

Internet Media Upload Caching for Poorly-Connected Regions

Paul Schmitt
UC Santa Barbara
pschmitt@cs.ucsb.edu

Ramya Raghavendra
IBM Research
rraghav@us.ibm.com

Elizabeth Belding
UC Santa Barbara
ebelding@cs.ucsb.edu

ABSTRACT

Media uploads and downloads, even those on the order of a few hundred kilobytes, commonly fail when attempted over lossy, low-bandwidth, and high latency connections. These conditions, which are common for networks in rural, resource-poor areas, result in the inability for residents of these areas to fully participate in the modern Internet. We study traffic traces collected from two networks and find high locality of interest as well as poor performance for Internet media. We also find that users often abort uploads due to network performance. Given that media content produced by local users is often heavily consumed by local users, we propose VillageCache, a system which allows for appropriate local transformation and redistribution of media uploaded through an Internet cache. We build VillageCache and find it successfully delivers cached media, providing an orders of magnitude improvement in file transfer performance, bandwidth reduction, and the virtual elimination of video stalls in the face of poor network connectivity. Caches with upload capabilities placed at the edge of poorly-connected networks will enable users to produce and consume media-rich content while mitigating the constraints present in under-resourced networks.

Categories and Subject Descriptors

H.3.5 [On-line Information Services]: Web-based services

Keywords

World Wide Web; Internet Multimedia; Caching; Upload; Low Bandwidth

1. INTRODUCTION

The great majority of modern Internet content is built with the assumption of high-quality, always-on network connectivity. However, network infrastructure in developing regions typically lacks in the bandwidth available to users [1],

and often exhibits poor latency and loss characteristics. While great strides have been made to improve connectivity through ICTD research, the gap between what media-rich sites demand and what rural network infrastructures are able to provide remains, and in many cases is growing.

Without full participation on the Internet, cultures unique to developing regions are at risk of being forgotten [2]. Web services that encourage participatory culture, often characterized as “Web 2.0,” are critical assets in the democratization of communication and provide a voice to people that previously went unheard. Traditionally, research in the ICTD space has focused on improving connectivity or the user experience for content *consumption*. This is often accomplished using content prefetching and predictive caching techniques; or requiring users to navigate through software platforms that alter the experience from the “native” versions of Internet sites. While these solutions help narrow the digital divide, the next logical step is to enable users to provide insight into their lives as content *producers* through mainstream sites, not heavily modified sites designed to avoid bandwidth consumption.

In recent years, web content providers have actively sought to improve the user experience for those connecting in developing regions. Google, for instance, has extensively grown their content delivery infrastructure by placing caches on local and regional ISP networks across the world [3]. However, despite the physical proximity to cached content, researchers have found that lack of peering and Internet exchange point (IXP) presence for some Internet service providers causes traffic between African clients and nearby content servers to take circuitous routes, with many routes detouring through European routers before returning to Africa [4]. Facebook has also recently pushed to make their service more available to users in developing regions by engineering their mobile app to use less bandwidth as they found that the poor network conditions in developing regions ‘resulted in slow load times and constant crashes’ [5]. The newly released ‘Facebook Lite’ [6] app only downloads high resolution copies of media if the user explicitly chooses to view a photo or video. The changes indeed make the Facebook app more efficient at consuming media in the face of poor network connectivity; however, uploading content remains an unsolved challenge. We strive to minimize bandwidth consumption for locally uploaded media by transforming and redistributing copies of local content at the network edge.

We perform in-depth, longitudinal analysis on network traces from two production networks in order to better understand network performance and user behavior in the con-

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.

ACM DEV 2015, December 1–2, 2015, London, United Kingdom.

© 2015 ACM. ISBN 978-1-4503-3490-7/15/11 ...\$15.00.

DOI: <http://dx.doi.org/10.1145/2830629.2830636>.

text of media production and consumption. One network is in a village in Sub-Saharan Africa and the other is in a tribal community in the United States. We find that web media content represents a large percentage of overall traffic in terms of bytes and performance of associated traffic is poor (i.e. large uploads from the local network often fail or are canceled entirely by users due to poor network performance). We further find high locality of interest in media as content uploaded by local community members is heavily consumed by users in the same network. Our findings lead to our proposed solution for allowing uploaded content to be captured, transformed, and redelivered from an Internet cache placed at the access network edge.

Our network trace analysis findings inform our design of VillageCache, a system designed for placement at the network edge that captures locally uploaded media, makes the necessary transformations, and delivers it to local users in order to avoid unnecessary bandwidth consumption on the Internet gateway link. We build a prototype with the assumption that users in such regions have a relatively fast local network connection (e.g. a WiFi LAN) such that they are able to locally transfer large files at reasonable speeds. Unlike a traditional web cache, our proposal avoids the requirement for a download of the object in order to be cached by transforming locally-uploaded media and directly injecting them into the cache for subsequent requests. This is critical as even a single download of high resolution media content is often difficult or impossible in constrained networks.

Our paper makes the following contributions:

- We perform an extensive packet trace analysis from two production networks, leveraging multiple techniques to shed light on usage and performance across multiple layers of the network protocol stack.
- Through our analysis, we find poor media performance for users in the observed networks. Additionally, we observe that moderate and large uploads are rarely successful over a constrained link. We find high locality of interest in locally produced media. These findings provide key observations for our solution.
- We propose new HTTP headers to enable Internet caches to transform and store locally uploaded media for subsequent requests without requiring a single download from a web content server.
- We evaluate VillageCache in a lab environment that emulates realistic network properties. We show empirically that the system eases bandwidth usage and vastly improves the user quality of experience. Our solution allows users in poorly-connected networks to interact with modern Internet media content, which was previously impractical due to network conditions.

