Mercado: Using Market Principles to Drive Alternative Network Service Abstractions

Xu Chen Jeffrey Erman Seungjoon Lee Jacobus Van der Merwe
AT& Labs–Research
Florham Park, NJ, USA
{chenxu,erman,slee}@research.att.com
School of Computing
University of Utah
kobus@cs.utah.edu

ABSTRACT

In this paper we propose the Mercado architecture as a vehicle to use economic incentives and the sophisticated capabilities of modern mobile networks and devices to change the mobile networking service abstraction to better utilize networking resources. Our proposed architecture generalizes to enable rich interaction and information exchange between mobile devices and the network. However, as a first step we focus our efforts on the scenario where a mobile device with non-real time network workload, would interact with the network to explore the financial incentives available if it were to delay using the network. Combining the network response with knowledge of the application semantics allows the application to perform its own economic vs utility tradeoff.

Categories and Subject Descriptors

C.2.1 [**Network Architecture and Design**]: Wireless Communication

Keywords

mobility policy application server, PCRF, delay-tolerant traffic

1. INTRODUCTION

The capabilities of mobile devices and the resulting demands they place on the network, continue to far exceed the capacity of mobile networks. The relative scarcity of wireless spectrum is at some level the fundamental root of this problem. However, we argue that at a more holistic level, the mobile networking service abstraction is also fundamentally contributing to the problem in that it has not adapted to the capabilities of modern mobile devices and the ways in which mobile networks are used.

Our specific proposal is based on a number of simple observations: (i) While real-time applications like voice, interactive video and messaging are still hugely important, mobile devices also enable many near-real time or non-real time applications such as email and movie or music downloads. (ii) Mobile devices are computationally sophisticated enough that different applications might reason about and interact with the network in different ways, depending on the "semantic needs" of the user. (iii) Modern mobile networks have sophisticated policy and charging mechanisms, which allows great flexibility in how network resources are utilized and charged for.

We propose the Mercado architecture as a vehicle to use economic incentives and the sophisticated capabilities of modern mobile networks and devices to change the mobile networking service abstraction to better utilize networking resources. Mercado aims to enrich the existing policy and charging mechanisms available in modern mobile networks with network derived intelligence to create new service abstractions and to directly expose such service abstractions to mobile devices and services. As a first step we focus our efforts on the scenario where a mobile device with non-real time network workload, would interact with the network to explore the financial incentives available if it were to delay using the network. Specifically, we expect that price differentiation, especially lower prices, will incentivize users and application developers to more carefully consider the actual network needs of their applications so as to optimize their economic self interest.

The incentive for mobile network providers to adopt this approach is manifested in two ways. First, delayed network use can be exploited by the provider to reduce traffic peaks during high demand periods. Such traffic smoothing can ease congestion and/or delay the need to upgrade network capacity. Second, the availability of a delay tolerant service abstraction might in fact increase network use for application that are inherently delay tolerant, thus resulting in new or enhanced revenue streams.

Fully exploring the financial implications in terms of revenues and costs are well beyond the scope of this paper. Instead, we attempt to answer a number of more modest questions: Are users willing to delay use of the network given financial incentives? Given the current network usage, would delayed network use for at least some applications provide any benefit to the network in terms of traffic reduction? Assuming our proposed service abstraction would encourage delay tolerant network use, how much more traffic would the network be able to accommodate through this abstraction?

Towards this end our contributions are as follows: We present the Mercado architecture and show how it can enable a delay-tolerant service abstraction. We present the well known discounted-utility model [1] as an appropriate framework in which to reason about users' willingness to delay network use in exchange for financial incentives. We present the results of a user survey about delay tolerant network use and show that these results correspond with the economic model. We apply the results from the discount-utility survey to measurement data from a mobile provider to determine the potential of a delay-tolerant service abstraction to reduce peak

CSWS'12, December 10, 2012, Nice, France.

traffic loads. We use the same data to perform a "what-if" analysis on the impact of increased delay-tolerant traffic use.

