in different wireless environments, we identify characteristics of the cellular data network that directly impact the quality of video service, and suggest areas for further improvement.**

I. Introduction

Over the last few years, mobile video has become an essential Internet service. The increasing ubiquity of smartphones, tablets, and other mobile devices, together with the growth of media-intensive Internet applications that drive video usage, ensures that this trend will continue.

In reaction to this trend, various industry bodies have been working to standardize the delivery of mobile video. In particular, Dynamic Adaptive Streaming over HTTP (DASH) [1] and WebRTC [2] are evolving standards that have been developed specifically to meet this demand and enable a high-quality experience for mobile users of video on demand and real time communication services, respectively. Both are designed to adapt to dynamic link characteristics, to deliver the best possible video quality in any scenario.

DASH is an MPEG standard that has been adopted by the 3GPP, and is closely related to other multi-rate segmented HTTP streaming services. DASH and related technologies are already in widespread use in many commercial applications for delivering video on demand (VOD) content.

WebRTC is a World Wide Web Consortium (W3C) standard for browser-based voice, video, and peer-to-peer data, and is under development with support from Google, Mozilla, and Opera. It specifies a standard Javascript API for enabling native voice and video communication in the browser. The Internet Engineering Task Force (IETF) RTCWEB [3] effort is a closely related working group tasked with defining protocols for transporting these real-time communication flows. A diverse ecosystem of web applications have grown up around the WebRTC standard in the last year.

Meanwhile, as demand for instant access to high-quality multimedia content grows, wireless carriers are racing to deploy upgraded networks that are better equipped to meet this demand. Serving wireless video is a significant challenge for already-stressed cellular data networks [4]. In addition to the high bandwidth requirements, video traffic imposes additional latency and packet loss constraints for acceptable service.

Yet despite widespread acknowledgment of the challenges associated with mobile video delivery, surprisingly little is known about how these directly affect consumers of mobile video. Although a great deal of effort has been concentrated on quantifying the aggregate effect of mobile video on core and access networks, there has been no study of how mobile video services - especially those based on new standards such as DASH and WebRTC - are experienced by individual pedestrian mobile users. In particular, the relationship between network metrics and metrics that are directly observed by the end user (e.g., frame rate) is not fully understood. This knowledge is essential to help inform the creators of these standards, as well as the developers that build on them.

In this work, we collect measurements from DASH and WebRTC implementations while moving at walking speeds through an 802.16e WiMAX network. Using measurements from the application, network, and wireless physical layers, we identify characteristics of the cellular data network that directly impact the quality of video service, and suggest areas for further improvement and optimization.

Our contributions are, therefore, the following:

- Empirical measurements quantifying key characteristics of network and video performance experienced by a pedestrian user in a realistic mobile setting. We describe measurements of wireless link quality, video bitrate, video frame rate, video frame size, packet loss, and other relevant statistics, for DASH and WebRTC services.
- Identification of behaviors in DASH and WebRTC implementations whose interaction with the wireless environment negatively impacts video quality. By noting these

behaviors, we hope to distinguish areas for improvement in these evolving Internet video standards.

The rest of this paper is organized as follows. First, in Section II, we place this project in the context of related work. In Section III and Section IV, we describe the measurement platform used and the wireless environment in which this experiment was conducted. Measurements from DASH and WebRTC are described in Section V and Section VI, respectively. Section VII concludes with directions for future work.

II. Related Work

This paper complements a growing body of related studies that describe enhancements to DASH and WebRTC.

A number of studies have suggested enhancements to DASH, including novel rate adaptation algorithms [5]–[7], methods for determining optimal segment duration [8], extensions to SVC video [9]–[11], peer-assisted delivery of DASH video [12], and the addition of QoE-aware proxies in the content delivery network to improve DASH performance [13]–[15]. Others have considered the problem of delivering DASH video at an acceptable quality over wireless networks, but with respect to aggregate cell performance rather than the experience of an individual user [16], [17].

Although WebRTC is a relatively new standard, its architecture and design has attracted some interest from the research community [18], [19]. In [20], the performance of the congestion control algorithm used in WebRTC was evaluated with respect to its ability to track available bandwidth and fairly share resources in an emulated WAN scenario. The general problem of rate adaptation for RTC applications in mobile networks has also been a recent subject of investigation [21].

These studies, many of which directly address performance problems described in this paper, generally rely on simulations or small case studies to verify the applicability of the proposed improvements. With the exception of a few studies [11], [22] which measure DASH in an emulated vehicular setting, we have not seen any work that describes the empirical user experience for current DASH and WebRTC implementations in a real mobile wireless setting. As users in this setting represent a major target demographic for these technologies, we seek to fill this gap with measurements collected "over-the-air" as a pedestrian user of a cellular data network.