2. TRAFFIC ANALYSIS

We examine network packet captures gathered from two production networks: one in the village of Macha, Zambia, the other in a tribal community within the United States in order to gain insight into Internet performance and user media consumption.

Macha, Zambia: The Macha network traces we analyze represent a full calendar year, from April 2011 through March 2012. As is somewhat common for networks in developing regions, we have a data hole, in this case during August through October as the network was not opera-

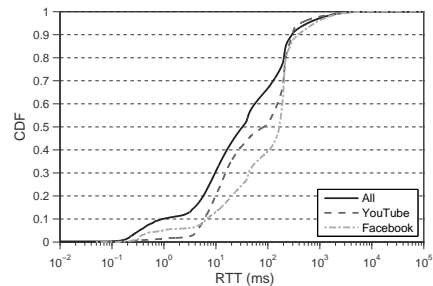


Figure 4: Observed RTT in Macha.

tional during that time. Internet access in Macha is available through a microwave terrestrial link shared over a local wireless network connecting approximately 300 community members. The Macha network is delivered using 802.11 devices acting as point-to-point links, hotspots, and layer-2 bridges. Public Internet access is gained through an Internet café. The bottleneck of the network is the shared Internet gateway link which has a committed speed of 2Mbps.

Tribal Digital Village (TDV): We also analyze six months of traffic collected in 2014 from the tribal digital village (TDV) network deployed in southern California¹. The network currently consists of roughly 350 homes subscribing to either 2 Mbps or 3 Mbps service. The Internet backhaul link is 500 Mbps with 200 Mbps provisioned at the gateway location and the remaining 300 Mbps provisioned for subsequent hops. Microwave links provide the backbone to reach locations up to roughly 80 km away. Last-mile access is provided using 2.4 GHz and 5 GHz WiFi equipment.

2.1 Traffic characterization

We capture traffic using tcpdump at the networks' gateway links and analyze the packets using Tstat [7] and Bro [8] to profile real-world usage. We first investigate TCP performance metrics for all traffic as well as traffic specifically related to YouTube and Facebook to profile usage of those services as they are the most popular services in terms of flows and bytes in the Macha network. Next, we explore the properties of Macha flows that are classified as 'user-interrupted.' Lastly, we characterize the user quality of experience (QoE) for observed YouTube video traffic. The focus of our analysis on the TDV network traffic is on Instagram media; Instagram is the most commonly accessed media type in this trace. Overall traffic performance analysis for this network is omitted for brevity. Our analysis motivates the design of VillageCache by revealing high locality of interest for locally-created media as well as poor network performance for consuming such media using the existing infrastructure.

2.1.1 TCP latency

We investigate latency in the Macha network, which dramatically affects TCP performance and packet loss. Our analysis considers average TCP round trip time (RTT), as calculated by Tstat, for each flow. As shown in Figure 4, both YouTube and Facebook traffic typically experience longer RTT values than other traffic. The mean RTT for all traffic is 159.0ms, while the mean RTTs for YouTube and Facebook flows are 173.7ms and 232.1ms, increases of 9.2% and 46.0%, respectively. This is likely due to the physical

¹<http://sctdv.net/>

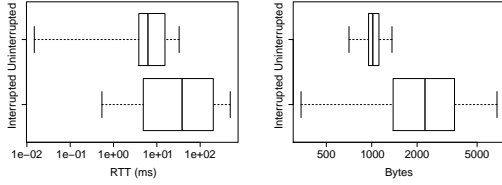


Figure 1: User-interrupted HTTP POSTs in Macha.

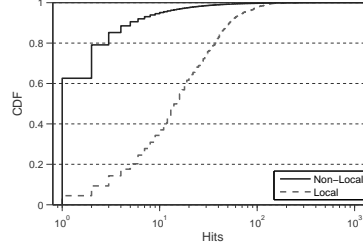


Figure 2: Local vs non-local Facebook image interactions in Macha.

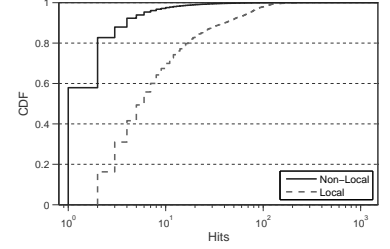


Figure 3: Local vs non-local Instagram media interactions in TDV network.

location of YouTube and Facebook content and circuitous routes to and from the rural ISP. Note that the relatively large number of reported RTT values less than 10ms are due to HTTP errors such as request timeout and malformed requests that were rejected by a local, non-caching Internet proxy, leading to LAN-speed RTT values.

2.1.2 Upload interruptions

To better understand how upstream bandwidth limits impact users' ability to contribute content to the Web, we analyze user behavior in regards to canceling flows using the metric presented in [9], where a flow is determined to be 'interrupted' if all of the following conditions hold: (1) A TCP RST is sent from the client to the server; (2) no TCP RST or FIN is sent from the server to the client; and (3) the time gap between the last data segment from the server and the actual flow end is less than or equal to the average RTT of the flow.

We focus on uploads by only investigating HTTP POST traffic; we compare RTT and bytes transferred between interrupted and uninterrupted HTTP upload flows. The Macha network traces include 9,324,212 POST flows, of which 61,232 were classified as interrupted. While this represents a small percentage of all POST flows, further analysis reveals these flows have significantly different characteristics, including bandwidth consumed, than uninterrupted flows. Figure 1(a) shows the measured TCP round trip time for interrupted flows is measurably different than flows that were allowed to complete. As shown, interrupted flows attain larger RTT values on the whole. The median for interrupted flows is 38.27ms while the median for uninterrupted flows is 6.23ms. Also of note is the difference in the upper quartile values between the two classifications. 25% of interrupted flows were longer than 200.99ms, while the longest 25% of uninterrupted flows were longer than 15.39ms. These results lead us to conclude that RTT plays a significant role in whether the user will cancel an upload or allow it to finish.