2. BACKGROUND

To put our work in context, in this section we provide a very brief overview of a modern mobile network architecture. Specifically, Figure 1 depicts the basic architectural components of a Long Term Evolution (LTE) radio access network, combined with an Evolved Packet Core (EPC) network, the combination of which is often referred to as a 4G mobile network [2]. [1] In this architecture all traffic, including voice, is packetized as it traverses the network.

Basic Functionality: With reference to Figure 1, the eNodeB provides the radio interface towards mobile devices and performs a variety of radio access network functions including resource management. Towards the EPC side, the eNodeB and all the EPC architectural components operate as an overlay infrastructure on top of a packetized substrate. Across the packet substrate the EPC components make liberal use of tunneling. For example, a user data session, (or default bearer in EPC parlance), would start at the UE and terminate at the P-GW (or Packet Data Network (PDN) Gateway), across the radio interface this bearer would map onto a radio bearer managed by the eNodeB, while across the EPC the bearer would map onto a succession of two tunnels, one from the eNodeB to the S-GW and another from the S-GW to the P-GW. The mobility management entity (MME) is a control-plane only component that is involved in access control and for determining where bearer related tunnels should terminate. I.e., when a mobile device is first switched on, it will attempt to authenticate with the MME (via the eNodeB). The MME would interact with the home subscriber server (HSS) which maintains subscriber related information to determine whether the device should be allowed access. Assuming the device is allowed onto the network, the MME would then use subscriber information to determine which P-GW the user's bearer should terminate on, and initiate signaling between the eNodeB, S-GW and P-GW to establish the bearer.

Policy and Charging Control (PCC): The remaining components depicted in the EPC part of Figure 1, relates to the policy and charging control functionality of mobile networks. These components collectively deal with three functions namely: (i) Gating, i.e., whether a session request can be allowed or not, (ii) Quality of Service (QoS), i.e., what QoS metrics should be applied to a session, and (iii) Pricing, i.e., how much and who should be charged for an allowed session. With reference to Figure 1, the policy and charging rule function (PCRF), is the control plane component that realizes this functionality. I.e., the PCRF might have rules that determine whether a session is allowed and what QoS treatment it might receive. The policy and charging enforcement function (PCEF) constitutes the data-plane counterpart to the PCRF. Finally, the PCRF and PCEF might both interact with an online charging system (OCS) to ensure correct billing for services provided.

IMS based VoIP: To illustrate the use of this architecture described above, consider a voice-over-IP (VoIP) service which might be realized by an IP multimedia subsystem (IMS), as depicted in Figure 1. Once VoIP specific signaling between a VoIP application on mobile devices and the IMS cloud has completed, the IMS system requests voice specific QoS treatment from the mobile network. I.e., IMS would interact with the PCRF to provide the appropriate information for the voice session. The PCRF would be configured with a set of rules to translate that into a request sent to the PCEF to establish a new (dedicated) bearer with voice-QoS.

[1]The gray "MPAS" component represents our proposed extension to the architecture and will be discussed in the next section.

