Over the Top Video: The Gorilla in Cellular Networks

Jeffrey Erman, Alexandre Gerber, K.K. Ramakrishnan, Subhabrata Sen, Oliver Spatscheck

AT&T Labs–Research, New Jersey, USA

{erman,gerber,kkrama,sen,spatsch}@research.att.com

ABSTRACT

Cellular networks have witnessed tremendous traffic growth recently, fueled by smartphones, tablets and new high speed broadband cellular access technologies. A key application driving that growth is video streaming. Yet very little is known about the characteristics of this traffic class. In this paper, we examine video traffic generated by three million users across one of the world's largest 3G cellular networks. This first deep dive into cellular video streaming shows that HLS, an adaptive bitrate streaming protocol, accounts for one third of the streaming video traffic and that it is common to see changes in encoding bitrates within a session. We also observe that most of the content is streamed at less than 255 Kbps and that only 40% of the videos are fully downloaded. Another key finding is that there exists significant potential for caching to deliver this content.

Categories and Subject Descriptors

D.4.8 [**Performance**]: Measurements—*web caching*

General Terms

Networking Optimization

1. INTRODUCTION

Thanks to the emergence of user-friendly smartphones and tablets, cellular networks have recently experienced a phenomenal rise in data traffic. One US cellular operator observed a growth of 8000% over the last 4 years [4]. According to a network equipment manufacturer [5], strong growth will continue at a rate of 92% per year over the next 5 years, driven primarily by video traffic. Indeed, they estimate that video traffic accounts for half of the cellular traffic today and that this share will increase to two thirds of the traffic by 2015. Given such predictions, it becomes crucial for network providers to understand and then determine how to optimize delivery of video traffic for cellular networks. This traffic type is often referred to as Over The Top (OTT) video, as the content doesn't typically come from the cellular carrier, but from third party providers that leverage the Internet connectivity of cellular customers.

Unfortunately, little is known about the characteristics of cellular video traffic. For example, today there does not even exist a study on the popularity of the actual video streaming protocols used on cellular networks: previous studies have either looked at general cellular traffic usage or just focused on WiFi [10, 11, 13–16, 18, 19]. In cellular networks, the most constrained and expensive resource is the wireless spectrum in the Radio Access Network (RAN) and it is critical that the video delivery is optimized for this environment. A key step in that direction is developing a deep understanding of the video content. In particular, the frame size, encoding rate and other video parameters are important factors to better undstand, to evaluate how much optimization opportunity exists, identify the appropriate optimization techniques and where (content provider, cellular provider, user equipment) to implement them. For instance, the knowledge of the current encoding rates and video abandonment probabilities (how much of a video is likely to be watched) could suggest the need techniques such as video pacing or transcoding at the source servers or at in the network middleboxes.

While not as expensive, and, therefore less critical, backbone resources upstream of the RAN can also be optimized based on the characteristics of video traffic. Indeed, on the wireline Internet, [12, 17] have highlighted that 80% of multimedia streaming traffic was delivered over HTTP, and in this paper, we will show that the rate for cellular network is even higher with 98%. Hence, proxy caching based techniques which have been proposed for wired video distribution over HTTP might also be applicable to cellular networks and should be investigated.

This paper is the first study to provide answers to these types of questions about video traffic on a large cellular network. It is based on a data set collected early in 2011 covering approximately three million smartphones and tablets in the US over 48 hours. Some of the key takeaways of our analysis are as follows:

Protocol mix: Video traffic accounts for 30% of the downstream cellular traffic during the busy hour and a couple of streaming protocols running over HTTP dominate. HTTP Live Streaming (HLS) [2, 7] accounts for 36% of the video traffic, while progressive downloads (defined in Section 2) account for 60%.

Content Providers: 77% of the traffic is concentrated in just the top 10 content providers.

Bitrate Encoding: 80% of the video objects are encoded at low rates, at or below 255 kbps.

Video Abandonment: Most videos are downloaded partially. Only 40% of the video objects are completely downloaded.

Cacheability: There exists substantial potential for caching. 24% of the bytes for progressive downloads requests can be served from cache.

Table 1: Data set overview

Data Set	Location	Start time	Duration	Objects	Traffic (TB)
NE	US N/E	2/16/2011 15:00 GMT	24 hours	4.8M	8.09 TBytes
WEST	US West coast	2/16/2011 15:00 GMT	48 hours	5.0M	9.05 TBytes

2. METHODOLOGY

We categorize the videos downloaded over HTTP into three classes based on the streaming method used:

Progressive Download (PD): A single video is downloaded with a single HTTP request for the entire object from the client. The client can access the partially downloaded data before the download is complete.

PD with Byte-Range Requests (PD-byterange): A single video download involves multiple HTTP byte-range requests to get different portions of the content.

HTTP Live Streaming (HLS): This belongs to a broad class of protocols called Adaptive Bit Rate Video streaming [3]. The HLS protocol is a proposal to the IETF from Apple [2, 7]. In HLS, a video is downloaded in a series of chunks that are encoded using the MPEG4 H.264 Transport Stream. These chunks are typically around 3 to 10 seconds of video. They allow the stream to adapt to the changes in the network condition and also other factors, such as the device CPU load - by increasing or decreasing the bit rate and resolution of the video in real time when it requests the next chunk. This protocol is used generally by content providers that have long duration video content, such as Netflix or Hulu. Other examples of Adaptive Bit Rate Videos include HTTP Smooth Streaming advocated by Microsoft with Silverlight and HTTP Dynamic Streaming advocated by Adobe [9].

