Celleration: Loss-Resilient Traffic Redundancy Elimination for Cellular Data

Eyal Zohar*
Electrical Engineering
Technion
Haifa, Israel
eyalzo@tx.technion.ac.il

Israel Cidon
Electrical Engineering
Technion
Haifa, Israel
cidon@ee.technion.ac.il

Osnat (Ossi) Mokryn†
Computer Science
Tel Aviv Academic College
Tel-Aviv, Israel
ossi@mta.ac.il

ABSTRACT

In this paper we present Celleration, a novel gateway-to-mobile Traffic Redundancy Elimination (TRE) system, designed for the new generation of data-intensive cellular networks.

Cellular TRE needs to account for the mobile device's limited battery power and the characteristics of the cellular network such as users' mobility, high packet-loss and long round-trip delays.

Celleration is based on a novel TRE technique, in which the cellular gateway observes the forwarded chunks to identify the beginning of a previously observed chunk chain, which in turn is used as a reliable predictor to multiple future chunks. These predictions establish an ad-hoc gateway-to-mobile TRE learning mechanism that leverages the gateway's history records and the user mobile device's cached content for an efficient TRE operation for both the backhaul and the wireless last-mile.

We present a data analysis of captured cellular traffic from 130 cellular sites and a long-term study of a social network. Finally, we analyze Celleration redundancy elimination and performance under high packet loss.

Categories and Subject Descriptors

C.2.m [**Computer-Communication Networks**]: Miscellaneous

General Terms

Algorithms, Design, Measurement

1. INTRODUCTION

The vast proliferation of smartphones and cellular tablets has brought a dramatic data traffic increase to cellular operators' networks [6]. Although Cellular bandwidth demand is traditionally associated with the last-mile air-interface, it is evident today that backhaul traffic is taking a growing share of the network cost. Yankee Group [13] estimates that the backhaul accounts for as much as 30% of a mobile operator's operating costs, and predict that the mobile network operators will face a massive investment in backhaul infrastructure. Emerging complementary solutions involve limiting users' data plans [1] and reducing traffic with various software solutions [17, 10, 5].

In this paper, we leverage Traffic Redundancy Elimination (TRE) to reduce the bandwidth consumption at both the backhaul and the wireless last-mile. Traffic redundancy stems from common end-users' activities, such as repeatedly accessing, downloading, distributing and modifying identical or similar information items (documents, data, web and video). TRE eliminates the transmission of redundant content. Hence, when an object is requested, the TRE sender does not transmit the parts already residing on the receiver's side. The TRE receiver, correspondingly, reconstructs the object using the newly transmitted parts as well as those already residing on its side. This method enables network cost reduction and data transfer speedup. Recent measurement studies have shown that the mobile environment is beneficial for caching and TRE solutions due to a highly predictable mobility pattern [18] and homogenous behavior [9].

In common TRE solutions, prior to the transmission of the data chunks, which are data parts parsed according to the data content, both the sender and the receiver examine and compare the chunk signatures. When redundant chunks are detected, the sender replaces the transmission of each redundant chunk with its strong signature [7, 15]. Most existing TRE systems are optimized for enterprise branches but are less efficient in a mobile environment, in which the client is detached from a fixed location. These systems are not designed to address high packet loss, unstable latency and unpredictable user bandwidth associated with the cellular infrastructure.

We present a novel TRE solution tailored for cellular networks, termed *Celleration*, which eliminates individual clients' redundant traffic across the cellular network. In this solution, a TRE gateway, located at the cellular network Internet entry-point, leverages the aggregated information transmitted through the gateway to enable a smart delay-free TRE. Using locally gathered information, the gateway detects similarities in repetitive chunk flows and performs an out-of-band ad-hoc learning against the mobile devices for possible elimination of future chunk transmissions. The gateway refrains from sending data which the ad-hoc learning mechanism finds to be residing at the receiver's end, while other data is forwarded with no delay. Thus, the gateway reduces the traffic over the path to the mobile device and speeds up the overall performance.