III. Measurement Platform

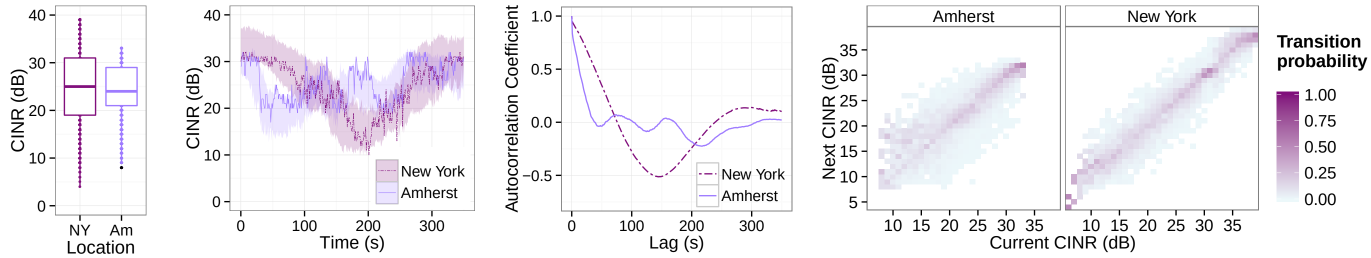
This work was conducted using dedicated experimental WiMAX 802.16e networks installed at the campuses of the Polytechnic Institute of NYU (NYU-Poly) and the University of Massachusetts Amherst (UMass Amherst). Each installation includes a commercial WiMAX base station (BS) operating in a licensed frequency band, as well as other components required to route traffic from WiMAX clients to the Internet or other networks. For highly controlled experimentation, this platform has a distinct advantage over commercial cellular networks because it allows us to isolate the effects of the wireless channel quality from other variables such as competing traffic, carrier routing and shaping policies, and radio configuration. This increases consistency and repeatability, while still being more true to life than a simulation or emulation environment.

In a previous study [23], we quantified the performance of this platform, particularly with regard to achievable data rates. We found that the data rates achieved on this platform are similar to typical data rates measured by others for users of commercial HSPA+ networks and users of commercial LTE networks [24]. This suggests that with regard to overall performance, this platform can be considered broadly representative of current wireless broadband networks.

The clients used in this study were commodity laptops equipped with a commercial WiMAX network adapter, a USB GPS dongle, and a webcam. On these, we installed modified versions of version 2.1.0 of the popular VLC media player, which includes a DASH plugin [25], and version 27.0.1453.65 of Google's Chrome browser for Linux, which has WebRTC capability. The software was instrumented using the OML measurement framework [26] to collect key network and video metrics from the application, and stream measurements to a local database during runtime. We also used logger applications to collect GPS coordinates from the *gpsd* daemon in Linux and wireless link characteristics from the WiMAX driver. The following measurements were collected:

- **VLC with DASH.** *VLC Input*: Decoded bitrate, decoded bytes read. *DASH Rate Adaptation*: Empirical bitrate from previous segment, bitrate to download for next segment, buffer status. *DASH HTTP Download*: Video segment ID, bytes read overall, bytes read for current segment, download time overall, download time for current segment.
- **WebRTC in Google Chrome.** *Audio Receiver*: Audio output level, bytes received, jitter received, packets received, packets lost. *Audio Sender*: Audio input level, bytes sent, packets received, RTT, jitter. *Video Receiver*: Bytes received, frame height received, frame rate decoded, frame rate received, frame width received, packets received, packets lost. *Video Sender*: Bytes sent, frame height sent, input frame rate, frame rate sent, frame width sent, packets sent, RTT. *Video Bandwidth Estimation*: Estimated available send bandwidth, estimated available receive bandwidth, target encoded bitrate, actual encoded bitrate, transmit bitrate.
- **WiMAX Logger.** Received signal strength indicator (RSSI), carrier to interference plus noise ratio (CINR), frequency.
- **GPS Logger.** Latitude, longitude, altitude, speed, time, fix mode (2D or 3D).

The DASH rate adaptation measurements are sampled each time the DASH client selects a rate for the next segment, and the HTTP download measurements are collected when the client downloads part of a segment. GPS measurements are collected whenever the GPS hardware has a report available. All other measurements are sampled at one second intervals.

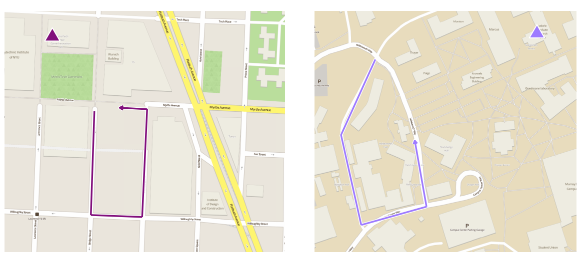


(a) (b) (c) (d)

Fig. 1: WiMAX link characteristics. Fig. 1(a) shows the distribution of CINR (signal quality) measurements at each location, with the horizontal lines in the boxplots corresponding to the mean, the upper and lower hinges to the first and third quartiles, and outliers beyond 1.5 interquartile range (IQR) of the hinges plotted as points. Fig. 1(b) shows CINR along each experiment path as a function of time for a single representative trial, with the shaded area indicating a range of one standard deviation above and below the mean CINR calculated across all trials. Fig. 1(c) gives the autocorrelation of CINR over a long timescale, while Fig. 1(d) shows CINR transitions over one second.

IV. Wireless Environment

Because the characteristics of a wireless channel are heavily dependent on the topography of an area, we collected measurements at two campuses in very different wireless settings. NYU-Poly is in an urban area composed mainly of high-rise commercial, civic, and residential buildings. It is a highly dynamic environment, with heavy moving vehicles and pedestrian traffic in the radio path. In contrast, the UMass Amherst campus is a suburban/semi-rural environment, with few tall buildings and little traffic.



(a) (b)

Fig. 2: Fig. 2(a) and Fig. 2(b) show the WiMAX BS locations and the mobility patterns followed in New York and Amherst, respectively. *Map data © OpenStreetMap contributors, tiles from skobbler.*

All of the measurements described in this paper were collected along a 400 m measurement path at each location, with ten sets of measurements collected at each path. The geographical layout of the paths are depicted in Fig. 2, and their wireless link characteristics are given in Fig. 1.