Figure 1(b) shows a box plot for bytes transferred over interrupted and uninterrupted sessions. We find that interrupted flows tend to have included more bytes than uninterrupted flows, though with a wider variance. Generally, we see that poor network connectivity, such as what we have witnessed in the Macha network, results in user frustration (and ultimately the user canceling a session) during large transfers. These discoveries motivate a system in which large file transfers are avoided across the constrained gateway link.

2.1.3 Facebook media locality

We analyze the locality of interest for Facebook media to discover potential benefits of localizing uploaded content.

We parse the traces to find the FBIDs for locally-uploaded Facebook images. To map FBIDs to Facebook uploads we employ a heuristic, where we search for image POSTs followed closely by an image GET, which includes the FBID, as the Facebook client downloads the image in order for the user to add metadata such as tags or friend identification. Using this information we are able to build a table of all Facebook image downloads for the year and determine whether those images were locally-created.

We examine the number of downloads per image for both local and non-local images. We define 'Local' as media that was uploaded by a local user and 'Non-Local' as media observed in the trace that did not originate in the local network. As shown in Figure 2, locally-generated images have a considerably higher hit-rate than those coming from external sources. We observe a mean hit-rate of 23.279 for local images, while non-local images are hit an average of 3.813 times. This high locality of content bolsters the argument for caching locally produced content at the network edge. The users most likely to download media are neighbors and friends of the user that uploaded the content.

2.1.4 Tribal Instagram traffic

We also investigate Instagram traffic on the TDV network in order to illustrate locality of interest. Prior work has shown Instagram to be the most popular service in the TDV network [10]. Additionally, other services such as Facebook and Google have recently moved to SSL encrypted sessions, making detailed header analysis in this trace impossible.

Similar to our Facebook analysis, we find ID values for all Instagram media (classified as public) uploaded during the observation period and compare IDs for subsequent downloads of content. Figure 3 shows the number of hits for locally-created media versus non-local media. As with our previous results for Facebook, we observe much more interest in locally created content compared to non-local content.

Next, we look at 'lifetime' of local and non-local Instagram media. We define lifetime as the time delta between the first and last observations of a media object in the traces. Similar to the hits observed, Figure 5 shows locally-produced media experience much longer lifetimes compared to non-local media. This result further propels the argument for keeping uploaded media cached at the network edge, even in well-connected networks.

2.2 Potential cache benefits

We illustrate the potential benefits of localizing content by observing the bandwidth burden and latency caused by the web services we target.

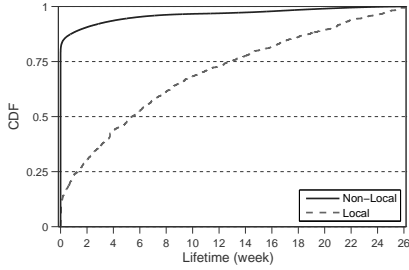


Figure 5: Lifetime for Instagram media in TDV network.

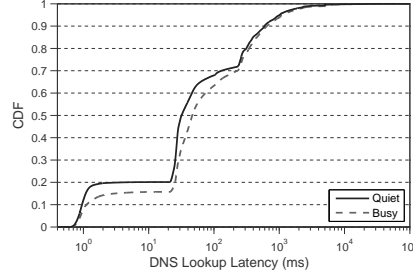


Figure 6: Macha DNS latency comparison.

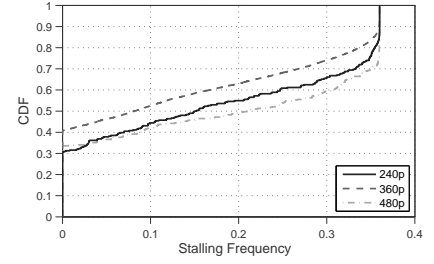


Figure 7: Macha YouTube QoE.

	Flows	GB	% Flows	% Bytes
YouTube	48,223	46.224	1.65%	36.00%
Facebook	571,530	9.156	19.61%	7.13%
Twitter	17,802	0.214	0.61%	0.17%
'Other' POST	202,857	1.375	6.96%	1.07%
'Other' GET	2,074,009	71.441	71.16%	55.64%

Table 1: Macha network traffic characteristics.

2.2.1 Service-specific network load

We analyze all web traffic to explore the bandwidth usage in the Macha network and find that the top two services in terms of bytes consumed are YouTube and Facebook, with a long-tail distribution of other Internet sites. As shown in Table 1, roughly 43% of all HTTP bytes are associated to either YouTube or Facebook, with YouTube leading all web sites with around 36% of all bytes. The table also shows Twitter statistics as Twitter is a representative example of the sites falling in the long-tail. Of note is the relatively small number of flows associated with YouTube traffic, yet the large percentage of bytes associated with the service. This makes video traffic an obvious target for a localization since those flows represent very large per flow byte transfers. Facebook traffic exhibits the opposite behavior, with many flows and comparatively fewer bytes. Further analysis revealed this was mainly due to lightweight Facebook IM traffic and a chatty web API. We believe Facebook videos and images remain valuable targets for VillageCache as they represent heavy bandwidth burdens on the network given their size and the overall popularity of the service.

2.2.2 Effect of YouTube and Facebook traffic on overall network performance

Large TCP flows impact other flows' attainable throughput. We investigate the impact of YouTube and Facebook media TCP flows on the overall network performance. We focus on one month of traffic, April 2011, and categorize time intervals into two types: intervals during which YouTube or Facebook flows were present, and intervals with no such traffic. Using the discovered time intervals we categorize DNS flows. 'Quiet' flows are DNS queries that began during times when there were no YouTube or Facebook flows. Conversely, 'Busy' flows are DNS flows whose first frame was detected during YouTube or Facebook activity. Over the course of one month, we detect roughly 1.2M 'Busy' DNS flows and 800k 'Quiet' DNS flows.