In contrast to existing server-based solutions, Celleration does not require the gateway to cache data, continuously track the devices storage state or to delay data for ad-hoc synchronization; instead, the gateway records only the short chunk signatures and learns their sequencing likelihood. The gateway then uses the ag-

HotMobile'12, February 28–29, 2012, San Diego, CA, USA.

*Also with Hasso-Plattner-Institute Research School at Technion.
†Also with Technion - Israel Institute of Technology.

gregated cross-user knowledge to predict individuals' future data and enable bandwidth savings for end-users.

We have conducted extensive data analysis and experiments that motivate the novel design point of Celleration. First, in Section 3 we explore the individual end-user redundancy and show a high redundancy ratio. More specifically, the data comprised from 130 cellular web sites and 17,000 users indicated 46% individual redundancy. In addition, we investigate the effect of retaining long-term memory in mobile devices on TRE performance and show that a TRE with 3-days memory achieves about 48% bandwidth saving compared to only 27% for existing short-term solutions [4][2]. We then proceed to show that a TRE which supports changes in client IP addresses enhances the bandwidth savings by 18%. Finally, in Section 5 we show that Celleration achieves high redundancy elimination even in the face of high packet-loss.

2. MOTIVATION AND RELATED WORK

In all TRE solutions, the sender needs to identify data parts which are locally accessible at the receiver, and eliminate their transmission. To this end, most current TRE solutions employ one of the following techniques:

1. Full synchronization - maintain full synchronization between the sender and receivers so that the sender has a consistent view of the receivers' storage.
2. Preliminary negotiation - the sender, before forwarding the data, checks with the receiver whether this data already resides at the receiver.
3. Client prediction - the client, upon receiving known data, sends the sender predictions for future data whose transmission is redundant.

The full synchronization approach is common in commercial TRE solutions [11, 14] that utilize proprietary middle-boxes placed at both ends of the communication path. The major drawback of the full synchronization technique in the mobile environment is that it requires either a continuous connection between the gateway and the mobile device or a massive state synchronization when a new connection is established. The continuous synchronization of both ends has also been adopted by several academic solutions [4, 2]. The above mentioned solutions require the server to maintain a fully and reliably synchronized data cache for each client. To adhere with the server's memory requirements, these caches are kept small (around 10 MB per client), thus making the system inadequate for long-term redundancy.

The preliminary negotiation approach is suggested in the early LBFS work [15] as well as in Wanax [7], which is a TRE system tailored for the developing world where WAN bandwidth is scarce. The major drawback of these solutions is the need of buffers and the added transmission delay.

The client prediction approach, termed PACK [20], serves as the departure point for Celleration. In PACK, according to currently received data and the history of data reception, the client sends predictions to cloud servers concerning redundant data that is about to be sent from the server. If the prediction is correct, the server eliminates the transmission of the corresponding data. PACK is designed to reduce cloud costs by offloading the TRE effort from the cloud to unutilized desktop clients. When dealing with cellular networks and sites, Celleration has several advantages over PACK's end-to-end scheme:

1. Client prediction requires more computational efforts on the client's side, while Celleration considers the limited power of cellular devices by shifting the chunking and signing efforts to the network operator's equipment.
2. Cross-user knowledge aggregated at the gateway keeps Celleration updated with recent content changes and relations between distinct objects. These help the gateway to predict future data based on historical data and meta-data.

A recent study [12] further motivates the introduction of a novel cellular solution. According to this research, end-to-end solutions that require full synchronization are ineffective in lossy mobile cellular networks. The loss recovery scheme offered by [12] improves the full synchronization approach but cannot prevent the added latency and some of the bandwidth waste due to lost packets.

3. DATA ANALYSIS

We present here a comprehensive analysis of traffic extracted across 130 cellular web sites and a social network site in order to evaluate the potential redundancy elimination in cellular networks. In addition, we analyze five leading news sites to apprehend the time behavior of content in frequently changed sites.

For redundancy computation we conservatively assume that each client starts with an empty cache. The chunking mechanism [20] uses an 8 bits anchor and minimal chunk size of 64 bytes, expecting an average chunk size of around 320 bytes.