For Android-based devices, we found usage of both PD and PD-byterange based approaches in our analysis. The usage is dependent on the type of video subsystem used on each device model. In iOS-based devices, for most short-videos (less than 10 minutes) PD-byterange is used; however, for some videos Apple's developer guidelines require the use of HLS [8]. We did not find any examples of the HTTP Smooth Streaming or HTTP Dynamic Streaming from Adobe in the traffic from the devices we studied but this may be biased based on the subset of devices in the network studied.

In addition non-HTTP based protocols do not account for a significant amount of traffic as we show in Section 3, and therefore we do not investigate them further.

2.1 Data Sets

The data for this study was collected in two national data centers of a large US based wireless provider. More specifically, the traffic is analyzed on the Gn interface between the Gateway GPRS Support Nodes (GGSN) and Serving GPRS Support Nodes (SGSN). Table 1 shows the details of the two data sets. They were collected at the same time and were covering a fraction of the traffic in each data center which mainly contains traffic of smartphones and tablet type devices. In total, the data sets represent usage from approximately 3 million subscribers between the two data sets. While this is a large data set, the results may not be representative of other networks or countries, depending on the combination of wireless devices, the types of content providers and the behavior of users. The collection was limited to 8-9TB of data traffic in each dataset due to 3TB of storage available to each data collector to store flow records. As the traffic volumes observed by each data collector differed between data centers, each data set was collected for different durations.

To perform our analysis we collected flow records, HTTP headers and the first 20KB of each video flow. Video flows were identi-

fied using real-time signatures. They include in particular all HTTP objects with a "video" or "FLV" mime type as well as RTMP (Real Time Messaging Protocol) and RTSP (Real Time Streaming Protocol) traffic. Limiting our collection to the first 20KB of each video represents a compromise motivated by the fact that most video formats contain sufficient information in the first 20KB of a video stream to infer encoding rates and other video parameters necessary for our analysis, and that collecting the entire video would have severely limited our study period or the number of subscribers in our study. The privacy of the subscribers was preserved, since the flow data was not mapped to individual devices and the study focused on the aggregate statistics across all the devices in the data set. All the video content analysis was complete on the data collector using automated tools and no content was exported off the data collector.

2.2 Data Preprocessing

After the data was collected it was preprocessed in multiple ways to support our analysis.

* The HTTP requests and responses were correlated and the actual amount of user layer data was calculated using the flow records and also the TCP sequence number differences for pipelined requests, as in many cases the byte volumes in the HTTP headers were unreliable (see Section 2.3). The device type is also determined from information set in the User-Agents of the request.

* Progressive video downloads (PDs) with multiple byte ranges requests were combined to represent one video object based on the URL used for the byte range requests.

* The multiple video chunks downloaded by HLS for a video are combined into a single video object (i.e. session as a video that is paused and later resumed in split into two seperate objects in our analysis). As there is no generally agreed method on how to combine HLS traffic, for each content provider we needed to develop seperate hueristics based on the request patterns to combine the HLS chunks to sessions. We only combined them for the major content providers which utilize HLS in our trace. In combination our heuristics allowed us to associate over 95% of all HLS traffic volume with their video object.

* To compute such video characteristics as video duration, codec, screen resolution and bitrate, the first 20KBytes of each video object was reassembled and replayed into the popular ffmpeg [6] tool, which provides such meta information to the user in clear text after analyzing the video headers. This allowed for 45% of the video objects to be analyzed by ffmpeg and have their video characteristics extracted. Unfortunately, not all videos contain enough information in the first 20KB or else place the meta data at the end of the video which does not allow ffmpeg to work on our data. In addition, some HLS content providers use DRM to encrypt their video objects which also limited ffmpeg's ability to extract these characteristics. Overall, ffmpeg was able to extract the full set of additional characteristcs from 45% of the video objects.

2.3 Sample Video Behaviours

In the rest of the paper much of the results focus on the complete video objects and HLS sessions. Figure 1 and 2 show two illustrative examples of the more interesting detailed behaviours observed.
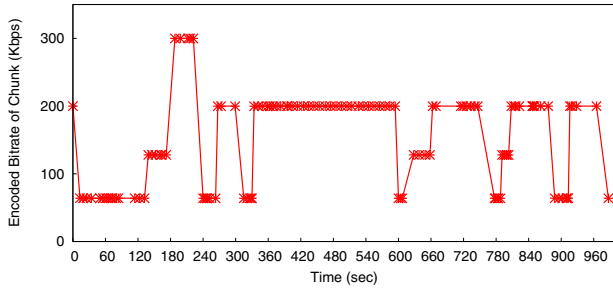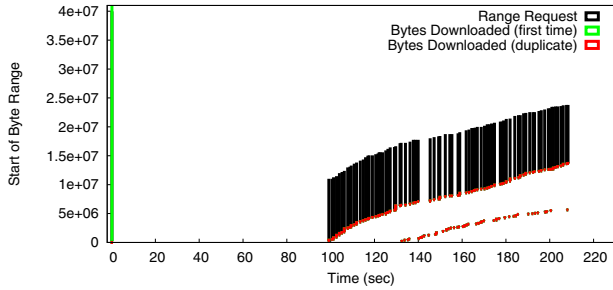
Figure 1: Adaptive Bit Rate: Hulu example

Figure 2: Malfunctioning Byte-Range Example

Figure 3: Application Mix of Overall Traffic using a stacked line chart.

Figure 4: Type of Video Traffic

An example of an HLS video session is shown in Figure 1. This was a Hulu video session where the adaptive protocol can be seen changing between 64, 128, 200, and 300 Kbps encoding rates quite frequently during the session. The interarrival times between chunks is approximately 10 seconds in this example. On average, we measure HLS flows changing bitrates 0.2 times per minute in the video session for content provider which we could determine the bitrate of the chunks. This highlights that available bandwidth in cellular networks can change several times during the course of a TV show or movie session and also that HLS protocols are actively adaptiving their bitrates to these flucutations.