Fig. 1(a) and Fig. 1(b) show the distribution of CINR values measured over all trials and the CINR as a function of time along each experiment path. At both locations, a similar mean carrier to interference plus noise ratio (CINR) of approximately 25 dB is observed, although a slightly greater range of CINR values is observed in New York.

However, the channel behavior in time is quite different for the two wireless settings. Because the wireless signal propagates through "street canyons" in the urban area, the wireless signal tends to be very consistent when moving within a single such "canyon." In the suburban environment, where variations in signal strength are attributed to shadowing from individual buildings and obstructions in the signal path, more dramatic variations are observed over a short timescale.

This property is supported by Fig. 1(c), which shows the autocorrelation of each channel for lags up to 350 seconds. For the Amherst channel, the autocorrelation function closely resembles the widely reported exponential model. This intuitively represents the idea that locations that are close together are highly correlated, with the correlation decreasing gradually with distance. In New York, the autocorrelation function is shaped more like an exponential decaying sinusoid [27]. For the urban wireless channel, there is a strong correlation between measurements on a single block in the street grid, even though these may be separated by some tens of seconds, and virtually no correlation across intersections. This is reflected in the shape of the autocorrelation function, which shows a strong correlation between samples collected on the same block, and a sharp change in correlation coefficient at each intersection.

Over a short interval, the wireless channel is more consistent in the urban environment. This is demonstrated in Fig. 1(d), which shows the empirical probability of transitioning from one CINR value to another over a timescale of one second. In New York, two CINR values observed one second apart almost always differ by less than 5 dB, while in Amherst, transitions greater than 10 dB are seen quite often.

Put simply, we may state based on Fig. 1 that a period of especially poor or especially good signal quality is likely to be of short duration in Amherst, and of long duration in New York. We note these differences because the video applications under consideration use short-term estimates of network metrics to adjust bandwidth usage; the time variance of the channel is therefore highly relevant to performance.

V. DASH

Our procedure for measuring DASH performance is as follows. The DASH-enabled VLC player [25] is installed on a laptop, which connects to the local campus network through a WiMAX access network. An Apache server on campus serves the media presentation description (MPD) file and video segments. The link between this Apache server and the WiMAX network is a wired connection under low load, so that any network-related behaviors we observe may be attributed to the wireless link. We stream DASH video to the client while moving at walking speeds along the experiment path described in Section IV. The experiment is repeated ten times at each location, with consistent results.

For the video, we used the Big Buck Bunny animated video from the DASH dataset [28], with 2-second segments encoded at bitrates of 100, 200, 350, 500, 700, 900, 1100, 1300, 1600, 1900, 2300, 2800, 3400, 4500, and 6000 kbps. Because the VLC media player cannot currently play back H264/MP4 videos with dynamic resolution, we used video encoded at a constant resolution of 854x480p and a 24 fps frame rate. We kept the default maximum buffer size of 30 seconds (i.e., the buffer is considered full if it holds 30 seconds of video), so the client may smooth over temporary link disruptions. Similarly, our choice of 2-second segments was intended to offer flexibility for adapting to bandwidth fluctuations.

The DASH client in VLC follows a simple adaptation scheme which uses buffer state and bandwidth history to decide what bitrate to request for the next video segment. We call this a *maximum bitrate-low buffer avoidance* policy.

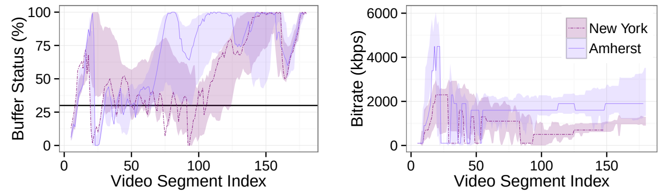
- If the video buffer is full, the client does nothing.
- If the video buffer is less than 30% full, the client requests next video segment at the lowest bitrate level, to avoid buffer depletion and a freeze in video.
- Otherwise, the client requests the next segment at the highest bitrate that is less than the empirical bitrate measured when downloading the previous segment.

DASH performance is known to be sensitive to the system parameters and the adaptation policy, a wide range of which exist in commercial systems. The reason we chose to evaluate this simple policy is twofold. First, we use the default settings in VLC and the default set of rates in the DASH dataset because it is likely that other researchers use these as a baseline against which to measure their proposed advancements. Therefore, it is useful to understand the performance of this scenario. Second, we seek to evaluate in a broad sense the general strategy of using measured data rate from a previous download as a major factor in selecting a future data rate. By using a very simple policy, we can examine the impact of this strategy component without obfuscation from other elements.

A. Buffer status

Fig. 3(a) shows the buffer status observed by the DASH client each time it chooses the next segment to download. At 100% buffer status, 30 seconds of video are in the buffer; at 0%, freezes in video playback will occur. The 30% point, at which the client drops to the lowest available bitrate to avoid buffer starvation, is marked by a horizontal line in Fig. 3(a).

The shape of the buffer status curve is a direct result of using past segment download times as a predictor of future segment download times. When the link quality is declining, the bandwidth is overestimated and the buffer level decreases, while the opposite effect occurs when the link quality is improving. Thus, the slope of the buffer status curve roughly tracks the slope of the CINR curve in Fig. 1(b).