3.1 Analysis of Traffic in a Cellular Gateway

We obtained a continuous 5 hours traffic recording of a cellular gateway in September 2011. The gateway connects 130 web sites to the cellular networks. The recording contains over 1.2 million HTTP sessions generated by 17,000 distinct cellular users. Each record has a device-level unique identifier that enables a reliable study of same-user activity. Overall, we found an average of 46% same-user redundancy in these 5 hours. This result is a conservative estimate of the amount of redundancy in cellular web traffic, because our calculation totally ignores chunks that already reside at the client's storage when the recording starts.

3.1.1 Redundancy by Content and Site Type

Traffic redundancy is caused by either repeating identical objects or self-similarity within content. Repeating objects are usually compressed multimedia like pictures, graphics, audio and video. Self-similarity usually appears in variations of textual content like HTMLs and documents.

Table 1 shows the redundancy according to file type. In addition, the table presents the average cross-user repeats per object, which does not directly save bandwidth but is related to the gateway prediction mechanism. GIFs are commonly used for graphic objects, such as logos, decorations and advertisements. JPEGs are commonly used for photography, so their size is larger and more variable than GIFs. Furthermore, many GIFs (about 10%) are shared between different sites. In addition, in most sites we found that HTML content was dynamic in time as further demonstrated in Section 3.3.

Table 2 shows that the redundancy varies from one site type to another. News sites have the lowest same-user redundancy, mainly due to changes in content over time and to non-repeating usage pattern. On the other hand, entertainment sites are relatively static

Table 1: Cellular sites dataset results: redundancy level by content

Content type	Variations	Same-user redun.	Cross-user downloads per object
GIF	None	72%	10 : 1
JPEG	None	33%	3.8 : 1
HTML	By device, time, personal, etc.	40%	2.1 : 1

Table 2: Cellular sites dataset results: self-redundancy by site type

Site type	Same-user redun.
News	38.7 %
Services	43.1 %
Shopping	45.5 %
Social network	51.5 %
Entertainment	61.0 %
Finance	63.0 %

and attract users to view the same item multiple times. Finance sites and social networks are very dynamic in nature, but the fresh content is repeatedly wrapped with similar content and graphics, which generates relatively high same-user redundancy.

3.1.2 Variety of Devices

Compared to the traditional web, cellular sites have to deal with many different browsers and displays. We found 1,262 different combinations of devices, operating systems, browsers, and supported technologies. These parameters have a major effect on such features of a site's presentation and usability as screen resolution, orientation, colors, and interaction methods.

Due to this variety, many cellular sites generate device-adapted content according to each site's technological maturity and business logic. This observation explains why traditional object-level cross-user HTTP proxies are inefficient in the cellular era; even if URLs were static, a proxy would not be able to tell in advance if the object being requested by a user were identical to another object previously downloaded by a different user with exactly the same URL.

3.1.3 Compression

We found that only 48% of the observed clients support object-compression, probably due to the limited computational power of mobile devices and/or the immaturity of mobile web browsers development. This stresses the importance of cellular server-side TRE as a means of reducing textual traffic. This traffic, mostly generated by HTMLs and documents, is commonly compressed today with gzip by all major non-cellular web browsers and web sites.

3.2 Social Network Dataset

We obtained the access log of a social network site for a period of 33 days at the end of 2010. The dataset enables a reliable tracking of returning users despite changes in the IP addresses, as all users identify themselves using a unique login ID to enter the site. This helps to measure the effect of long-term chunk caching on the TRE efficiency. It also enables the tracking of the device type and network operator by using the user-agent field and the client IP address.

Our first step is to measure the amount of per-user traffic redundancy by using the measurement scheme presented in [19]. Then, we analyze our findings and show why existing TRE schemes cannot efficiently eliminate this potential redundancy due to the cellular network circumstances. In addition, the fact that this site equally serves non-cellular users helps us to learn more about the differences between the two audiences.

3.2.1 Redundancy and Usage Patterns

In order to evaluate the redundancy elimination potential of different TRE schemes, we analyzed the usage patterns and content-similarity of all the social network dataset users and compared the results according to the device and network type.