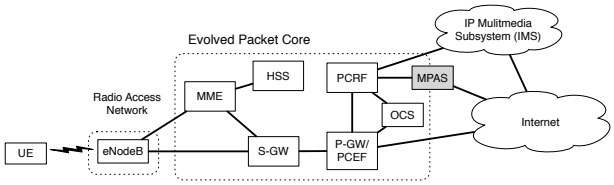

Figure 1: LTE/EPC Architecture with MPAS

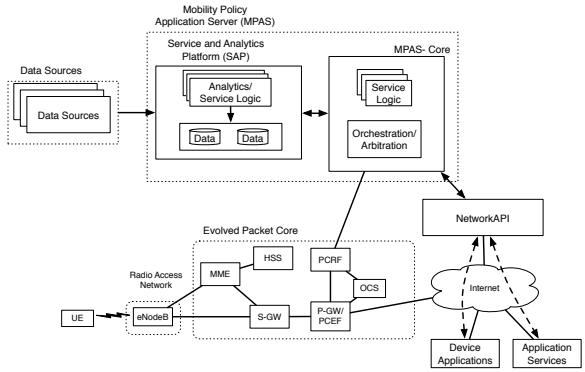

Figure 2: Mercado Architecture

3. MERCADO ARCHITECTURE

New mobile applications and services are, for the most, created without any network support or even awareness that they operate across a mobile network. While some might argue that this network agnostic approach is desirable, we reason: (i) That mobile networking has fundamental constraints that make it unique from other networking environments. (ii) That mobile networking technology (telco-centric as it may be) deal with these fundamental challenges in a highly sophisticated and capable manner. (iii) That the problem with modern mobile networks in not so much with the underlying technology, but with the *service abstraction* that is exposed to applications and services. (iv) That applications/services *and* the network might benefit from exposing alternative service abstractions. (v) And, finally, that such new service abstractions might expose and/or make use of information/intelligence derived from the network.

Given the functionality provided by the policy and charging control (PCC) components in the mobile network architecture (gating, QoS and pricing), these components provide the foundation on which to expose mobile functionality through a new service abstraction. Towards this end we propose an extended mobile architecture as depicted in Figures 1 and 2. Specifically, as shown in Figure 1, we introduce the *mobility policy application server (MPAS)* as a new network component that interfaces with PCC functionality via the PCRF. MPAS exports service abstraction(s) towards applications and services via an application developer environment reachable via the Internet.

In addition to exposing service abstractions related to the three PCC functions, we propose to enrich service abstractions through the use of intelligence derived from the network. The architectural impact of this is depicted in Figure 2. Specifically, we envision a variety of data sources feeding into the MPAS architecture, where a service and analytics platform (SAP) would process this data to make it available as part of the service abstractions offered to applications and services. As shown in Figure 2, the service and analytics platform might maintain its own data stores, e.g., to allow trend-

ing. The figure also depicts the fact that the service abstractions exposed via MPAS might involve different service-logic functions and that MPAS would therefore be required to perform orchestration and arbitration between different service abstractions.

We now describe how the architecture presented could be used to realize a new service abstraction instance. The abstraction uses pricing incentives to encourage delay tolerant applications to delay use of the network in exchange for receiving network services at reduced rates.

3.1 Delay tolerant service abstraction

Figure 3 depicts the somewhat simplified interactions involved in using the architecture presented above to realize a delay tolerant service abstraction. In this case, the data sources used by the service and analytics component of MPAS (see Figure 2) include network load information, which is used as part of the deciding whether to accept a delay tolerant request and to determine when to grant an accepted request. Specifically, as shown in Figure 3, we are assuming the existence of an application on the UE that is aware of the availability of the delay tolerant service abstraction and has delay tolerant data to send or receive. The UE issues a request (1 in the figure) via the NetworkAPI/MPAS combination, which includes the volume of data to be sent/received and a deadline indicating how long a delay it is willing to tolerate to have this request be satisfied. MPAS uses network information, derived from its data sources, to decide whether this request could be accepted or not. The network information used includes current load information (e.g., for requests with short deadlines) and historic load information (e.g., for request with long deadlines).

In Figure 3 we are assuming that MPAS decides to honor the request so that an accept is communicated back to the UE (2). Depending on the deadline, MPAS might at this point issue a request to the PCRF to track any location changes of the UE (3). MPAS now uses the network load information (current and historic), as well as the set of delay tolerant requests to determine when to schedule transmission for a particular UE. When MPAS makes a schedule decision, it issues a grant notification to the UE (4) with an indication of how long the grant is valid. At the same time