We consider DNS lookup latency, which we define as the time between an original client request and a server answer, including valid as well as error responses. This approach is similar to previous work [11]. Since DNS traffic uses UDP as the transport protocol, latency measurements more accu-

rately illustrate network conditions such as queuing delay as messages are encapsulated in a single packet and sessions are connectionless. DNS lookup latency is also a critical network metric as DNS performance strongly impacts client web traffic performance; each HTTP object must be resolved from a human-readable URL to an IP address. Figure 6 is a CDF plot of the DNS latencies observed. From the figure, we can estimate that the local network includes a DNS server corresponding to latency values of roughly 1ms. We can also observe the first-hop DNS server which corresponds to latencies from 20ms to up to around 200ms. Longer latency values are likely the result of DNS cache misses at the first-hop. In this case the performance of Busy and Quiet flows converge as congestion represents a relatively small portion of the overall lookup latency. We observe that the presence of YouTube or Facebook flows appears to have a detrimental effect on network performance when using either the local or first-hop server. DNS lookups performed during 'Busy' times experience a median increase of roughly 20ms compared to DNS lookups made during 'Quiet' times. Given the UDP transport used by DNS traffic, we can posit that the differences are evidence of network congestion.

2.2.3 YouTube QoE

We observe the performance of YouTube in the Macha network to illustrate the poor quality of experience for users due to the constrained link. Prior work [12] has shown that the most important factor in quality of experience for YouTube videos is the phenomenon of video stalls.

We examine the videos detected in the trace files with regards to the frequency of stalling events. YouTube videos are encoded with variable rate codecs; however, we categorize the videos by resolution and use the corresponding average bit rate for comparison. Over the course of the year, Tstat identifies 51,232 requests for videos encoded at 360p (default) resolution, 1,857 240p requests, and 1,529 480p requests. Hoffeld et al. [13] finds that stalling frequency can be estimated for YouTube videos using the following equation, where x equals the V (video bit rate) divided by B (flow bandwidth):

$$F(x) = -1.09e^{-1.18x} + 0.36$$

The resulting stalling frequency value is defined as the ratio of the number of stalls and the duration of the video. Using this value we can estimate the maximum number of stalls that will occur given a video duration. Figure 7 shows a CDF of the stalling frequency values for videos observed in the traces. We see that videos encoded with all three resolutions perform rather poorly in the observed network. For example, the median stalling frequency for 240p videos

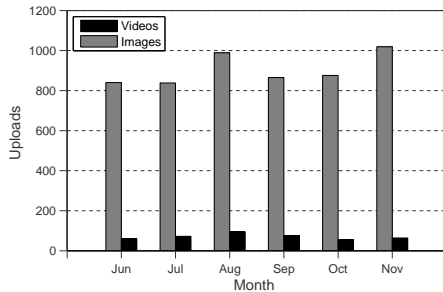


Figure 8: TDV network Instagram uploads.

encoded at 256kbps is roughly 0.2, given which the model estimates a maximum of 12 stalls per 60 seconds of video. Overall, we observe that the QoE for YouTube video in the Macha network is quite poor and as such the network provides one example of an environment that would benefit from localizing the media.

2.2.4 Instagram object duration

An important consideration for Internet caches is the duration that an object must be cached. We examine user interaction with Instagram media in the tribal network traces to find the time deltas between interactions on each media object. We assume a cache will use a least recently used scheme (LRU). Through our analysis we find that the time between hits is less than two weeks roughly 90% of the time with a median of less than one day. These results suggest that a caching system could be effective while retaining only a few weeks of content.

2.2.5 Instagram media storage

We compute the storage that would be required to cache locally-produced Instagram media observed in the TDV network traces. Figure 8 shows the number of images and videos uploaded each month in the tribal network. We see that uploads are rather steady across the observation period. We use the Instagram API to download all locally uploaded media objects to discover the total bytes uploaded for each month. We observe a monthly mean of 369.06 MB over the six month observation period. Given these findings, we believe that a cache deployed in networks similar to the observed networks would have attainable storage requirements.

3. VILLAGECACHE DESIGN

Our traffic analysis findings lead to the conclusion that a repository for locally uploaded media content at the network edge that can redeliver local content to mobile users without consuming Internet bandwidth would significantly ease congestion on the gateway link and offer improved performance. This particularly applies to Facebook and YouTube traffic as their associated traffic represents a large proportion of the bandwidth consumed in our Macha traffic analysis. Our design assumes that targeted networks are able to support fast local transfers and the Internet gateway is the bottleneck link. We envision the upload enabled cache will be placed at the Internet gateway, similar to our point of traffic collection in the observed networks, as such a location would typically allow for coverage of an entire rural community. Figure 9 gives an overview of the VillageCache system components.

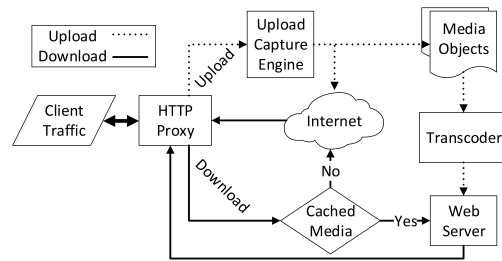


Figure 9: System architecture.

3.1 Upload capture engine

User uploads can be captured using multiple methods, depending on the desired user experience and level of acceptable intrusion. We explore possible directions for upload capture and discuss tradeoffs for each. All solutions result in a functionally equivalent system in which copies of uploaded media are kept locally and transformed in order to be transparently served back to users in subsequent requests.