Table 3 presents the results divided into four categories: PC (includes laptop) users over either a cellular or non-cellular network, and non-PC devices over either a cellular or non-cellular network. The measured redundancy is the percentage of chunks that were repeatedly transmitted to the client. We simulated three TRE schemes that differ in their ability to detect returning users. We found that in all schemes, cellular users had higher traffic redundancy, probably because cellular users are more likely to embrace a single end-device (e.g. personal smartphone) than non-cellular users. We also measured the number of page views per site-level session (from first page view to last) and found that it was lower for cellular users, possibly because of the limited user interface.

Table 3: Social network dataset results: usage patterns and redundancy elimination potential of 3 different TRE schemes

	Cellular		Non-Cellular	
	PC	Non-PC	PC	Non-PC
Page views per session	51.6	35.7	51.7	38.0
Multiple end-devices	4.4 %		11.8 %	
End-devices per user (avg.)	1.18		1.49	
Sessions per user (avg.)	7.56		10.36	
Intra-session redun.	33.4 %		32.3 %	
Inter-session, intra-IP redun.	52.0 %		50.4 %	
Inter-session, inter-IP redun.	67.7 %		64.4 %	

Note that while we found a clear case for TRE style caching, object level caching is inappropriate in this case due to the use of dynamic URLs. This social network, like many other modern sites, deliberately prohibits object caching to protect business interests such as privacy, content copyrights, etc.

3.2.2 Long-Term Caching

To assess the bandwidth saving potential of long-term caching, we measured the social network redundancy while changing cache sizes at the client-side. We set different cache sizes by changing the maximal age of data kept in the client's storage.

Figure 1 illustrates the redundancy level as a function of cache sizes. The upper line represents the redundancy for TRE schemes that identify returning clients despite changes of their IP addresses. The lower line is the upper bound for TRE schemes that do not track returning clients that change their IP addresses (e.g., [4][2]). Clearly, the larger the client's cache is, the larger is the amount of traffic that can be eliminated.

Further analysis shows that most users do not have multiple sessions within a single hour. In addition, the average redundancy within a single session (intra-session) is about 33% compared to an inter-session maximal daily redundancy of 77%. This means that a major part of the redundant data is found across sessions; hence, the cache needs to store multiple sessions over many hours. These findings showing that a TRE with one-hour cache saves at most 28% of the bandwidth match the results presented in Figure 1.

Figure 1: Social network, site traffic redundancy on the 4th day with different cache sizes

3.2.3 IP address changes

Several TRE schemes assume that the sender can identify the receiver by the packet level IP address. We found that cellular IP addresses are very dynamic. In particular, in the social network dataset examined, 67.1% of the cellular sessions used an IP address that was previously used by another user or device. This phenomenon was not observed in non-cellular networks, where only 2.2% of the IP addresses were reused. In addition, the social network login information revealed that in around 95% of successive connections of non-cellular users the same IP was used.

We also found that 62.1% of the devices that connected via a cellular network also connected via a non-cellular network. Some of these devices oscillated between cellular and non-cellular networks. Consequently, we conclude that an efficient TRE for cellular networks should not use IP addresses as the sole end-point identification.

3.3 News Sites

News sites are an interesting challenge for TRE mechanisms, as they present frequently changing content. We examined the data obtained from five leading news sites over a 24 hour period. The main page of each site was downloaded repeatedly 10 minutes apart. This of course does not represent a typical single user's surfing pattern. Nevertheless, we present a deeper inspection of the data in order to better understand the nature of incremental changes occurring between consecutive downloads.

Table 4 presents the analysis results of the news sites dataset. We found that more than 88% of the traffic generated from these sites over the 24 hour period is redundant.

Minor data changes were observed on each download while major changes occurred on average every couple of hours, with up to 22% new data. A common minor change is related to time strings or random tokens that appeared in the HTTP header, HTML or a script. Certain sites also rapidly changed the embedded advertisements.

4. CELLERATION

Following our findings in Section 3, we conclude that cellular networks require a TRE solution which can use the long-term data residing in the mobile device. At the same time, it should endure the latency which is added by several existing solutions.