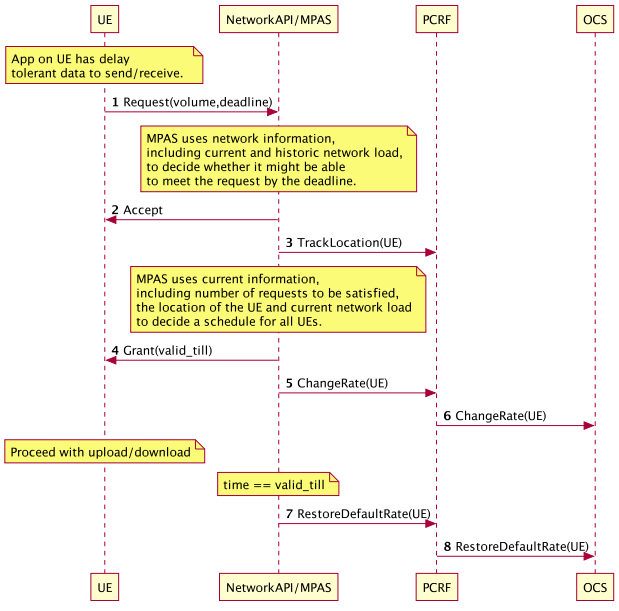

Figure 3: Delay tolerant service abstraction

MPAS issues a temporary rate change notification to the OCS so that the UE would be charged according to the lower delay tolerant rate while the grant is valid. The application on the UE can now proceed with its upload/download which will be billed at the delay tolerant rate. Once the "valid till" time or the previously specified volume limit is reached, MPAS interacts with the OCS to restore the default rate plan for the UE.

4. INCENTIVES

In this section, we address two key assumptions underpinning our proposed delay-tolerant service abstraction: (i) users would be willing to tolerate delay (or conversely make scheduled future use of the network) in exchange for paying a lower price, and (ii) the service abstraction can provide benefit to wireless network service providers.

4.1 Benefit for End Users

In this subsection, we focus on end users' trade-off between delayed network use and discount and present mathematical and empirical results.

Analysis using Discounted Utility: Discounted utility [1] is a model used to capture the trade-off between cost and benefit when a person makes decisions over time. The key concept is that a person *discounts* future utilities, while the discount factor might vary among individuals. While there exists a large body of literature on discount factor, for simplicity, we assume a constant discount factor $0 \leq r < 1$ for a person. In the discounted utility model, an individual tries to maximize the sum of utilities over time: $DU(r) = \sum_{t=0}^{\infty} r^t u_t(x)$ where t is elapsed time units, and $u_t(x)$ is the utility of decision x at time t.

Let us assume that delayed download is only $0 < \alpha < 1$ of the immediate download price. We further assume that each person's discount factor is a random variable following a distribution function $F(r)$. Although we do not show the details, we can derive an optimal waiting time for each user and show the following:

$$P(t) = Pr[\textit{user is willing to wait till t}] = 1 - F(\alpha^{1/t})$$

From this equation, we observe the following: (1) The longer the delay is, the fewer users are willing to wait; (2) The higher the price is (or the smaller the discount is), the fewer users are willing to wait. We next present the result of a user survey to confirm this observation.

Empirical Study using Survey: As discussed in the previous subsection, there are a number of aspects that a user may consider before adopting delayed download: discount rate, user's own preference, and possibly application types. To understand these aspects better, we have built a simple web page to perform a survey. Specifically, we chose 5 application types: Cloud-sync, App Download, Short Video, Long Video, and Email (see Section 5). We also consider three different discount rates: 25% off, 75% off, and free. In the survey page, there are a total of 15 questions, one for each application type and discount rate pair. In each question, a user is asked to select "the amount of time you are willing to wait" among the following 7 values: no delay accepted, 1 min, 10 min, 30 min, 1 hr, 6 hr, and 12 hr. Our survey was taken by a group of 40 volunteers from industry and academia during February and March 2012.

Survey Results: We calculate complimentary cumulative distribution function (CCDF) from the survey results. In Figure 4, we show the CCDF plot for App Download. We observe that the more respondents are willing to wait with higher discount rate, which is consistent with the analytical findings in Section 4.1. Specifically, if the delayed download is free, 82% of the respondents are willing to wait for at least 1 minute (while the other 18% of them are not

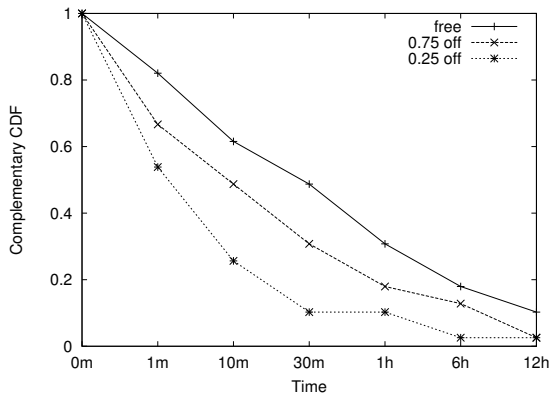

Figure 4: Survey results for App Download: Fraction of respondents willing to wait till different deadlines.