(a) (b)

Fig. 3: Buffer status (Fig. 3(a)) and bitrate selected (Fig. 3(b)) each time a DASH video segment is retrieved. The line gives values for a representative trial, while the shaded region shows one standard deviation above and below the mean for all trials. In Fig. 3(a), the 30% point buffer state is marked by a horizontal line.

This may be an undesirable behavior, especially in an urban area where a given channel quality tends to persist for an extended period of time, since there is no chance for the buffer to recover during a prolonged period of poor signal quality. Indeed, in New York, the DASH client experiences buffer depletion during low CINR periods.

B. Video bitrate

The buffer status directly impacts the video bitrate selected by the DASH client for each segment, shown in Fig. 3(b). The mean bitrate across ten trials was 1003 kbps in New York and 1744 kbps in Amherst, with standard deviations across trials of 416.1 and 715.0 kbps, respectively.

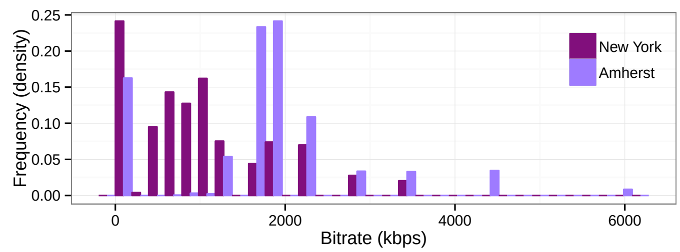


Fig. 4: Distribution of bitrates for DASH video segments.

Fig. 4 shows the frequency with which each bitrate is selected by the DASH client. At both locations, we observe a high proportion of instances where the smallest bitrate is used, due to the adaptation strategy of dropping to the lowest bitrate whenever the buffer is approaching starvation. These coincide with areas in Fig. 3(a) where buffer status drops below 30%.

C. Segment Download Time

For a DASH video client to avoid freezes, the average download time per segment should be less than the segment playing time. Fig. 5 shows the distribution of segment download times for our measurements with 2-second segments, with the mean download time given as the horizontal line in the boxplots, the upper and lower hinges corresponding to the first and third quartiles, and outliers beyond 1.5 interquartile range (IQR) of the hinges plotted as points. Overall, we measured a mean download time of 1.56 seconds per segment, with 75% of segments downloaded in less than 1.87 seconds. However, the outliers - though rare - include segment download times as high as 125 seconds, especially in the urban setting where signal interruptions tend to be of long duration. Because the client uses sequential HTTP downloading, long download times block the downloading of future segments, which often causes playback to freeze (depending on buffer status).

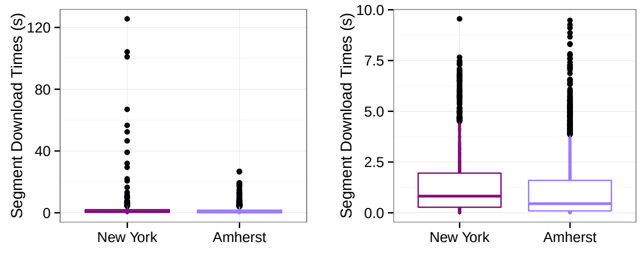


Fig. 5: The distribution of segment download times is shown in Fig. 5(a). Fig. 5(b) is a zoomed-in version of the same data, showing only download times that are shorter than ten seconds.

It is worth noting that as the segment download time increases, the bandwidth measured over the duration of the download becomes less representative of the immediate past, and therefore less predictive of the bandwidth for the next download period. The extent of this relationship is governed by the autocorrelation of the wireless channel, shown in Fig. 1(c). Thus long segment download times (as in the urban network) tend to make rate-based adaptation logic less effective. Long download times (those above the third quartile) were observed to appear in clusters because of this effect.

D. Video playback freezes

A key metric for video streaming performance is the frequency and duration of freezes in video playback. We count video freezes lasting four seconds or longer by monitoring the *decoded bytes read* metric reported by VLC, which increases continuously as long as the video is playing. For this metric, the behavior of the DASH client is very different depending on the wireless setting. In Amherst, on average, 0.5 interruptions (lasting between 4 and 6 seconds) were observed during the 375-second playback period. In New York, for the same playback period, an average of 2.3 interruptions lasting four seconds or more were observed, with some as long as 68 seconds. Fig. 6 shows the position and duration of each interruption in a representative experiment run.

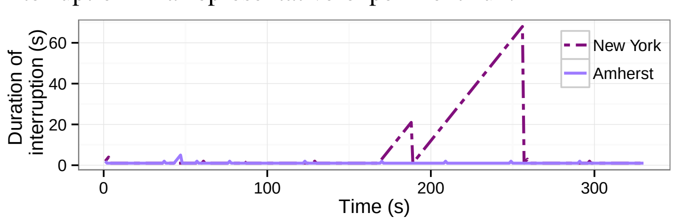


Fig. 6: The position and duration of each interruption in a representative DASH trial in New York and in Amherst are shown here. Note that the measurements described here are recorded every second, while the buffer status measurements in 3(a) are only sampled at the beginning of a segment download.

The extended freezes in New York correspond to the occasional long segment download times (Fig. 5) and prolonged periods of buffer depletion (Fig. 3(a)). In Amherst, there are no extended freezes because the buffer is depleted only for short intervals before rebounding.

E. Discussion

We saw that the performance of an adaptive video delivery strategy in a mobile setting depends on the detailed behavior of the wireless network in time. Despite the similarity of the average channel conditions in New York and Amherst, the different dynamics of signal propagation in the urban environment lead to significantly poorer DASH performance.