In this section, we describe Celleration, which is a cellular TRE service that complies with our findings. Unlike previous solutions, Celleration is loss-resilient, i.e., it does not mistake TCP retransmissions of lost data with transmissions of redundant data which is already cached in the mobile device storage.

Figure 2 demonstrates the settings of the system. The TRE is performed between the packet data network gateway (*gateway*) and the mobile device. Celleration enables the gateway to predict future data to be sent, and eliminate its transmission in case it already resides in the mobile device's storage. Consequently, Celleration never delays traffic and does not utilize data buffering.

4.1 On-the-Fly Prediction

The gateway forwards all crossing data to the mobile device unless it has previously confirmed that the data already resides in the device. The gateway activates three mechanisms:

Table 4: News sites dataset results

	BBC	CNN	Yahoo	WPost	NYTimes
Total redun.	95.5 %	93.8 %	88.0 %	96.5 %	89.4 %
Avg. file size	93 KB	82 KB	124 KB	224 KB	188 KB

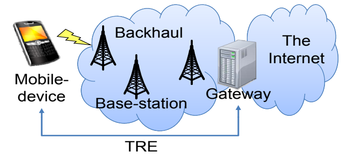


Figure 2: Celleration's basic components

1. Flow coding: The gateway continuously chunks the crossing flows using the algorithm of [20] and signs each chunk using SHA-1. It stores the chunks' signature sequences in its local cross-user signature store. In addition, each chunk's size and signature are sent as meta-data to the mobile device to spare the client the effort of chunking and signing as illustrated in Figure 3a.
2. Ad-hoc learning of mobile devices: When the gateway recognizes the crossing flow by the data's signatures, it looks up the cross-user signature sequences store for potential future data and inquires whether the mobile device has this potential data in its storage as illustrated in Figure 3b. We leave the details of the cross-user signature sequences store to future work because of page budget limitations.
3. Flow reduction: If the ad-hoc learning indicates that some data already resides in the mobile device, the gateway refrains from forwarding that data to the device when it arrives, thus reducing the transmission over the path to the mobile device as illustrated in Figure 3c.

Note that the gateway always forwards data that the mobile device has not yet acknowledged, keeping the operation delay-free.

4.2 Ad-Hoc Learning

The on-going ad-hoc learning between the gateway and the mobile device is activated upon need during the connection time. This gateway-initiated process, illustrated in Figure 3c, takes place as follows:

1. The gateway sends a list of signatures, each made of (serial, hash, size)
2. The mobile device returns a list of time-limited approvals (serial, time)
3. The gateway refrain from forwarding approved chunks, and sends control data instead (serial, TCP sequence)

The predictions are based on the most likely patterns derived from previous flows of all users. This scheme also enables the mobile device to derive the predicted chunks from alternative sources, such as sibling peers or a regional cache server.

The above described ad-hoc learning tightens the gateway and the device synchronization with time. This process adds only a small traffic overhead without requiring a permanent connection between the gateway and the device. This approach frees the gateway from delaying transmissions, which may be harmful for the TCP retransmission mechanism.

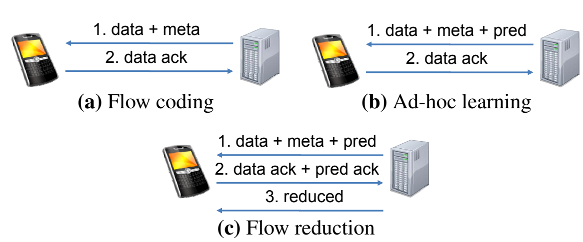


Figure 3: Celleration: operation outline

4.3 Flow Reduction

The above described basic flow reduction algorithm is illustrated in Figure 4a. In the more advanced mode termed the speculative reduction, the gateway may also send predictions when it is not sure if the crossing data chunks are stored in the mobile device. The speculative algorithm estimates the probability of these chunks being stored in the mobile device. This prediction is based on various parameters, among which there are the success history of the specific mobile device and chunk availability in other mobile devices. This speculative reduction is illustrated in Figure 4b.

5. EVALUATION

In this section, we evaluate Celleration paying special attention to the characteristics of the cellular network.