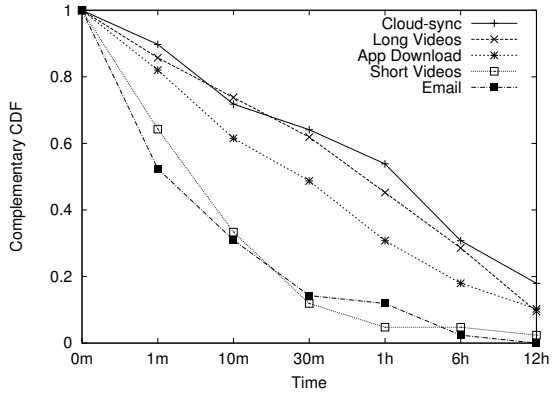

Figure 5: Survey results for all application types: Fraction of respondents willing to wait till different deadlines, when the delayed download is free.

willing to wait at all). In comparison, only 69% are willing to wait for 1 minute or longer if the delayed download is 25% off.

In Figure 5, we show CCDF curves for all the application types when the delayed download is free. We observe two groupings with App Download in the middle. One is Short Video and Email, which is on the lower left of App Download, indicating less delay-tolerant application types. The other is Cloud-sync and Long Video on the upper right, indicating more delay-tolerant application types. We use the relative proportions from this survey in our evaluation study in Section 6.

4.2 Benefit for Service Providers

In this section, we focus on the benefit from flexibly scheduling delay-tolerant traffic, by which a network provider can reduce the peak usage and thus achieve cost saving by avoiding network capacity expansion.

Idealized Smoothing Strategy: We consider an idealized smoothing strategy that minimizes the peak traffic volume based on the perfect knowledge of entire demand during the time period of interest (called T). While we cannot use this strategy in practice, this provides a guideline for maximum achievable benefit and also serves as a lower bound when we evaluate other heuristic strategies.

We consider a linear program using the following inputs and variables. v_t^d denotes the traffic volume of requests arriving at time t with the delay deadline d. Note that $\sum_d v_t^d$ means the total traffic volume arriving at time t. Let us define variable $x_t^d(s)$, which denotes the fraction of v_t^d served at time s. The objective is to find a solution of $x_t^d(s)$ that minimizes the peak traffic during the time window T.

$$\min \max_{s \in T} \sum_{t \in T, d} v_t^d \, x_t^d(s) \tag{1}$$

$$\text{s.t.} \sum_{t \le s \le t+d} x_t^d(s) = 1, \, \forall t, d \tag{2}$$

$$0 \le x_t^d(s) \le 1, \, \forall t, d, s \tag{3}$$

Constraint 2 ensures that all requests are served within the deadline. Constraint 3 reflects that $x_t^d(s)$ are fraction variables between 0 and 1.

While our primary objective is to minimize the peak traffic, there are multiple solutions that achieve the optimal peak. In our evaluation, among those solutions, we use a solution that minimizes the total delay weighted by $\frac{v_t^d}{d}$ to achieve lower overall delay and better user experience. In Section 6, we demonstrate that flexibly scheduling delay-tolerant traffic can lead to significant benefit.

5. DATA ANALYSIS

We now apply the discount-utility model survey results to the real measurement data collected from a large US-based wireless provider to generate a realistic dataset for use in our later performance evaluations of Mercado.

Dataset: We collected data from a 3G service infrastructure. The data was collected from February 28 to March 5, 2012 covering a cluster of SGSNs in a large US West Coast market with approximately 2.8 million subscribers. To perform our analysis, we utilized anonymized flow records containing 1-minute aggregate data. Each anonymized subscriber record contains the traffic volume per application and content provider.