The mechanism behind this is described in Section V-A, where we saw that an overreliance on past download times for rate selection causes the buffer level curve to track the slope of the channel quality. Compared to a policy aimed exclusively at keeping a constant buffer level, for example, the bandwidth estimation strategy is more likely to lead to buffer depletion in an environment that is subject to prolonged periods of poor link quality. In fact, we saw extended segment download times (Section V-C) leading to interruptions lasting more than 60 seconds (Section V-D), as well as a lower average bitrate overall (Section V-B), in the urban setting.

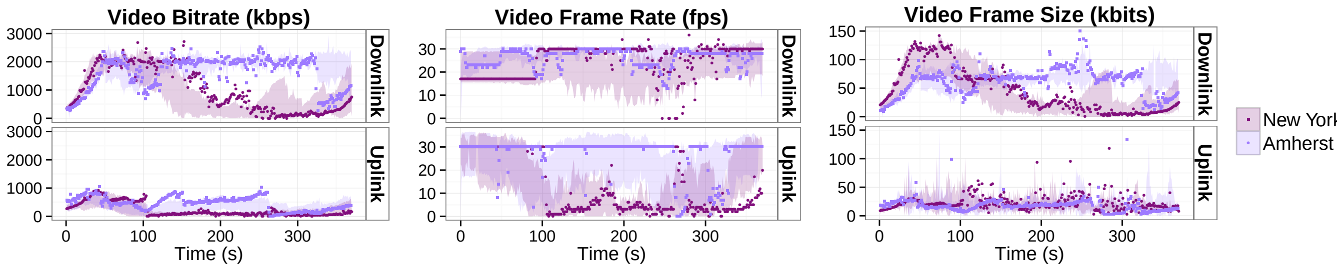
Several particular performance issues that we describe would be ameliorated somewhat under the more complex download strategies in commercial DASH players. However, the general results we describe - the problem with overreliance on bandwidth estimation, and the understanding that minor aspects of link behavior can have dramatic effects - apply broadly to the design of adaptive video delivery protocols.

VI. WebRTC

To evaluate WebRTC, we use a WiMAX-equipped laptop, which is connected to the campus network through a WiMAX access network, to run a WebRTC video chat webapp in Google Chrome. At the same time, another laptop connected to the campus network across an Ethernet link runs the WebRTC webapp in Google Chrome and initiates a video chat session with the WiMAX-connected client. The WiMAX-connected client moves at walking speeds along the experiment path described in Section IV during the video chat, while the Ethernet-connected client remains static. Each client streams video and audio from its webcam and microphone, and receives video and audio streams from its peer.

In real-time video communications, video quality may be constrained by the client hardware. The webcams used in our experiment have a maximum resolution of 640x480p, and the laptops have sufficient computing power to encode video at this resolution at a rate of 30 fps.

Because WebRTC video and audio quality have not been formally measured in a mobile setting, we are interested in gaining a general understanding of its performance. We also want to investigate whether WebRTC accurately adapts to the available bandwidth and uses the wireless channel effectively. Like DASH, WebRTC is designed to adapt to different network conditions. In the case of WebRTC, which uses UDP as the transport layer for audio and video streams, a key goal is to deliver video that is as high in quality as the network



(a) (b) (c)

Fig. 7: WebRTC video performance. For each plot, the shaded area shows one standard deviation above and below the mean, while the points give values for a representative trial. These values are presented from the perspective of the mobile, WiMAX-connected peer.

may support, without disrupting competing multimedia and web traffic. The WebRTC congestion control and bandwidth estimation protocols have been designed very carefully with these goals in mind [20], [29].

WebRTC adapts to changing link conditions by estimating the available send bandwidth and passing this estimate to the encoder as a target bitrate, using a protocol based on [30]. In the particular implementation used in this study, the sender available bandwidth estimate at time i, $A_s(i)$, is calculated according to the packet loss ratio, p, as follows:

$$A_s(i) = \begin{cases} 1.08\,A_s(i-1) + 1000 & \text{if } p \leq 0.02 \\ A_s(i-1) & \text{if } 0.02 < p \leq 0.10 \\ A_s(i-1)(1-0.5p) & \text{if } p > 0.10 \end{cases}$$

This estimate is further limited by the bandwidth calculated by the TCP Friendly Rate Control formula [31], and by the receiver-side estimate of available bandwidth, $A_r(i)$, with $A_{TFRC}(i) \leq A_s(i) \leq A_r(i)$. The receiver-side estimate of available bandwidth, $A_r(i)$, is calculated according to the system state. During the *increase* state, $A_r(i)$ is increased by a factor which is a function of the round trip time and the estimated measurement noise variance. When overuse of the link is detected, the system enters a *decrease* state in which $A_r(i) = 0.9\,R(i)$, where $R(i)$ is the incoming bitrate. If underuse is detected, $A_r(i)$ is kept constant for some time in a *hold* state before returning to the increase state.

A. Performance overview

Fig. 7 and Fig. 8 show key metrics of video and audio quality for WebRTC sessions sustained over the WiMAX link under the mobility patterns described in Section IV. We note several areas of particular concern.

As in our DASH measurements, we see a dramatic difference in performance between the two locations, with the suburban setting consistently outperforming the urban setting. However, because the adaptation policy in WebRTC is much more complex than the DASH policy we used, we cannot easily determine the exact mechanism underlying this difference.