5.1 Implementation Details

Our implementation is application transparent at both the sender and the receiver. The receiver-sender TRE protocol is embedded in the TCP Options field for low overhead, similar to the published source code of [20].

The implementation maintains a basic signature store in the gateway by keeping for any chunk signature the last observed subsequent chunk signature in a LRU fashion. A future extension to this basic implementation is a more advanced prediction mechanism that would take into consideration potentially valuable parameters such as the content's origin server, time of day, etc.

The chunking mechanism uses an 8 bits anchor and minimal chunk size of 64 bytes, resulting in an average chunk size of around 320 bytes. We found this size to achieve high TRE level, adding an overhead of less than 3% per chunk prediction. Each prediction carries the lower 32 bits of the SHA-1 signature, the chunk length (12 bits), and a serial (8 bits). It should be noted that the lower 32 bits of the hash are sufficient for collision avoidance, because a collision might occur only if the following hold: (1) the gateway predicts chunk A. (2) the device has in its limited storage chunk $B \neq A$. (3) $lsb32(sha1(A)) = lsb32(sha1(B))$. (4) $len(A) = len(B)$.

5.2 Setup

Our simulation runs on Linux with Netfilter Queue [16]. At the gateway, we used an Intel Core 2 Duo 3 GHz, 2 GB of RAM and a WD1600AAJS SATA drive desktop. The client's side was a HTC Desire smartphone running Android 2.2.1. Since currently Android does not support Netfilter Queue, we ran the TRE client side on a laptop and connected the smartphone to the gateway through the laptop's WiFi.

5.3 Dealing with High Latency

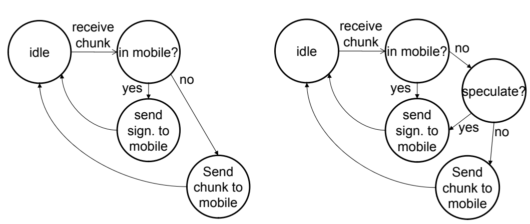
The problem with the high latency in today's cellular networks is a Bandwidth Delay Product (BDP) that is much larger than the initial TCP window sizes [8]. More specifically, with RTT of 200mSec and data rate of 2Mbps the BDP is 50KB, while a typical Android initial receiver window is below 6KB and many servers use an initial congestion window that follows the 4*MSS rule [3]. Under these circumstances, Celleration gateway may forward some of the redundant data before the mobile device's prediction acknowledgements reach the gateway. This forward operation keeps the operation delay-free but may decrease the TRE efficiency.

To measure the effect of the latency on Celleration, we simulated a symmetric long latency path of 200mSec RTT by delaying packets between the gateway and the mobile. Looking at the collected cellular traces, we found this RTT to represent today's networks, although we explored modern HSDPA traces that reach RTT as low as 70mSec. To this end, we evaluate Celleration with some of the data of Section 3.3.

Figure 5 presents 144 downloads from CNN's main page, downloaded repeatedly by a mobile device each 10 minutes over 24 hours. This workload may not represent a typical single user's surfing pattern, but it provides a controllable environment is which we demonstrate how the algorithm deals with high latency and frequent data changes. Each download was encoded in the gateway to about 220 chunks and compared with signatures of former data that flew through the gateway. Each vertical line in the graph represents a single download, where the graph's dots represent chunks that were completely new to the mobile device at the time of the download. The blank areas represent chunks that already were in the device's storage when the new page was downloaded, which was about 92% of each download's chunks on the average.

In this experiment, Celleration was able to eliminate 97% of the chunks that the mobile device already had. The other 3% were not eliminated but forwarded to the mobile device with no delay. We have analyzed the results and found two reasons for that:

- Latency - The first packets in each TCP session help the gateway to predict future data and are forwarded with no delay. Seemingly, the number of such forwarded packets may increase when the BDP increases. In practice, we have found that this number is bounded by TCP window sizes on both sides; the sender window and the initial receiver window. In this experiment, it was Android's 6KB initial receiver window that bounded the immediate forwards hit-ratio depression to less than 3%.
- False predictions - Another source for redundancy that was not eliminated was the presence of chunks that fluctuated between different versions of the page. We found that in each download, up to 12 existing chunks (average 2.5) were not a part of the previous download but had been recently seen in an older version of the page up to 60 minutes before. These findings indicate