The second example is of an observed PD-byterange download (Figure 2). The black bars show the range of the actual data downloaded and the green bars the byte-range requests size. The red bars show data downloaded of the video that was already previously downloaded. As seen in the graph, the byte-ranges are much larger than the actual content downloaded before the device closes the HTTP connection and starts another. We also observe that for longer videos a large initial download is typically followed by a sequence of smaller downloads for the same byterange. Further lab testing showed that the behavior was reproducible. For instance, after some larger videos were fully downloaded, and until the user had finished watching the video, additional duplicate downloads would occur. This case was observed frequently and generated a significant amount of unnecessary duplicate video content in the trace.

3. TRAFFIC CHARACTERIZATION

In this section, we show highlights of the video traffic we studied. While our analysis was completed on both the NE and WEST data sets, many of the results are quantitatively similar, therefore, we
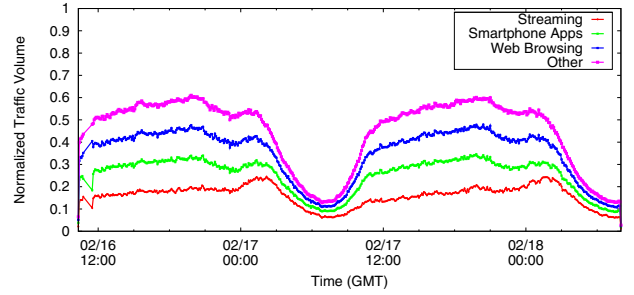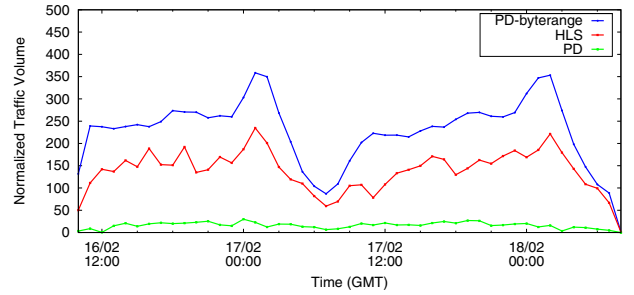
have chosen for more clarity to provide the WEST data set results only unless noted otherwise.

Overall, video streaming traffic accounts for 36% of the traffic in the data studied. This is shown in Figure 3 in a stacked line chart. During the busy hour the video streaming share is actually 30% but peaks at 50% during the off peak hours.

The streaming video traffic is primarily delivered using HTTP-based methods. The use of HTTP-based method is true across multiple smartphone OS's as well. Traditional multimedia protocols like RTSP and RTMP account for only 1.3 % and 0.4 % of the traffic respectively. A more in depth analysis actually shows that the use of RTSP and RTMP on smartphones and tablets is much lower than reported as nearly all such traffic originates from either a laptop with a cellular card or a laptop tethered to a smartphone.

The number of video objects and the data volume in each category of the HTTP-based methods is shown in Table 2. One surprising category of video content was being generated by one advertising company (labeled separately as Advertisement Objects in the table). Interestingly, the number of objects of this category is rather large accounting for 44.5% of the objects in the NE data set and 44.8% in the WEST data set, even if by volume this type of traffic accounts for less than 0.1% of all video traffic. On closer examination we discovered that even though this content is marked as video content in the mime header, it actually contains just text-based user tracking information and no video content at all. As this study is focused on understanding video content in cellular networks, we excluded this data from the analysis presented in the remainder of this paper.

Table 3 shows the video characteristics per content provider and have been obfuscated to hide marketshare. The % objects fields after the "Top" columns shows the share the top category account for, and the UGC label refers to User Generated Content. In this table,

Table 2: Video Object Types

Data Set	HLS Objects	PD Objects	PD byterange Objects	Adv. Objects	HLS TBytes	PD TBytes	PD byterange TBytes	Adv. TBytes
NE	0.2 (5.4%)	0.8M (15.2%)	1.9M (34.9%)	2.1M (44.5%)	2.6 (29.5%)	0.4 (4.6%)	5.9 (65.7%)	< 0.1 (.1%)
WEST	0.2 (4.8%)	0.7 (15.3%)	1.7M (35.1%)	2.1M (44.8%)	2.9 (36.3%)	0.3 (4.2%)	4.8 (59.4%)	< 0.1 (.1%)

Table 3: Content Provider Breakdown

Content Providers	Objects	Bytes	HLS	PD-BR	PD	Top Type	Objects	Top Screen Size	Objects	Avg Bitrate	Avg Dur.	Median Dur.
Video Stream 1	2.8%	33.2%	100%	0%	0%	HLS	100%	Unknown	100%	613.59	10.8	9.5
UGC 1	37.4%	19.2%	0%	99.53%	0.47%	video/3gpp	92%	176x144	3%	161.59	4.09	3.5
Adult 1	2.8%	6.1%	0%	99.66%	0.34%	video/mp4	100%	240x176	6%	955.8	4.82	7.0
Adult 2	1.5%	4.1%	0%	99.45%	0.55%	video/mp4	98%	240x176	2%	1057.63	4.84	10.0
UGC 2	4.2%	4.0%	0%	94.25%	5.75%	video/3gpp	58%	640x360	4%	334.44	3.95	3.4
Adult 3	2.1%	3.5%	0%	99.45%	0.55%	video/mp4	100%	320x240	2%	498.71	5.98	3.5
Music 1	6.0%	2.3%	0%	15.43%	84.57%	video/mp4	100%	480x270	63%	1146.51	0.26	0.3
Video Stream 2	0.3%	2.0%	99.21%	0%	0.79%	HLS	99%	Unknown	99%	435.05	9.16	4.8
Music 2	0.8%	1.4%	98.37%	1.45%	0.18%	HLS	98%	Unknown	98%	1339.04	0.96	0.6
Adult 4	0.6%	1.3%	0%	99.57%	0.43%	video/3gpp	100%	640x480	0%	389	0.72	2.0
Social Network 1	2.8%	1.3%	0%	99.68%	0.32%	video/mp4	100%	400x300	2%	364.4	0.85	0.6
Social Network 2	1.7%	1.0%	99.98%	0.01%	0.01%	HLS	100%	Unknown	100%	319.72	1.06	1.5
Adult 5	0.9%	0.7%	0%	99.68%	0.32%	video/mp4	100%	320x240	5%	474.97	2.33	2.0
Adult 6	0.2%	0.6%	0%	99.78%	0.22%	video/mp4	100%	320x240	3%	549.54	3.42	3.6