The uplink (UL) stream, which is directed from the WiMAX client to its Ethernet-connected peer, has significantly worse video performance. This may be attributed to the asymmetry of the cellular link, on which only 25% of radio resources are allocated to the UL. Also, because the mobile sender has

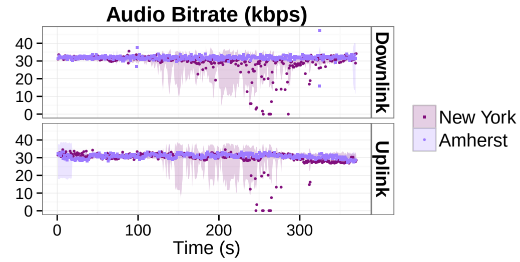


Fig. 8: WebRTC audio performance shown as representative values (points) and a range of one standard deviation above and below the mean (shaded area).

a much weaker transmitter than the BS, the UL channel is generally of poorer quality. In the context of mobile video, this constrains the bitrate of the video stream from the mobile user. This is especially unfortunate because the video feed from a mobile user tends to have high motion content, and is disproportionately affected when frame rate is reduced to meet a low target bitrate.

Unlike video, audio bitrate is not subject to rate adaptation in the current WebRTC implementation in Google Chrome. Thus, we observe that audio bitrate (Fig. 8) is mostly consistent at 30 kbps, degrading only in periods of extremely poor channel quality. This is in contrast to the video metrics (Fig. 7), which vary dramatically with channel quality.

B. Latency and loss

Key indicators of real-time communication quality include latency and packet loss. Table I and Table II give the average round trip time (RTT) and packet loss ratio reported by the WebRTC application, and the standard deviation across trials.

Round Trip Time (RTT)

Location	Stream	Mean (ms)	SD (ms)
New York	Audio	74.24	37.84
New York	Video	83.33	47.30
Amherst	Audio	65.01	32.36
Amherst	Video	63.84	28.26

TABLE I: Round trip time over WiMAX link for a WebRTC session.

While the latency over the wireless link is quite low, packet loss is high, especially in the urban setting and for the UL channel. Because there are no competing traffic flows affecting these experiments, we may conclude that the packet loss is entirely due to degradation of the wireless link. This symptom may be alleviated to some degree with link-layer

Packet Loss Ratio (Total Lost Packets / Total Packets Sent)				
Location	Direction	Stream	Mean	SD
New York	Downlink	Audio	0.075	0.013
New York	Downlink	Video	0.055	0.017
New York	Uplink	Audio	0.347	0.063
New York	Uplink	Video	0.094	0.022
Amherst	Downlink	Audio	0.002	0.001
Amherst	Downlink	Video	0.010	0.003
Amherst	Uplink	Audio	0.015	0.004
Amherst	Uplink	Video	0.019	0.001

TABLE II: Packet loss ratio for a WebRTC session.

retransmission, but this will increase latency and jitter.

We note that the packet loss for the New York UL audio channel is especially high. Because audio bitrate is not reduced when the channel is bad, a few short periods where almost 100% of packets are lost have a disproportionate effect on the overall packet loss calculation.

C. Wireless link usage

We do not know the actual data rate that is available to the WebRTC client at any given time, and so we cannot determine exactly how much of the wireless channel is consumed by the WebRTC feed. However, we can compare the performance of WebRTC and DASH with regard to wireless link usage. In the WiMAX network, DL and UL resources are allocated separately, in such a way that the UL usage in UDP applications (such as WebRTC) does not affect the DL performance. Therefore, we can directly compare the DASH download rate and the combined DL audio and video bitrate in WebRTC.

Audio Bitrate			
Location	Direction	Mean (kbps)	SD (kbps)
New York	Downlink	28.84	3.84
New York	Uplink	29.69	3.91
Amherst	Downlink	31.87	1.32
Amherst	Uplink	30.99	1.97

TABLE III: Summary statistics for audio bitrate in WebRTC.

Video Bitrate			
Location	Direction	Mean (kbps)	SD (kbps)
New York	Downlink	883.93	203.43
New York	Uplink	200.53	94.34
Amherst	Downlink	1738.55	665.66
Amherst	Uplink	360.41	56.22

TABLE IV: Summary statistics for video bitrate in WebRTC.

Table III and Table IV show the average and standard deviation across trials of the bitrate for WebRTC audio and video. Considering only the DL, the bitrate used by WebRTC is 9.0% lower than the DASH bitrate observed in New York and 1.5% higher than the DASH bitrate in Amherst. Thus, we see that in the urban setting, the DL WebRTC feed uses the link more conservatively than a TCP flow would. We attribute this to the higher packet loss observed in New York (Table II), which leads WebRTC to believe that the link is congested and scale back its usage accordingly. However, because the loss is due to channel degradation rather than congestion, reducing the target bitrate does not improve the packet loss ratio.

D. Video metrics

The changing target bitrate passed to the video encoder is apparent to the user in the fluctuation of frame rate (Fig. 7(b)) and frame size (Fig. 7(c)). (The video resolution remains constant at 640x480 pixels). From these figures we infer that the video encoder first reduces frame size; when this becomes very small, it then reduces the frame rate.

Video Frame Rate			
Location	Direction	Mean (fps)	SD (fps)
New York	Downlink	21.54	1.45
New York	Uplink	12.72	3.23
Amherst	Downlink	27.81	1.83
Amherst	Uplink	21.86	2.44

TABLE V: Summary statistics for video frame rate in WebRTC.