The flow records were then categorized into the following categories that potentially are delay tolerant traffic used in our survey: (i) **Clould Sync:** Syncroniziation traffic with Apple iCloud, Amazon cloud service, etc. (ii) **App Downloads:** Apple, Android, Microsoft App store downloads and updates. (iii) **Short Video:** Content provider delivering short video clips (e.g., YouTube). (iv) **Long Duration Video:** Movie and TV downloads (e.g., Hulu and Netflix). (v) **Email:** all email protocols (e.g., POP3, IMAP, SMTP). (vi) **Other:** everything else (no delay category).

After the flow records are categorized, we then processed to produce session records that have a start time, duration, traffic category and number of bytes transferred. Each session was then assigned a delay value randomly between 0 minutes to 12 hours in proportion to the empirical distribution of accepted delay for the application category using our survey results in Section 4.1.

Data Overview: An overview of the sessions data by application category is presented in Table 1. The short duration video category has the largest amount of traffic among the 5 categories we use. However, as seen in the survey results in Table 1 short duration video has also the least tolerance for delay. After applying the survey proportional distributions (shown in Figure 5) to this delay tolerant traffic, Table 2 shows the percentage of traffic overall in each delay category. We observe that based on the current traffic distributions the amount of long delay tolerent traffic (6+ hours) is low but given economic incentives this portion of traffic has most potential to grow. Note that all flows in "Other" category are aggregated into one entry for each time bin.

Table 1: Data Sessions Overview

Category	% Bytes	Total Bytes (GB)	Num Sessions (K)
App Download	4.7%	391	310
Cloud	0.6%	48	660
Email	5.1%	417	1530
Long Video	4.5%	369	31
Short Video	14.4%	1182	270
Other	70.8%	5824	8

Table 2: Delay Distribution

Delay Category	0-delay	1m	10m	30m	1h	6h	12h
Percent Traffic (%)	79.8	7.2	5.1	2.8	2.1	1.7	1.3

6. EVALUATION

In this section, we quantify the benefit from peak smoothing in the current network condition and also in a setting with increased delay-tolerant traffic volume. We compare the following two schemes in our evaluation: (i) **No-delay:** We schedule all the requests immediately without delaying. This corresponds to the current service offering. (ii) **Idealized:** We use the formulations in Section 4.2 that minimize the peak traffic (primary objective) and relative delay (secondary objective). In some of the plots, we also show the amount of *non-movable* traffic (i.e., 0-delay) to illustrate how much traffic is subject to delaying. Here we assume that all delay tolerant traffic requests are accepted, and determine the peak traffic volume and overall delay metrics.

No-delay vs. Idealized Scheme: We first quantify the amount of reduction in peak traffic volume when we can delay certain traffic requested by users. In Figure 6, we show the traffic volume for different scenarios when we use the current traffic volume and delay tolerant traffic mix as shown in Tables 2. We normalize the traffic volume by dividing it by the peak volume of no-delay scenario. With reference to Figure 6, we can schedule almost all the requests immediately without affecting the peak volume until 16:30, and hence the curve is almost the same as the no-delay case. After 16:30, the idealized scheme starts to delay traffic with longer deadline (e.g., 12h deadline) to later time to avoid increasing the peak volume, which is then scheduled in a few hours (around 20:00). We observe that the peak of the idealized scheme is around 12% lower than the peak of the no-delay case. This modest reduction is because the majority of traffic (79.8% in Table 2) is non-movable, as is illustrated in the figure. Among the movable traffic, around 35% of traffic has 1-minute deadline, which provides less flexibility in scheduling the requests.

Experiments with Increased Delay-tolerant Traffic: We next present results where we increase the amount of traffic for certain application (e.g., assuming increased popularity of applications such as *Cloud-sync* or due to enabling technologies such as the Mercado architecture.) Among the application types we used in our survey, we select *Cloud-sync* and *Long Videos* types and increase their traffic by a constant factor of 5, 10, 20, and 50. We still use the same proportion for different deadlines obtained from our survey. In Figure 7, we present the results when we increase the traffic for *Cloud-sync* and *Long Videos* by a factor of 50. We observe that the peak of the no-delay case increase by a factor of more than 3.5. However, the peak of the idealized scheme increases by a factor of 2.2.