it can be seen that video content delivered is dominated by a few top content providers: the top 10 account for 77% of the video streaming. We also observe that each content provider is using mainly a single method to deliver the videos. While some of the previous distributions had some distortions due to some smaller content providers, comparing only the top 2 content providers, which are also the top HLS and top PD-byterange providers, a clearer picture emerges. HLS has longer duration video sessions and a higher average bitrate.

Table 4 is a similar table to the previous but shows the video characteristics by device type. The most common container and screen size resolutions of the videos with the % objects columns after the "Top" columns showing the share the top value accounted for in the data.

Figure 5 shows the average bitrate encoding for the HLS and PD byte-range videos. The PD videos can be seen to be encoded at 3 main rates of 87 Kbps, 255 Kbps, and 1150 Kbps. For the HLS streams, the average bit rates are relatively stable with an average delivered bit rate around 500-600 Kbps. The similarity is not surprising as HLS is an adaptive bit rate protocol that adapts to the available bandwidth and resources on the device. Many of the content providers using HLS have a variety of bitrates that are available. For example, *Video Stream 2* in Table 3 has 64, 128, 200, 300, 400, 650, 1000, 1500 Kbps, as encoded rates of the chunks we saw delivered. Several other content providers using the same CDN encode their video chunks at: 110, 200, 450 and 800 Kbps. Unfortunately for *Video Stream 1*, this content provider encrypts its traffic, so we do not know the exact encoding rate levels used the individual chunks for this provider.

Figure 6 shows the object size distribution of the individual "chunks" of the PD-byterange and HLS videos, as well as the stitched total videos object sizes. As pointed out in Section 2.3 the chunk sizes of byte range requests for one device are rather odd. In fact, 70% of all requests in our data set are for small chunks which follow the large initial chunk.

The average HLS chunk is 362 KB with most ranging between 100-300 KB in size. The interarrival times between chunk requests are clustered around 3 and 10 second intervals, with a small number of sessions at 20 and 30 second intervals as well. We were able to verify the 10 second chunk encoding using the video durations extracted on many of the video chunks decoded by ffmpeg.

Surprisingly, when comparing the HLS object sizes and durations to the PD-byterange results in Figure 6 and Figure 7, the HLS durations are shorter than the PD-byte-ranges durations. This is the result of HLS being influenced by a small number of content providers. The video duration of one HLS content provider is always between 1 to 1.5 minutes long. While other content providers have much longer session durations such as *Video Stream 1* and *Video Stream 2*. It is also interesting to note that, for these streaming providers, few sessions last the entire movie or TV show length. For many cases, sessions are interrupted and are then later seen to be resumed. On average, for these content providers each video is resumed 0.19 times.

Another interesting observation is that for HLS video the video containers of the videos are already in an optimized state for mobile video. For PD and PD-byterange video objects, we were expecting to see more variety initially with different formats supported and used on different types of devices. However, we found that not to be the case, and that most videos were already in optimized containers for 3GPP video. They were encoded using H.264 and are in either "video/mp4" or "video/3gpp" containers for 80.7% of the objects and 95.2% of bytes. Only 1.1% of objects and 0.8% of bytes were in FLV format in the WEST data set and 2.1% and 1.2% of bytes were in FLV format in the NE data set. This is also consistent across different devices type that FLV is not a popular format as observed in Table 4.

Finally, we studied how many videos were fully downloaded. Fortunately for PD videos we were able to measure the total amount of data actually downloaded before abandonment (Figure 8) and compare it to the video size reported. Some videos are observed to download more than 100% of video. This is a combination of the issue observed in Section 2.3 and also from fastforwardng and rewinding when a video is not fully cached. What can be observed is that only 40% of videos are completely downloaded, and for 50% of videos only 60% of the video was downloaded. While this does not show how much was watched, due to buffering, it gives a glimpse into the amount of video that is abandoned before being fully watched.

4. VIDEO POPULARITY AND CACHING

Having observed that streaming video accounts for a dominant part of the cellular broadband traffic and that the lion's share of this traffic is carried over HTTP, a natural question is what are effec-