Video Frame Size			
Location	Direction	Mean (kbits)	SD (kbits)
New York	Downlink	37.98	13.97
New York	Uplink	21.45	9.24
Amherst	Downlink	64.06	32.93
Amherst	Uplink	21.07	13.26

TABLE VI: Summary statistics for video frame size in WebRTC.

Table V and Table VI give summary statistics for the frame rate and frame size, respectively.

Here, we see clearly a significant difference in user-facing metrics between the urban wireless network in New York and the suburban setting in Amherst, with the former having a 23% lower frame rate for the DL a 42% lower frame rate for the UL, and a 41% smaller frame size for the DL. We also see how the video transmitted from the mobile user is constrained to a low frame rate and small frame size due to the limited UL resources available on the wireless link.

E. Discussion

In our investigation of WebRTC, we saw that a heavy reliance on packet loss (among other factors) as an indicator of congestion leads to underutilization of the wireless channel and poor video quality, especially in a setting with high packet loss. This is despite the WebRTC protocols having been designed with careful consideration of their sensitivity to packet loss, and with regard to the benefits and disadvantages of techniques such as retransmission, forward error correction (FEC), and sending negative acknowledgments (NACK) [29]. (In the experiments described here, no FEC or NACK packets were observed.) We hope that this study will serve as an additional data point to help tune WebRTC to improve mobile performance in future versions.

We also confirmed that the uplink video from a mobile device is disproportionately affected, due to the asymmetry of the wireless link. As mobile video communications and mobile user-generated video content gain an increasing share of Internet traffic, this will become a greater point of concern.

VII. Conclusion

In this work, we examine the effect of the wireless network on video metrics affecting the service quality of an individual mobile user of a DASH or WebRTC video service. In particular, we identify characteristics of the cellular data network that directly impact the quality of video service. Although many of these issues are known to some extent, here we quantify for the first time their direct effect (in current implementations) on the service seen by an individual user in a real mobile scenario. We find that despite attempts to adapt to channel conditions, these services did not achieve acceptable service quality for mobile users under challenging network conditions.

As future work, we intend to investigate methods to alleviate some of the issues we have identified. This effort includes the development of video delivery techniques that adapt to wireless channel conditions using more information than just the observed data rate or packet loss (e.g., time behavior of the channel, or effect of congestion avoidance on packet loss).

ACKNOWLEDGMENTS

This work is supported by the New York State Center for Advanced Technology in Telecommunications (CATT) at the Polytechnic Institute of New York University, by NYU WIRELESS, and by the National Science Foundation under the Graduate Research Fellowship Program Grant No. 1104522 and under Grant No. CNS-0714770.

REFERENCES

[1] T. Stockhammer, P. Fröjdh, I. Sodagar, and S. Rhyu, "MPEG systems technologies - part 6: Dynamic adaptive streaming over HTTP (DASH)," *ISO/IEC, MPEG Draft International Standard*, 2011.

[2] Adam Bergkvist, Daniel C. Burnett, Cullen Jennings, and Anant Narayanan, "WebRTC 1.0: Real-time communication between browsers," World Wide Web Consortium, W3C Editor's Draft, Mar. 2013. [Online]. Available: http://dev.w3.org/2011/webrtc/editor/webrtc.html

[3] H. Alvestrand, "Overview: Real time protocols for brower-based applications," Internet Engineering Task Force, Internet Draft, Feb. 2013. [Online]. Available: http://tools.ietf.org/html/draft-ietf-rtcweb-overview-06

[4] J. Erman, A. Gerber, K. K. Ramadrishnan, S. Sen, and O. Spatscheck, "Over the top video: the gorilla in cellular networks," in *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference (IMC '11)*. ACM, 2011, pp. 127–136.

[5] C. Liu, I. Bouazizi, M. M. Hannuksela, and M. Gabbouj, "Rate adaptation for dynamic adaptive streaming over HTTP in content distribution network," *Signal Processing: Image Communication*, vol. 27, no. 4, pp. 288–311, Apr. 2012.

[6] K. Miller, E. Quacchio, G. Gennari, and A. Wolisz, "Adaptation algorithm for adaptive streaming over HTTP," in *Proceedings of the 19th International Packet Video Workshop (PV '12)*, 2012, pp. 173–178.

[7] G. Tian and Y. Liu, "Towards agile and smooth video adaptation in dynamic HTTP streaming," in *Proceedings of the 8th International Conference on Emerging Networking Experiments and Technologies (CoNEXT '12)*. New York, NY, USA: ACM, 2012, pp. 109–120.

[8] C. Liu, I. Bouazizi, and M. Gabbouj, "Segment duration for rate adaptation of adaptive HTTP streaming," in *Proceedings of the 2011 IEEE International Conference on Multimedia and Expo (ICME '11)*, 2011, pp. 1–4.

[9] T. Andelin, V. Chetty, D. Harbaugh, S. Warnick, and D. Zappala, "Quality selection for dynamic adaptive streaming over HTTP with scalable video coding," in *Proceedings of the 3rd ACM Multimedia Systems Conference (MMSys '12)*. ACM, 2012, pp. 149–154.

[10] Y. Snchez de la Fuente, T. Schierl, C. Hellge, T. Wiegand, D. Hong, D. De Vleeschauwer, W. Van Leekwijck, and Y. Le Loudec, "iDASH: improved dynamic adaptive streaming over HTTP using scalable video coding," in *Proceedings of the 2 ACM Multimedia Systems Conference (MMSys '11)*. ACM, 2011, pp. 257–264.