(a) Basic mode **(b)** Speculative mode

Figure 4: Algorithm: gateway flow reduction

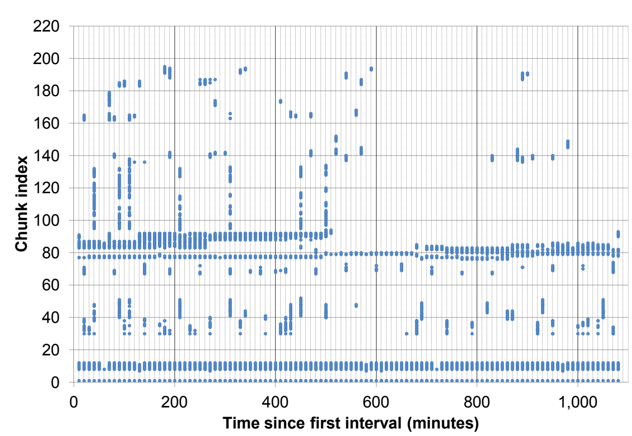


Figure 5: Visual presentation of new data in CNN's main page, downloaded repeatedly 10 minutes apart for 24 hours

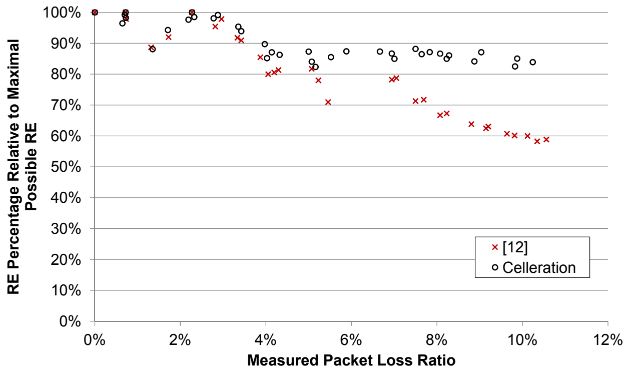


Figure 6: Redundancy elimination between the gateway and a smartphone, when the cellular network losses packets

that in some cases it may be beneficial to use signatures of frequent older versions, instead of the simple LRU mechanism we implemented.

5.4 Under High Packet Loss

To simulate a lossy network, packets were randomly dropped at the gateway. To evaluate Celleration, we compared it to [12] which enhances previous TRE solution to cope with packet loss. The enhanced scheme detects potentially lost packets and prevents the sender from eliminating their retransmission.

Celleration has an inherent packet-loss resilience mechanism due to the interactive ad-hoc learning mechanism. Chunks that do not reside already at the receiving end, are not acknowledged by the receiver, and therefore are not eliminated.

Figure 6 presents the redundancy eliminated by Celleration, compared to [12] given a lossy network with up to 11% packet loss. Celleration eliminates more than 80% of the redundancy at a 10% packet loss rate, while [12] achieves only around 60% elimination.

5.5 Battery Power

Celleration takes into account the limited power of cellular devices by shifting the chunking and signing efforts to the network operator's gateway. To justify this approach, we compared Celleration's power consumption with PACK [20]. To isolate the energy consumption of PACK's TRE operations, we turned off all communications in the smartphone and ran an application written for this mission. The application performs only the additional operations in which PACK client differs from Celleration client: it chunks a (random) data buffer and signs each chunk using SHA-1. We ran the application at an adjusted processing speed of 1.5 Mbps for 5 hours, in which the battery level decreased from 80% to 43%. By comparing these results to an idle phone, we concluded that in this scenario Celleration could save 30% of the battery power drain. It should be noted that we did not account for other TRE operations at the mobile device, such as storing and fetching chunks, as they are mandatory and common for all TRE solutions.

6. SUMMARY

In this work we have presented Celleration, a TRE designed for cellular networks. Celleration is tailored around the characteristics of the cellular network, devices and usage patterns to achieve a loss-resilient delay-free efficient operation. It eliminates a considerable amount of bandwidth at both the backhaul and the wireless last-mile of the network while preserving handset battery power.