Table 4: Devices Breakdown

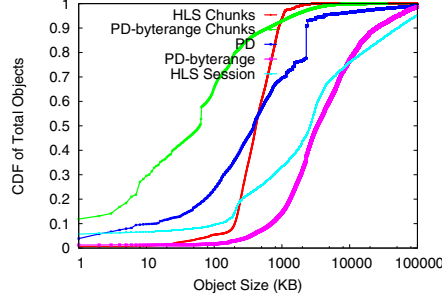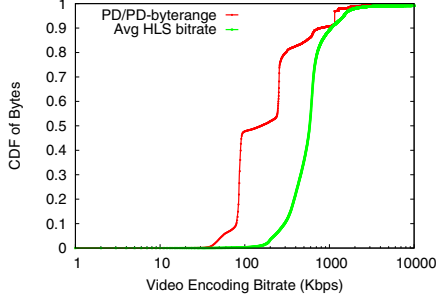| Device	Top Content Type	% Objects	Top Screen Size	% Objects	Average Bitrate	Median Duration
Smartphone 1	video/mp4	58%	480x270	3%	278.9	2.85
Tablet 1	video/mp4	55%	320x240	2%	315.5	2.78
Laptop 1	video/x-ms-wmv	46%	640x360	7%	689.5	0.50
Smartphone 2	video/mp4	76%	320x240	2%	169.8	3.55
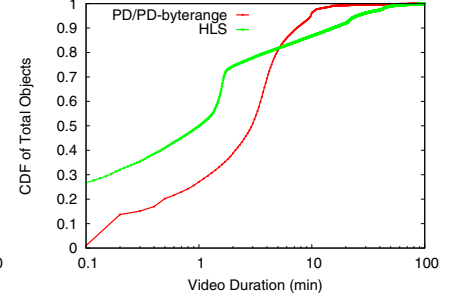
Figure 5: Distribution of Encoded Video Bitrates for PD,PD-byterange

Figure 6: Object Size Distribution of Video Objects

Figure 7: Distribution of Video Durations
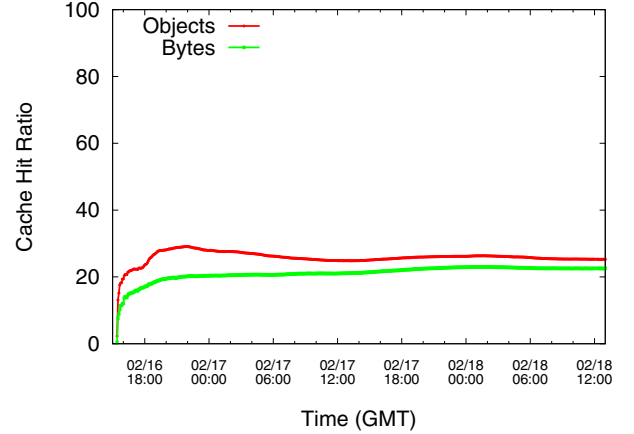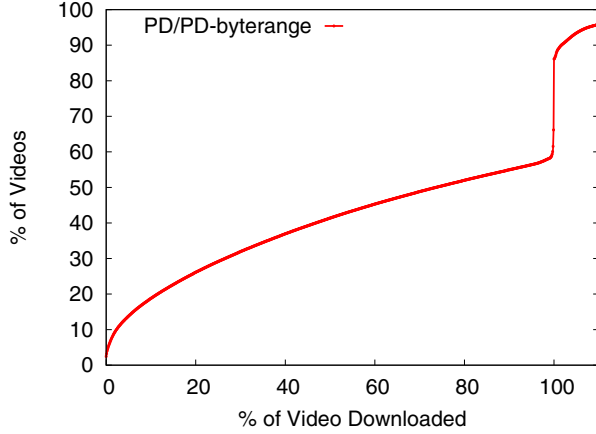
Figure 8: Percentage of each Video Object Downloaded

Figure 9: Cache Hit Ratio of Video and Cache Size

tive ways to deliver this content. Even though the impact of video caching on the RAN is limited, as highlighted in the Introduction, the growth of video traffic in cellular network still warrants a close study of the topic to understand the possible bandwidth savings in the backkhaul networks and the potential to improve end-user experience by reducing delay. To answer how much potential exists one must in particular consider multiple aspects of the traffic characteristics: whether content providers allow their video objects to be cached, the spatio-temporal popularity characteristics of the requested video objects, and whether the content is encrypted, etc.

We first examine the requested objects based purely on the cache control directives in the HTTP request response messages. These enable specification of *Cacheability* : the server can specify whether a requested object can be cached, and then can indicate if a particular request can be served from cache or needs to be fetched from the server. *Freshness* : the duration for which a downloaded copy of a cacheable resource is valid for serving a request.

Combining the relevant information bits from the request-response pairs for a video object, we can categorize all the requests in terms of *Cacheability* as:

uncacheable: the request can not be cached;

cacheable-local: the requested object is cacheable and the request can be served from the cache without contacting the server as long as it is fresh;

cacheable-validate: the requested object is cacheable, but the cache needs to check with the server that its local copy is valid every time the object is served to a client.

We find that for PD-all (both PD and PD-byterange) traffic, 8.0% of all the requested data are uncacheable, 63.2% are cacheable-local and 28.8% are cacheable-validate. For HLS traffic the corresponding percentages are 78%, 11% and 11%, respectively. An upper bound on the proportion of requests (by bytes) that can be potentially served from the cache (assuming that the content is fresh for cacheable-local cases and that the copy is still valid for the cacheable-validate cases) is a very large 92.0% of all the requested traffic for PD-all and only 22% for HLS.

We next consider the distribution of the freshness for object requested with a server-specified freshness duration > 0. Across all
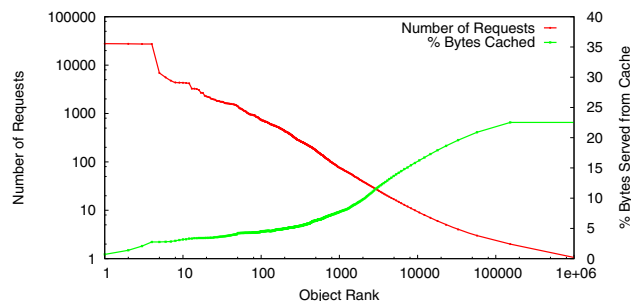
131