3.1.1 Transparent upload scraper

The most desirable upload caching system is one that affords users the ‘native’ experience using web services. Ideally, the presence or absence of an upload cache is completely invisible to the user. Thus, we developed a transparent upload scraper mechanism. We chain a Squid proxy cache with a second proxy server running Tinyproxy on the same machine in order to transparently scrape file uploads. Only HTTP POST requests are forwarded to Tinyproxy while Squid services all other HTTP requests. We modify the Tinyproxy source to check the POST URL and content. If the POST is a Facebook or YouTube media upload, Tinyproxy saves the bytes contained in the POST request to a local file.

The transparent scraping method must include caveats including privacy and compatibility. First, as web services move toward SSL-encrypted traffic we must operate as an SSL man-in-the-middle using the Squid’s ssl-bump mechanism explained further in section 3.3.2. This obviously raises privacy concerns as our proxy would have access to users’ encrypted upload traffic. In terms of performance, transparent scraping does nothing to improve network performance during the initial upload. User uploads remain constrained by the Internet gateway link. Lastly, we build the scraper assuming HTTP sessions with the web services include an object identifier at the end of the upload in order to properly name the object in the local cache. This relies on the targeted web services to include such information in HTTP session traffic.

3.1.2 Custom local media upload portal

Improved performance can be attained if we are willing to sacrifice the native web service experience for user uploads. Since large file uploads during times of bandwidth saturation result in long sessions and frustration for all, we allow users intending to upload media to YouTube or Facebook to explicitly *time-shift* their transfers by uploading to a custom local portal. In order to time-shift uploads, we require user authorization to work on their behalf using OAuth. We gain access to user OAuth tokens through a local upload portal page in which users log in to their respective YouTube or

Facebook accounts and authorize the system. Users can upload video or images to the local page at LAN speeds as they avoid the gateway link. OAuth offers increased security versus locally saving user credentials, as the server only retains OAuth tokens and not clear-text credentials.

Upload time-shifting is accomplished by scripts that run during typically low bandwidth usage times such as late at night. The scripts use YouTube and Facebook API calls, along with the stored user OAuth tokens, to upload files without requiring any user interaction. We have targeted YouTube and Facebook for the initial design; however, similar techniques could be applied to any web service that uses OAuth authentication and an upload API.

The custom portal solution offers desirable performance and avoids bandwidth saturation via local transfers and time-shifting. However, users are required to explicitly participate through a non-native platform. Additionally, time-shifting results in a delay before the uploaded media are available via the web service. While these tradeoffs are substantial, we believe the solution may be acceptable in highly constrained networks or those prone to disconnection.

3.1.3 Custom HTTP POST request/response

Given the inherent privacy and future compatibility drawbacks of our transparent scrape mechanism, we submit a simple, powerful solution to allow for locally-uploaded media to be captured and appropriately transformed at the network edge for subsequent requests. We propose the introduction of optional HTTP headers associated with POST request / response pairs. Modern Internet proxy caches (e.g. Squid) are readily capable of performing specialized operations for specific HTTP requests and headers, making our solution feasible.

The POST request must specify that some part of the request is potentially cache-able media to signify to the cache that the bytes contained should be kept in memory until the server response is received. The corresponding POST response includes flags that indicate whether the media should in fact be cached, and specifies transformations that must be performed on the object, such as name, file type (e.g. jpg, png, mp4, etc), resolution, along with traditional cache headers such as timeout. Using this information, the cache system can convert the object to the appropriate format in order to enable subsequent cache hits. Such a system essentially mimics the media transformations that are done by the web services. Our proposed HTTP headers include:

POST Request

- **media-upload-cache:** Boolean indicating object is cache-able and should be temporarily kept by edge device.
- **media-upload-range:** Byte range of the cacheable object.

POST Response

- **media-upload-no-cache:** If present, discard object without placing in cache.
- **media-upload-resolution:** Resolution(s) to target with transcoded media.
- **media-upload-codec:** Codec(s) to transcode media.
- **media-upload-mime-type:** IANA MIME type for cacheable objects.
- **media-upload-name:** String to name object when placed in cache.

This solution is powerful because web services can change their media API without notice. A general solution is not possible without cooperation from the service. Since we argue for *optional* HTTP headers, service providers not interested in enabling upload caching can simply not employ the headers in their POST mechanisms. Services that do have an interest in improving user experience in poorly-connected networks can simply add the appropriate POST request and response semantics to their service.

3.2 Transcoder

We transcode uploaded media to multiple formats on the cache server in order to serve a broad range of client requests. We use the open-source tools `ffmpeg` and `imagemagick` to convert the original videos and images to formats with the same resolutions and codecs that web services use which are readily available [14]. This means we do not have a bit-for-bit copy of media objects, rather we create equivalent local objects to what is served by the web services. In the case of YouTube sessions, the client player requests the specific video format using the ‘itag’ value present in the HTTP request. We examine this value and provide the corresponding version of the video back into the HTTP/HTTPS session from the local server. As formats offered by YouTube evolve, we anticipate that `ffmpeg` will maintain the ability to produce files using the same codecs and container formats. Facebook images are converted to both WebP and JPEG formats of multiple sizes while videos are converted to MP4.

3.3 HTTP proxy

In order to redirect HTTP traffic to local resources when appropriate, we require a proxy that allows for analysis and modification of HTTP requests. For the prototype, we use Squid 2.7 running in transparent mode to provide both proxying and caching services to the local network. In order to function properly, the system must be placed at a location in the network hierarchy where all traffic destined for and returning from the targeted Internet services will pass through. This can be accomplished in a number of ways, including the use of firewall rules, or by placing the proxy inline with the gateway traffic. We chose Squid because it is widely deployed, well documented, and straightforward to alter in the case of necessary future changes. Lastly, Squid allows for HTTPS traffic proxying and redirection with the `ssl-bump` extension, a feature not widely available in other proxies.