In future work, we plan to explore efficiency and scalability issues related to the gateway's prediction mechanism.

7. REFERENCES

[1] Unlimited Data Plan: Is It Coming to an End? *International Data Corporation (IDC)*, October 2010. `http://www.idc.com/getdoc.jsp?containerId=AP3053409S`.

[2] B. Aggarwal, A. Akella, A. Anand, A. Balachandran, P. Chitnis, C. Muthukrishnan, R. Ramjee, and G. Varghese. EndRE: An End-System Redundancy Elimination Service for Enterprises. In *Proc. of NSDI*, 2010.

[3] M. Allman, S. Floyd, and C. Partridge. Increasing TCP's Initial Window. *RFC 3390*, 2002.

[4] A. Anand, A. Gupta, A. Akella, S. Seshan, and S. Shenker. Packet caches on routers: The implications of universal redundant traffic elimination. In *Proc. of SIGCOMM*, 2008.

[5] A. Balasubramanian, R. Mahajan, and A. Venkataramani. Augmenting Mobile 3G Using WiFi. In *Proc. of MobiSys*, 2010.

[6] S. Higginbotham. The Mobile Tsunami Is Near: Blame Netflix and Apple. *GigaOM*, January 2011.

[7] S. Ihm, K. Park, and V. Pai. Wide-area Network Acceleration for the Developing World. In *Proc. of USENIX ATC*, 2010.

[8] H. Inamura, G. Montenegro, R. Ludwig, A. Gurtov, and F. Khafizov. Tcp Over Second (2.5G) and Third (3G) Generation Wireless Networks. *RFC 3481 (BCP 71)*, 2003.

[9] R. Keralapura, A. Nucci, Z.-L. Zhang, and L. Gao. Profiling Users in a 3G Network Using Hourglass Co-Clustering. In *Proc. of MobiCom*, 2010.

[10] L. Le and E. Hossain. Multihop Cellular Networks: Potential Gains, Research Challenges, and a Resource Allocation Framework. *Communications Magazine, IEEE*, 2007.

[11] E. Lev-Ran, I. Cidon, and I. Z. Ben-Shaul. Method and Apparatus for Reducing Network Traffic over Low Bandwidth Links. *US Patent 7636767*, November 2009. Filed: November 2005.

[12] C. Lumezanu, K. Guo, N. Spring, and B. Bhattacharjee. The Effect of Packet Loss on Redundancy Elimination in Cellular Wireless Networks. In *Proc. of IMC*, 2010.

[13] J. M. Pigg. Mobile backhaul evolution. *Yankee Group Webinar*, April 2010.

[14] S. Mccanne and M. Demmer. Content-Based Segmentation Scheme for Data Compression in Storage and Transmission Including Hierarchical Segment Representation. *US Patent 6828925*, December 2004. Filed: December 2003.

[15] A. Muthitacharoen, B. Chen, and D. Mazi'eres. A Low-Bandwidth Network File System. In *Proc. of SOSP*, pages 174–187, New York, NY, USA, 2001. ACM.

[16] netfilter/iptables project. libnetfilter_queue, October 2005. `http://www.netfilter.org`.

[17] R. Pabst, B. Walke, D. Schultz, P. Herhold, H. Yanikomeroglu, S. Mukherjee, H. Viswanathan, M. Lott, W. Zirwas, M. Dohler, et al. Relay-Based Deployment Concepts for Wireless and Mobile Broadband Radio. *Communications Magazine, IEEE*, 2004.

[18] E. Rosensweig, J. Kurose, and D. Towsley. Understanding Traffic Dynamics in Cellular Data Networks. In *Proc. of INFOCOM*, 2011.

[19] N. T. Spring and D. Wetherall. A Protocol-Independent Technique for Eliminating Redundant Network Traffic. In *Proc. of SIGCOMM*, 2000.

[20] E. Zohar, I. Cidon, and O. Mokryn. The Power of Prediction: Cloud Bandwidth and Cost Reduction. In *Proc. of SIGCOMM*, 2011.