In Figure 8, we show how peak values change when we vary the traffic increase factor for *Long Videos* and *Cloud-sync*. We also consider the per-minute average traffic to illustrate how the increase in overall traffic volume compares with the peak values. As expected, the peak of the idealized case increases as we increase

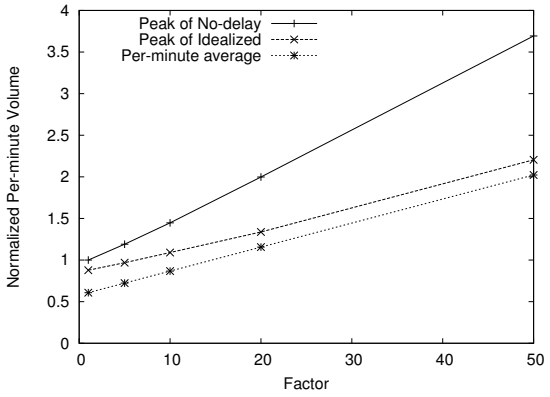

Figure 8: Peak traffic volumes and average per-minute volume when we increase the volume of *Long Videos* and *Cloud-sync* by a factor ranging from 1 to 50.

the traffic. However, we observe that it grows at about the same (or slightly slower) rate as the average traffic volume. In contrast, the peak of the no-delay case increases significantly faster. This result indicates that future traffic growth may require *disproportionate* network capacity expansion to meet the demand, while the Mercado architecture provides an enabling mechanism for reducing the required bandwidth increase.

Delay: To determine the increased delay for delay tolerant traffic due to our approach, we calculated the delay of each request weighted by the byte count. We performed this analysis for both the current workload and the workload where we increased the delay tolerant traffic. For the current workload, we observed that while the majority of delay-tolerant requests are scheduled immediately, once a request is scheduled later (e.g., arriving during the peak time), it can be delayed for a considerable amount time (e.g., up to 207 minutes for some request with 12-hour deadline). As expected, in the case where we increased the traffic by a factor 50, we observed that more delay-tolerant requests experience delays as we admit more traffic without increasing the peak volume.

7. RELATED WORK

Given that the focus of our work is on exposing a delay-tolerant service abstraction, our work is related to the general area of delay tolerant networking [3]. A key difference, however, is that with delay tolerant networking, network conditions *mandate* delay tolerance from applications. In our work we recognize that applications might be inherently delay-tolerant, and then exploit that through a new service abstraction.

More closely related to our work are efforts that attempt to regulate application network use from the mobile device [4, 5]. Both these works exploit knowledge of the low level radio scheduling to influence application interaction with the network. In [4] an analysis tool is presented that allows application developers to better understand the interaction of their application with the network. Their main focus appears to be to reduce energy consumption caused by inappropriate network usage by applications. The work presented in [5], uses a similar awareness of the low level network mechanisms to expose an operating system API to enable delay-tolerant applications to opportunistically transmit traffic using "leftover resources" across the wireless interface. This latter work has much the same motivation as our own work, i.e., recognizing that appli-

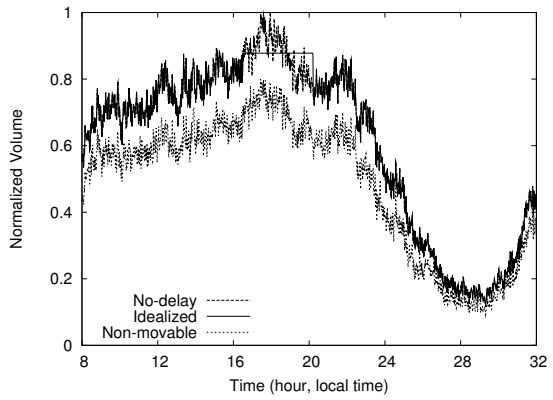

Figure 6: Peak volumes of no-delay and idealized cases using the current data volume and traffic mix.