3.3.1 Redirection to local media copies

Squid allows for HTTP redirection where a listener process examines HTTP requests and chooses whether or not to modify them. We leverage this functionality by building a custom redirector that determines whether or not a request is for cached media. In the case of Facebook images and videos, the ‘FBID’ contained within the query string is used. YouTube requests are parsed to find the ‘ID’ of the requested video. The redirector searches the local web server to check whether there is a local copy available in the appropriate format for the requested file. If there is a local copy the URL request is altered to use the local web server; otherwise the URL request is passed on to the remote server unmodified.

It is important to note that the redirection engine only redirects image and video file requests and nothing else. A

typical Facebook or YouTube HTTP session includes many requests for all of the objects on the page, including HTML, thumbnails, CSS, etc. These other objects are still retrieved from Internet servers rather than any local cache. While this consumes bandwidth, these objects are small in comparison to media content. Importantly, this behavior allows our solution to maintain content privacy settings for both services. For example, if a user does not have permission to access a YouTube video, the HTTP request does not include the video ID in the HTTP header. Instead, the request is of an image showing the user that they do not have the appropriate permissions to access that file. This functionality remains when our system is used, thus preserving YouTube’s privacy settings.

3.3.2 HTTPS proxying

Many online services, including YouTube and Facebook, have recently moved toward SSL-encrypted HTTPS rather than unencrypted HTTP. This trend toward encrypted connections will continue and we anticipate that eventually most, if not all, Internet traffic will utilize HTTPS. To that end, we use the `ssl-bump` extension for Squid in which the proxy intercepts HTTPS sessions and examines them in a manner similar to unencrypted traffic. In order to make this work, clients are dealt certificates from our proxy machine as the server dynamically generates certificates for clients as connections are created. It is important to note that this technique is required for our prototype system only and not meant for a production environment. We feel this feature is valuable despite the potential increased visibility to users. If desired, `ssl-bump` can be disabled altogether by changing a single line in the Squid configuration file.

3.4 Web server

In order for local redirection to function, we must store media files locally and make them available via a web server. As shown in Figure 9, the web server delivers local content back to the proxy during browsing sessions. We use the Apache web server as it allows for streaming modules for various video formats and is widely deployed. Also, as streaming formats continue to evolve it is likely that Apache will remain an early target for compatibility.

4. EVALUATION

We construct VillageCache in a lab environment to assess its performance and the efficacy of localizing content distribution at the network edge. We use a server running Debian Linux 7.0 with Squid and connect to the Internet through the server with various mobile clients including Android tablets and phones via 802.11g/n WiFi. We place a machine upstream from the cache server in order to emulate the Macha network using `netem` and `tc` to throttle the link bandwidth as well as introduce latency similar to what we observed in Section 2.1.

4.1 File transfer time

We first explore file transfer times. We emulate a bottleneck 2Mbps link with an average RTT of 134ms that is uniformly distributed with a maximum variation of 20ms. We execute all file transfers 10 times with no competing background traffic and compare mean completion times.

The main goal of our system is to provide users in resource-poor networks the opportunity to fully participate in the

modern Internet by improving media content performance. We compare HTTP GET times for various file sizes in two network scenarios to better illustrate the vast performance penalty when connecting via a constrained link. The first scenario consists of a client connected to cached resources via a local WiFi LAN. In the second, client traffic must go through the bottleneck link with parameters derived from prior packet analysis, as described above. We use varying file sizes up to 2GB. As expected, Table 2 shows that locally-stored web objects in poorly-connected networks are transferred in a fraction of the time it would take over a constrained link. Moreover, even moderate size file transfers across the constrained link take longer to complete than we can reasonably expect users to wait. Lastly, Internet cafés in developing regions often charge for Internet access based on time online rather than bytes used. In such situations it is clearly beneficial to localize media wherever possible.

4.2 YouTube

We evaluate the quality of experience for locally cached YouTube videos. These experiments were completed before a change in the HTTP object naming scheme utilized by YouTube. The current operation of YouTube makes object caching impossible, which led us to proposing the use of optional HTTP headers. Nevertheless, we present our experimental results to show the clear potential for caching such large media.

In our experiment automated browsers watch six 5-minute YouTube videos with different resolutions and video codecs passing through a constrained link. We vary the number of concurrent clients between 1 and 6. For each examined resolution (240p, 360p, and 480p) we run 10 sequences of 1 client, then two concurrent clients, and so forth up to six concurrent clients. The total is 210 videos viewed for each resolution with the performance analyzed in aggregate. We additionally run the experiment for 360p videos with 1Mbps UDP background traffic on the bottleneck link to examine the effectiveness of VillageCache during times of congestion.

As shown in Figure 10, the stalling frequency for each of the transfers is extremely low, particularly in comparison with Figure 7, resulting in stall-free playback for locally connected users. These results illustrate the dramatic improvement of the user experience when consuming files stored at the network edge rather than those stored in distant data centers.

4.2.1 Bandwidth savings

Bandwidth is a scarce and valuable commodity in poorly-connected networks. Our proposed system eases the heavy burden of media by avoiding the costliest transfer during a session, the media file itself. We compare the bandwidth consumed on the Internet gateway link by a client connecting without the use of VillageCache versus that consumed by a cache-enabled client using proxy logs to show requested files and the corresponding response size in bytes. As shown in Figure 11, the bandwidth savings witnessed for cached videos is strikingly large.