[11] C. Müller, D. Renzi, S. Lederer, S. Battista, and C. Timmerer, "Using scalable video coding for dynamic adaptive streaming over HTTP in mobile environments," in *Proceedings of the 20th European Signal Processing Conference (EUSIPCO '12)*, 2012, pp. 2208–2212.

[12] S. Lederer, C. Müller, and C. Timmerer, "Towards peer-assisted dynamic adaptive streaming over HTTP," in *Proceedings of the 19th International Packet Video Workshop (PV '12)*, 2012, pp. 161–166.

[13] N. Bouten, J. Famaey, S. Latre, R. Huysegems, B. Vleeschauwer, W. Leekwijck, and F. Turck, "QoE optimization through in-network quality adaptation for HTTP adaptive streaming," in *Proceedings of the 8th International Conference on Network and Service Management (CNSM '12)*, 2012, pp. 336–342.

[14] W. Pu, Z. Zou, and C. W. Chen, "Video adaptation proxy for wireless dynamic adaptive streaming over HTTP," in *Proceedings of the 19th International Packet Video Workshop (PV '12)*, 2012, pp. 65–70.

[15] R. K. P. Mok, X. Luo, E. W. W. Chan, and R. K. C. Chang, "QDASH: a QoE-aware DASH system," in *Proceedings of the 3rd ACM Multimedia Systems Conference (MMSys '12)*. ACM, 2012, pp. 11–22.

[16] O. Oyman and S. Singh, "On capacity-quality tradeoffs in HTTP adaptive streaming over LTE networks," in *Information Theory and Applications Workshop (ITA '12)*, 2012, pp. 54–55.

[17] S. Singh, O. Oyman, A. Papathanassiou, D. Chatterjee, and J. Andrews, "Video capacity and QoE enhancements over LTE," in *Proceedings of the 2012 IEEE International Conference on Communications (ICC '12)*, 2012, pp. 7071–7076.

[18] A. Amirante, T. Castaldi, L. Miniero, and S. Romano, "On the seamless interaction between webRTC browsers and SIP-based conferencing systems," *IEEE Communications Magazine*, vol. 51, no. 4, pp. 42–47, 2013.

[19] A. Johnston, J. Yoakum, and K. Singh, "Taking on WebRTC in an enterprise," *IEEE Communications Magazine*, vol. 51, no. 4, pp. 48–54, 2013.

[20] L. De Cicco, G. Carlucci, and S. Mascolo, "Experimental investigation of the google congestion control for real-time flows," in *Proceedings of the 2013 ACM SIGCOMM workshop on Future Human-Centric Multimedia Networking (FhMN '13)*. New York, NY, USA: ACM, 2013, pp. 21–26.

[21] Q. Xu, S. Mehrotra, Z. M. Mao, and J. Li, "PROTEUS: network performance forecast for real-time, interactive mobile applications," in *Proceedings of the 11th International Conference on Mobile Systems, Applications and Services (MobiSys '13)*, 2013.

[22] C. Müller, S. Lederer, and C. Timmerer, "An evaluation of dynamic adaptive streaming over HTTP in vehicular environments," in *Proceedings of the 4th Workshop on Mobile Video (MoVid '12)*. ACM, 2012, pp. 37–42.

[23] F. Fund, C. Wang, T. Korakis, M. Zink, and S. Panwar, "GENI WiMAX performance: Evaluation and comparison of two campus testbeds," in *Proceedings of the 2nd GENI Research and Educational Experiment Workshop (GREE '13)*, 2013.

[24] J. Huang, F. Qian, A. Gerber, Z. M. Mao, S. Sen, and O. Spatscheck, "A close examination of performance and power characteristics of 4G LTE networks," in *Proceedings of the 10th International Conference on Mobile Systems, Applications and Services (MobiSys '12)*, 2012, pp. 225–238.

[25] C. Müller and C. Timmerer, "A VLC media player plugin enabling dynamic adaptive streaming over HTTP," in *Proceedings of the 19th ACM International Conference on Multimedia (MM '11)*. ACM, 2011, pp. 723–726.

[26] O. Mehani, G. Jourjon, T. Rakotoarivelo, and M. Ott, "An instrumentation framework for the critical task of measurement collection in the future internet," NICTA, Eveleigh, Sydney, NSW, Australia, Tech. Rep., October 2012.

[27] Y. Zhang, J. Zhang, D. Dong, X. Nie, G. Liu, and P. Zhang, "A novel spatial autocorrelation model of shadow fading in urban macro environments," in *IEEE Global Telecommunications Conference (GLOBECOM '08)*, 2008, pp. 1–5.

[28] S. Lederer, C. Müller, and C. Timmerer, "Dynamic adaptive streaming over HTTP dataset," in *Proceedings of the 3rd ACM Multimedia Systems Conference (MMSys '12)*. ACM, 2012, pp. 89–94.

[29] S. Holmer, "On fairness, delay and signaling of different approaches to real-time congestion control," in *IAB / IRTF Workshop on Congestion Control for Interactive Real-Time Communication*, Vancouver, BC, Canada, Jul. 2012.

[30] H. Lundin, S. Holmer, and H. Alvestrand, "A Google congestion control for real-time communication on the world wide web," Google, Internet Draft, Sep. 2011. [Online]. Available: http://tools.ietf.org/html/draft-alvestrand-rtcweb-congestion-00

[31] M. Handley, S. Floyd, J. Padhye, and J. Widmer, "RFC 3448: TCP friendly rate control (TFRC): protocol specification," Internet Engineering Task Force, Tech. Rep., Jan. 2003.