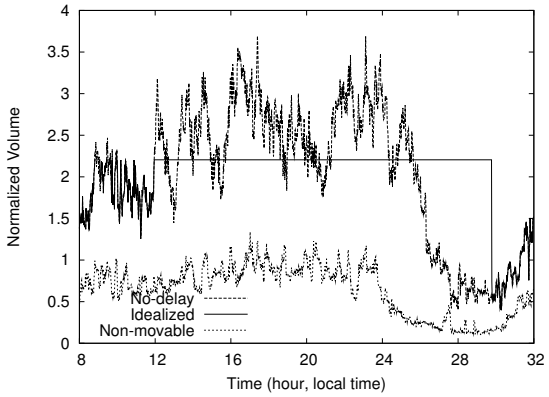

Figure 7: Peak volumes of no-delay and idealized cases when we increase the traffic of *Long Videos* and *Cloud-sync* by a factor of 50.

cations are delay-tolerant, but attempt to exploit that from a completely different (and complementary) angle.

Another body of related work relates to the interaction between network resource usage and pricing, for example [6, 7, 8]. The work described in [6] deals with a pricing approach to perform resource allocation on the wireless link. As such this work deals with resources and pricing at a much finer granularity than Mercado. At a much coarser granularity, time-dependent pricing is considered in [7]. This work looks at the feasibility of time dependent pricing using simulations based on analytical modeling. Finally, the work presented in [8] considers a generalized second price action to allow users who are willing to pay more to receive preferential treatment by the network. The approach is also evaluated using analytical models and simulation. Our work differs from these earlier works in that our approach is enabled by a pragmatic extension to the existing mobile network architecture and specifically in the realism afforded by our use of data from an operational cellular network to evaluate our approach.

Finally, to the extent that our work builds upon and expands the standard mobile networking architecture it is related to the relevant standards bodies [9]. While parts of our work might eventually be subject to standardization by these bodies, we expect service abstractions and in particular the data sources and analysis that enrich such abstractions to provide service differentiation and thus not be subject to standardization.

8. CONCLUSIONS

In this paper, we have proposed a generalized architecture called Mercado that can enable rich interaction and information exchange between mobile devices and the network. As a first step, we focused on the scenario where a mobile device with a delay tolerant data request would interact with the network and through the use of economic incentives delay, when possible, use of the network during congestion until a request can be scheduled afterwards.

We are currently working on a production-grade proof-of-concept realizations that implements the core building blocks proposed in the paper (e.g., MPAS and its interfaces to PCRF and simple example SAP). Once this basic infrastructure is in place, we plan to experiment with various use cases including the one we elaborated on in this paper.

9. REFERENCES

[1] P. Samuelson, "A note on measurement of utility," *Review of Economic Studies*, vol. 4, no. 2, pp. 155–161, 1937.

[2] M. Olsson, S. Sultana, S. Rommer, L. Frid, and C. Mulligan, *SAE and the Evolved Packet Core - Driving The Mobile Broadband Revolution*. Amsterdam, Boston Elsevier LTD., 2009.

[3] K. Fall, "A delay-tolerant network architecture for challenged internets," in *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, SIGCOMM '03, 2003.

[4] F. Qian, Z. Wang, A. Gerber, Z. Mao, S. Sen, and O. Spatscheck, "Profiling resource usage for mobile applications: a cross-layer approach," in *Proceedings of the 9th international conference on Mobile systems, applications, and services*, MobiSys '11, 2011.

[5] H. A. Lagar-Cavilla, K. Joshi, A. Varshavsky, J. Bickford, and D. Parra, "Traffic Backfilling: Subsidizing Lunch for Delay-Tolerant Applications in UMTS Networks," in *Third ACM Workshop on Networking, Systems, and Applications on Mobile Handhelds (MobiHeld 2011)*, October 2011.

[6] P. Marbach and R. Berry, "Downlink resource allocation and pricing for wireless networks," in *Proceedings of IEEE INFOCOM*, 2002.

[7] C. Joe-Wong, S. Ha, and M. Chiang, "Time-dependent broadband pricing: Feasibility and benefits," in *Proceedings of the International Conference on Distributed Computing Systems (ICDCS)*, 2011.

[8] Y.-F. Chen, R. Jana, and K. N. Kannan, "Using generalized second price auction for congestion pricing," in *GLOBECOM*, 2011.

[9] "The 3rd Generation Partnership Project (3GPP)." `www.3gpp.org`.