These savings are what we would expect from a typical web object cache. However, a system such as ours includes additional benefits that are particularly critical to poorly-connected networks. First, cached locally uploaded videos *never* need to be downloaded from the web as they are immediately placed into the cache at the time of upload. Typi-

Size (MB)	Local LAN (s)*	2Mbps Link (s)†
1	0.31	30.68
5	1.12	126.33
25	5.47	641.28
100	20.44	1,525.13
500	109.88	5,806.40
1,000	226.96	11,514.54
2,000	576.11	21,973.54

* Performance with cache

† Performance without cache

Table 2: Average file download times.

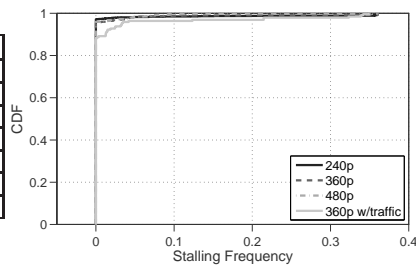


Figure 10: Cached YouTube QoE analysis.

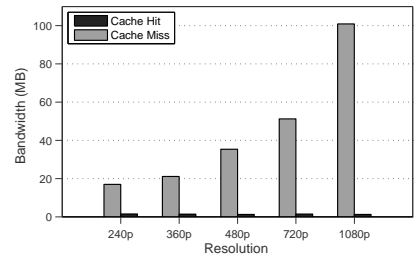


Figure 11: Cached video bandwidth savings.

cal web caches require the object to be downloaded from the source *at least once* to be properly associated with the corresponding web resource. As we see with the file transfer time measurements (Table 2) and user interruption of flows (Figure 1) behind a constrained link, even one large file transfer may be prohibitive for some networks. Our proposal allows users in these networks to participate in media-heavy web services, which was previously rendered impossible due to poor network conditions. Second, we have shown that users heavily consume locally uploaded media. This creates a multiplying effect for bandwidth saved in our evaluation; the actual bandwidth saved is much higher since local media will be disproportionately downloaded more often than non-local media.

5. RELATED WORK

In recent years there has been much research in the improvement of Internet experience and information dissemination for users in networks with poor connectivity [15, 16, 17, 18]. Caching techniques are often explored when confronted with a constrained link. Common approaches such as prefetching and aggressive caching can be built into browser plug-ins [15]. These types of systems help improve performance on individual computers. Our system proposal, on the other hand, is engineered with the cache placed in the network hierarchy where it can serve the entire community rather than individual machines. RuralCafe [19] is designed to allow for web searching over intermittent links in developing regions. TEK [20] also focuses on enabling web search from poorly connected networks using email. We have somewhat similar goals; however, we target a different context in media upload and consumption rather than web search.

Cloudlets [21] have been proposed to reduce delay, particularly for mobile clients. However, it is assumed that the cloudlet computer or cluster is well-connected to the Internet. We approach the problem assuming poor connectivity to the Internet from the local network.

Our work is informed by VillageShare [22], where content locality and time-shifted uploads in a rural African network are explored. We leverage the fact that content produced in poorly-connected networks is heavily consumed by users within them. However, where VillageShare was conceived as a custom local filesharing application, our vision is to make common, unmodified applications work in the absence of strong network connectivity. In our design, interactions are indistinguishable with sessions where content comes directly from the Internet service from the user perspective. This behavior is intentional and based on the notion put forth by Wyche [23], aiming for “ICT interventions grounded in

users’ existing practices rather than introducing new and unfamiliar ones.”

Lastly, Google and Facebook have recently proposed high-profile projects whose aim is to provide Internet connectivity to people in the world’s most remote places using technologies such as balloons or drones. Our work is complementary and still beneficial in such environments as caches help most with constrained (e.g. high latency, low bandwidth, high loss, etc) links to the Internet, a likely result of such projects. Connecting the most remote places on Earth will require a multi-pronged approach; localizing content where appropriate is one of many potential avenues for exploration.

6. DISCUSSION AND CONCLUSION

In this paper we have presented analysis of media consumption in two real-world networks to demonstrate the potential for localizing Internet media content in poorly-connected networks. Our study concludes that users access media-rich sites even with poor connectivity but are discouraged from interacting with media content by network conditions. Using our findings we propose a simple solution to cache locally uploaded media at the network edge so as to avoid the requirement for even a single download for the object to be available to other local network users. We evaluate a system prototype of VillageCache in a lab environment and show that such a cache can drastically improve the user quality of experience when accessing local content as well as ease the load on the Internet gateway link.

Aside from improving the end user experience, our proposal is mutually beneficial for the content providers as it offloads low-bandwidth, slow-moving flows from their data centers. As more service providers look to gain users in the most remote places, alternative content distribution techniques are likely to be a major consideration.

Network-level localization of Internet media lends itself to multiple avenues of future research. Given the nature of poorly-connected networks, a delay-tolerant implementation of the system should be explored. An eventual consistency model could be implemented where local upload requests made during times of congestion could be time-shifted and uploaded to the cloud at a later hour when there is less traffic. Web services typically allow for OAuth authentication for API calls, allowing for a middlebox to gather local uploads and move them into the cloud on behalf of the users in a secure manner. Such a solution would alleviate the issues surrounding the initial content upload that we observed in Section 2.1.

Another interesting area for exploration is peering between locally-deployed caches. Network-level localization is

convenient for system design, however different distribution models could be investigated leveraging social network analysis across multiple local installations. ‘Similar’ networks could then pre-share content amongst themselves in order to preemptively localize content that is likely to be locally requested. A final example, continuing with the idea of multiple deployed systems in a region, is a mobile app that could act as a data mule for copying local files to other local systems when the user migrates between them.