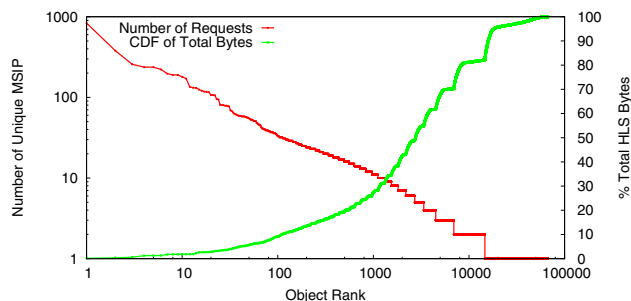**Figure 10: Object Popularity of PD-all Videos**

Figure 11: Object Popularity of HLS Videos

the video types, we find that overall freshness duration is $< 1 hour$ for 9.4% of the requested traffic, and exceeds 1 day for 32.3% of the requested traffic. There are 2 concentrations: 14.5% of the traffic has freshness duration of 1 hour and 43.8% of the traffic have freshness duration of 1 day. The results show that substantial fraction of the traffic seems to have very short freshness durations set. Given that video objects normally do not change at short timescales, and that increasing the freshness duration would increase the likelihood of serving more requests from cache, content providers should examine whether longer freshness durations (eg. a day) would meet their application needs.

4.1   Caching Simulation

In addition to the HTTP caching directives, the cache size and the spatio-temporal pattern of request arrivals also determine the extent to which requests can be served from the cache. To understand the maximum obtainable benefits and to focus on the impact of the spatio-temporal pattern of request arrivals, we consider an unlimited cache size and simulate HTTP forward caching for the entire traffic data for PD-all. Conceptually the result achieves the spatio-temporal caching benefits of a cache located at the National Data Center (NDC), which hosts the GGSNs through which all the measured traffic flows. Our simulator is implemented to follow the HTTP caching standard in [1].

Figure 9 depicts, for a given time instant t, the fraction of all traffic that was served from the cache from the beginning of the simulation until time t. At the end of the 48 hour trace, we see that around 23.5% of all the requested traffic (in terms of bytes) was served from the cache. Recall that 92.0% of the PD-all traffic was cacheable based on caching directives, therefore 25.5% of the cacheable PD-all traffic was actually served from cache. The remaining 74.5% of the cacheable traffic had to be sourced from the server - either because they accounted for the first download of an object, or because the cached object was stale when requested.

Figure 10 shows the ranked list (red curve) of all the PD-all videos requested, in decreasing order of popularity and their cumulative contributions to the proportion of total traffic that was served using cached content (green curve). There is significant skew in popularity. The top-100 and top-1000 most popular objects account for 11.3% and 19.2% of all the requests respectively. The caching simulation reveals that the traffic served from cache that was associated with the top 100 and top 1000 objects is 4.5% and 7.7% of the total requested traffic respectively. Recalling that overall, 23.5% of all the requested traffic (in terms of bytes) was served from the cache, a policy that caches just the top-1000 objects of the total 1.2 million unique PD-all videos would already realize a third of the overall caching benefits. Detailed examination showed that

the top 5 objects were the same video being served from different servers of a content provider and that most of the top 200 objects were advertising videos. Another interesting point is that the popularity distribution has a long tail which potentially limits the overall caching that can be achieved: 50% of all requests for cacheable objects, were for objects that were requested once.

Recall that unlike PD-all videos, a relatively small proportion of HLS traffic is cacheable because the HTTP cache control directives set by the content provider do now allow it to be cached. In addition, we also found that much of the HLS objects were encrypted - this adds complexity to using HTTP cache-based delivery for such objects, as clients would also need appropriate keys to decrypt the objects. While the combination of wide-spread non-caching directives and encryption are negatives from a caching viewpoint , it is still interesting to examine the popularity characteristics of content being transmitted using HLS. Specifically, assuming HTTP directives were more permissive and it were possible to address the encryption-induced challenges, would it make sense to use caching for this content ? As a first step to answering this question, in Figure 11 we plot the ranked list (red curve) of all the HLS videos requested, in decreasing order of popularity and their cumulative contributions to the proportion of total traffic (green curve). Of the total 66000 unique HLS objects, the top 1000 and the top 2800 account for 27.4% and 50% of all the HLS traffic respectively. This popularity skew suggests that caching even a small fraction of the popular HLS videos can lead to substantial caching gains.

5.   CONCLUSION

This paper is the first large scale fine grain analysis of video traffic generated by 3 million devices on a cellular network. Some of the findings include the fact that one third of the cellular traffic comes from Over The Top video traffic, that one third of that traffic is HLS and that, in practice, adaptive bitrate encoding seems to be effective as frequent bitrate adaptations are observed to sustain the video stream (0.2 bitrate changes per minute). Moreover, we also observed that only 40% of the videos are completely downloaded and that 80% of the videos are encoded at or below 255kbps. Finally, we studied the cacheability of that video content. These results will help guide future video modeling and optimization work.

6.   REFERENCES

[1] Hypertext Transfer Protocol – HTTP/1.1 . http://www. w3.org/Protocols/rfc2616/rfc2616.html, 1999.
[2] iOS Technical Note TN2224. http://developer.apple.com/library/ios/ #technotes/tn2224/_index.html, April 2010.

[3] Adaptive Bitrate Protocols. `http://en.wikipedia.org/wiki/Adaptive\_bit\_rate`, 2011.

[4] AT&T SXSW Press Release. `www.att.com/Common/docs/SXSW_Network%20Fact_Sheet.doc`, March 2011.

[5] Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2010Ú2015. `http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-520862.html`, February 2011.

[6] FFmpeg. `http://www.ffmpeg.org/`, 2011.

[7] HTTP Live Streaming. `http://tools.ietf.org/html/draft-pantos-http-live-streaming-06`, 2011.

[8] iOS Developer Library. `http://developer.apple.com/library/ios/#documenation/networkinginternet/conceptual/streamingmediaguide/UsingHTTPLiveStreaming/UsingHTTPLiveStreaming.html`, 2011.

[9] S. Akhshabi, A. Begen, and C. Dovrolis. An Experimental Evaluation of Rate-Adaptation Algorithms in Adaptive Streaming over HTTP. In *MMSys'11*, San Jose, USA, 2011.

[10] J. Chesterfield, R. Charkravorty, J. Crowcroft, P. Radriguez, and S. Banerjee. Experiences with multimedia streaming over 2.5G and 3G Network. In *BroadNets'04*, Chicago, USA, 2004.

[11] J. Erman, A. Gerber, M. T. Hajiaghayi, D. Pei, S. Sen, and O. Spatscheck. To Cache or not to Cache: The 3G case. In *IEEE Internet Computing*, 2011.

[12] J. Erman, A. Gerber, M. T. Hajiaghayi, D. Pei, and O. Spatscheck. Network-Aware Forward Caching. In *WWW'09*, Madrid, Spain, 2009.

[13] H. Falaki, D. Lymberopoulos, R. Mahajan, S. Kandula, and D. Estrin. A first look at traffic on smartphones. In *Proc. IMC*, 2010.

[14] H. Falaki, R. Mahajan, S. Kandula, D. Lymberopoulos, R. Govindan, and D. Estrin. Diversity in Smartphone Usage. In *MobiSys'10*, San Francisco, USA, 2010.

[15] A. Gember, A. Anand, and A. Akella. A Comparative Study of Handheld and Non-Handheld Traffic in Campus WiFi Networks. In *Proc. Passive and Active Measurement*, 2011.

[16] J. Huang, Q. Xu, B. Tiwana, Z. M. Mao, M. Zhang, and P. Bahl. Anatomizing Application Performance Differences on Smartphones. In *MobiSys '10*, pages 165–178, New York, NY, USA, 2010. ACM.

[17] G. Maier, A. Feldmann, V. Paxson, and M. Allman. On Dominant Characteristics of Residential Broadband Internet Traffic. In *IMC'09*, Chicago, USA, 2009.

[18] G. Maier, F. Schneider, and A. Feldmann. A First Look at Mobile Hand-held Device Traffic. In *PAM'10*, pages 161–170, Berlin, Heidelberg, 2010. Springer-Verlag.

[19] Y. J. Won, B.-C. Park, S.-C. Hong, K. B. Jung, H.-T. Ju, and J. W. Hong. Measurement Analysis of Mobile Data Networks. In *PAM'07*, pages 223–227, Berlin, Heidelberg, 2007. Springer-Verlag.

Summary Review Documentation for

"Over the Top Video: The Gorilla in Cellular Networks"

Authors: J. Erman, A. Gerber, K. Ramakrishnan, S. Sen, O. Spatscheck

Reviewer #1

Strengths: The authors collect data from the link connecting the SGSN and GGSN nodes in a large cellular network, and on two different locations. They analyze it to identify the videos, the protocol they are relayed on, the resolution, as well as cacheability. Very nice study.

Weaknesses: The paper seems to be rushed at times. One figure is repeated the same in two places (5 and 8), the authors never introduce the content providers and make mention to them without and explanation. But these could be fixed in the final version.

Comments to Authors: This is a very nice short paper. The authors use a new source of data, carefully collected for 24 and 48 hours in two different locations in a large cellular network. They collect flow information and packet information (the first 20KB), thus being able to also look into the streaming bit rate of the video, the type of video, its resolution, information that is not easily accessible. Moreover, they are able to see the protocol that carries the video in question and study its properties.

Interesting findings are:
- majority of video is carried over HLS
- the transmission of the video is done over varying bit rates that can be observed through the trace, indicating the actual need for streaming rate adaptation to wireless conditions
- most content today is streamed at rather low streaming rates
- 40% of the videos never complete
- given that the majority of video traffic is transported over HTTP, the question of cacheability is asked and the authors simulate the actual gains one could expect from such a solution by replaying the collected traces.

One question I had is whether the authors know if there are any content acceleration middleboxes in the path observed. That would be essential to understand since then your infrastructure monitors the behavior of the middlebox and not the origin video server.

More detailed comments:
- Section 2.1: I assume that the capacity for data collection is 9 TB and not 3 (otherwise text disagrees with Table1)
- Figure 3 says that it displays normalized volume, but in that case I would expect the x-axis to be between 0 and 1 or 0 and 100%. In your final version, you need to describe the metrics displayed in the figures more carefully.
- The paragraph below Figure 5 points to Figure 4, but that figure does not contain the information discussed.
- Video providers are discussed in page 4 without any prior reference and only appear in the next page. You need to reorganize the paper for better exposition.

- I was actually surprised that the top resolution covers only 3% of the videos. How many resolutions do you see being used in total?

All in all, I think this is a very decent attempt for this first study of its kind. The writing is rushed but I think the content is there and it's interesting. I would be in favor of acceptance.

Reviewer #2

Strengths: The paper presents characteristics of a large volume of video traffic on a cellular network. There are some interesting data points, e.g., the distribution of different types of video streaming methods.

Weaknesses: The data seems not well analyzed, even for a short paper and some of the results are too anecdotal. The paper is poorly written in parts.

Comments to Authors: Video streaming over cellular networks is definitely a growing part of traffic, and this analysis from a large cellular network is certainly of great interest. However, I found the paper to have inadequately analyzed, what certainly is a very rich dataset.

As an example, Figure 2 shows an anecdotal download using the PD-byterange method. The results are strange and unexplained in the paper. Does this kind of download happen a lot, or is it just a rare occurrence?

Fig 5 and Fig 8 appear to be identical.