It is critical to enable full participation on the Internet by all users, particularly those whose cultures are at risk of marginalization. To that end, we hope network-level localization of locally-produced content provides a model that enhances user quality of experience while minimizing reliance on constrained Internet gateway links. In order to provide content to people in the most distant reaches of the world, services must rethink content placement and distribution strategies. Allowing decentralization can enable usage from networks that previously could not support full Internet participation.

7. ACKNOWLEDGMENTS

This work was funded through NSF Network Science and Engineering (NetSE) Award CNS-1064821 and the U.S. Army Research Laboratory and the U.K. Ministry of Defence under Agreement Number W911NF-06-3-0001. We would like to thank our partners in Tribal Digital Village for assisting us with traffic collection. We thank Barath Raghavan for comments and assistance in improving the manuscript.

8. REFERENCES

- [1] E. Brewer, M. Demmer, M. Ho, R. Honicky, J. Pal, M. Plauche, and S. Surana, “The challenges of technology research for developing regions,” *IEEE Pervasive Computing*, vol. 5, no. 2, pp. 15–23, 2006.
- [2] N. M. Dauenhauer and R. Dauenhauer, “Technical, emotional, and ideological issues in reversing language shift: Examples from southeast alaska,” *Endangered languages: Current issues and future prospects*, pp. 57–98, 1998.
- [3] M. Calder, X. Fan, Z. Hu, E. Katz-Bassett, J. Heidemann, and R. Govindan, “Mapping the expansion of Google’s serving infrastructure,” in *IMC ’13*, (Barcelona, Spain), 2013.
- [4] A. Gupta, M. Calder, N. Feamster, M. Chetty, E. Calandro, and E. Katz-Bassett, “Peering at the internet’s frontier: A first look at ISP interconnectivity in Africa,” in *Passive and Active Measurement*, (Los Angeles, CA), 2014.
- [5] K. Burnham, “Facebook overhauls android app for emerging countries.” <http://www.informationweek.com/software/social/facebook-overhauls-android-app-for-emerging-countries/d/d-id/1278740>, 2014. [Online; posted 20-June-2014].
- [6] V. Shankar, “Announcing Facebook lite.” <http://newsroom.fb.com/news/2015/06/announcing-facebook-lite/>, 2015. [Online; posted 4-June-2015].
- [7] M. Mellia, A. Carpani, and R. L. Cigno, “Measuring IP and TCP behavior on a edge node,” in *Globecom ’02*, (Taipei, Taiwan), 2002.
- [8] V. Paxson, “Bro: A system for detecting network intruders in real-time,” *Computer Networks*, vol. 31, pp. 2435–2463, Dec. 1999.
- [9] D. Rossi, M. Mellia, and C. Casetti, “User patience and the web: a hands-on investigation,” in *Globecom ’03*, (San Francisco, CA), 2003.
- [10] M. Vigil, M. Rantanen, and E. Belding, “A first look at tribal web traffic,” in *WWW ’15*, (Florence, Italy), 2015.
- [11] J. Jung, E. Sit, H. Balakrishnan, and R. Morris, “DNS performance and the effectiveness of caching,” *IEEE/ACM Transactions on Networking*, vol. 10, pp. 589–603, Oct. 2002.
- [12] T. Hoßfeld, M. Seufert, M. Hirth, T. Zinner, P. Tran-Gia, and R. Schatz, “Quantification of YouTube QoE via crowdsourcing,” in *ISM 2011*, (Dana Point, CA), 2011.
- [13] T. Hoßfeld, R. Schatz, T. Zinner, M. Seufert, and P. Tran-Gia, “Transport protocol influences on youtube videostreaming QoE,” *University of Würzburg, Institute of Computer Science, Tech. Rep*, 2011.
- [14] Wikipedia, “Youtube — wikipedia, the free encyclopedia.” https://en.wikipedia.org/wiki/YouTube#Quality_and_formats, 2015. [Online; accessed 19-September-2015].
- [15] J. Chen, D. Hutchful, W. Thies, and L. Subramanian, “Analyzing and accelerating web access in a school in peri-urban india,” in *WWW ’11*, (Hyderabad, India), 2011.
- [16] J. Chen, R. Power, L. Subramanian, and J. Ledlie, “Design and implementation of contextual information portals,” in *WWW ’11*, (Hyderabad, India), 2011.
- [17] M. Chetty, D. Haslem, A. Baird, U. Ofoha, B. Sumner, and R. Grinter, “Why is my Internet slow?: making network speeds visible,” in *CHI 2011*, (Vancouver, Canada), 2011.
- [18] S. Isaacman and M. Martonosi, “The c-link system for collaborative web usage: A real-world deployment in rural Nicaragua,” in *NSDR 2009*, (Big Sky, MT), 2009.
- [19] J. Chen, L. Subramanian, and J. Li, “Ruralcafe: web search in the rural developing world,” in *WWW ’09*, (Madrid, Spain), 2009.
- [20] W. Thies, J. Prevost, T. Mahtab, G. Cuevas, S. Shakhshir, A. Artola, B. Vo, Y. Litvak, S. Chan, S. Henderson, *et al.*, “Searching the world wide web in low-connectivity communities,” in *WWW ’02*, (Honolulu, Hawaii), 2002.
- [21] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, “The case for vm-based cloudlets in mobile computing,” *IEEE Pervasive Computing*, vol. 8, no. 4, pp. 14–23, 2009.
- [22] D. L. Johnson, V. Pejovic, E. M. Belding, and G. van Stam, “VillageShare: Facilitating content generation and sharing in rural networks,” in *ACM DEV 2012*, (Atlanta, GA), 2012.
- [23] S. P. Wyche, A. Forte, and S. Yardi Schoenebeck, “Hustling online: understanding consolidated facebook use in an informal settlement in Nairobi,” in *CHI 2013*, (Paris, France), 2013.