Some of the results presented in the paper, i.e., choice of video adaptation rates used by different applications, are not really actionable. The only part of the paper that I personally found particularly useful was the cacheability discussion.

Overall, this seems like a large dataset that requires more interesting analysis and the authors probably have barely scratched the surface with this work.

Reviewer #3

Strengths: Important topic.

Weaknesses: This reader is somewhat concerned with the representativeness of the results from this study, due to various indications shown in the paper. For example, Section 2.3 stated that video objects were replayed into the popular ffmpeg tool to extract out the needed information, and "In our data set ffmpeg was able to parse 45% of the video objects".

Another concern is the rather different ratio of uncacheable data between PD and HLS (8% vs 78%), which was stated but not clearly explained. The paper explained this is largely due to encryption of HLP. But how should one view this result together with the Section 2.3 statement in the above ("was able to parse 45% of the video objects")?

Section 4 stated that " the impact of video caching on the RAN is limited". Is this referring to the above result, or something else?

Comments to Authors: None.

Reviewer #4
Strengths: This is a classic network measurement paper. The measurements are strong, interesting, and provide some new insights.

Weaknesses: The paper tends, at times, to feel like a data dump.

At times some of the underlying issues of data quality seem a little brushed under the carpet. For instance, how do the authors account for a possible bias caused by some brands of devices more represented than others in their dataset, in comparison to other wireless networks?

Comments to Authors: The paper seems to be along the same lines as "P2P, the Gorilla in the Cable (2003)" by some of the same authors. Although the topic is actually different, the similarity of titles leads one to want to know the relationship between the two papers.

There are a few statements that are just a little off. For instance, the two datasets were collected at the same time, but one is longer than the other. I guess it is just the same start time? The data collector had 3 TB storage, but the traces where both over 8 TB? Please clarify.

There is a general statement about the number of subscribers covered in the study, but it isn't made clear if this is a total of which we see a sample (in each area), or if this is the number sampled.

Why does Figure 9 go above 100%?

There are quite a few small bugs and typos in the paper.

Reviewer #5
Strengths: - Novelty: new aspect of mobile cellular network usage captured for the first time, with some important new insights (on the amount of incomplete videos, and the relative spread of their popularity).
- Level of detail: practitioners and people working on detailed protocols of video encoding could find interesting bits.
that there are a couple of anecdotal evidences it is nice to monitor how YouTube works in a systematic fashion.

Weaknesses: - Not much, some limitations could be mentioned a bit more clearly, and naturally it would be awesome to understand spatial property as well, but there is so much one can do in 6 pages.

- The authors could have contrasted their findings with wire networks to emphasize the new characteristics found in this context.

Comments to Authors: This paper is I think a no-brainer accept. It is not revolutionary but it provides detailed insights into video traffic on cellular network, by using collected measurements and presenting a very clear set of numbers. Among them one that stands out particularly is that a large part of the videos are not completed (although it is only seen on a part of the data set), and that caching is not going to reduce the traffic much, although it will be effective for the most popular videos.

This is well done and contains quite clear description of video encoding characteristics. Seems a great match for a short paper.

A few points to improve:

1. I think that it is important to put some of the results in perspective of your limitation. In particular, the fact that only PD videos are examined from the completion standpoint is (1) only establishing this result on a part of the data set, (2) may be biased (some of these videos may take more time to load as they retrieve from multiple sources and needs reassembling, and hence may create too much impatience in the users). This point ought to be clarified.

2. When you mentioned caching, it could also be the case that caching improves delay, which seems to be an important thing given the relative fraction of uncompleted download indicating that users are affected by delay. It would be nice to see that point discussed.

3. I am surprised not to see any discussion of previous findings of videos on other networks. I imagine that the rate could be quite different (although it depends on which year we consider). But would the popularity be the same? Would the amount of abandons and video lengths be different?

4. The restriction to remove ad-video seems appropriate, but it seems also a bit incomplete (you mention you remove another domain, but isnt there more like these ones)? Could you provide some numbers (based on duration) to justify that no more are present?

This seems in particular important as you indicate later that the most popular videos are ads.

Response from the Authors

We thank the reviewers for their constructive comments.

We have addressed all the minor issues in the paper and made the appropriate adjustments (e.g. we have removed the duplicate figure pointed out in the review).

While we have done a careful example of the sample video behaviors in Section 2, this was more to illustrate the behavior than to come to our conclusions based on them. For example, we have observed that HLS sessions, in general, show their ability to adapt to changing conditions. Figure 1 is an example to illustrate this more carefully. The issue of duplicate downloads, as

highlighted in Figure 2 is not an exception, but a common behavior across multiple applications that we have observed on multiple platforms (device OSs). We have explained this anomalous behavior better in Section 2, and also described how the behavior can be reproduced. To further establish that this is not just based on the observation of a few example situations, Figure 8 shows that for a significant percentage of videos more than 100% of the content is downloaded by the end-point. We believe that this adequately addresses the issue brought up by the fourth reviewer.

In addition to providing a novel characterization of video traffic on 3G networks, we also make useful observations about how a carrier can appropriately handle this class of traffic. This study is also important for application developers to understand their impact on the network and to more optimally use the cellular network.

This study and observations are based on traffic generated by millions of users in a large tier-1 network. The large footprint of the data set gives us a lot of confidence about the generality of our main findings. While there may be some idiosyncrasies e.g., the device mix studied, we have looked at other platforms with different devices and OS mixes and many of these conclusions still hold. Finally, we have also reminded readers that the results naturally may not be completely representative of the video traffic in every cellular network, since the combination of wireless devices, the types of content providers and the behavior of users might be different.

We leave the comparison of our results on cellular networks with video characteristics on wireline network for future work. Thanks for the suggestion