Adaptive Usage of Cellular and WiFi Bandwidth: An Optimal Scheduling Formulation

Ozlem Bilgir Yetim and Margaret Martonosi
Princeton University
{obilgir,mrm}@princeton.edu

ABSTRACT

Worldwide, mobile data connectivity is now widespread, but not yet ubiquitous due to coverage limits and cost concerns. Mobile data offloading to WiFi—where available—could greatly decrease the usage of cellular data networks. In delay-tolerant applications, one could delay network communication in order to exploit free WiFi connections expected to appear soon. However, WiFi connectivity is limited, and even delay-tolerant applications must meet quality-of-service deadlines. To explore such bandwidth scheduling issues, we develop an optimal MILP-based scheduling framework. Our framework schedules multiple application data streams with varying size and delay tolerance, onto networks with varying coverage and bandwidth, in order to minimize cellular data usage. The ability to subdivide data streams into scheduling units is important, because it allows applications to exploit brief windows of WiFi coverage and it allows tradeoffs between solution quality and solver runtime.

Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design—*Wireless communication*; G.1.6 [**Numerical Analysis**]: Optimization—*Linear programming*

Keywords

Cellular, Wireless, Optimal Scheduling Formulation, DTN

1. INTRODUCTION

While worldwide access to internet connectivity has improved, many environments (e.g. rural regions, vehicular networks) still experience intermittent network coverage and severely-constrained bandwidth. Even in well-developed areas with good network coverage, many cellular data users face pay-by-the-byte usage fees or data throttling caps that make it preferable to use cellular data networks wisely [2, 16]. Thus, both to address infrastructure limits, as well as to abide by user cost and coverage preferences, it makes sense to consider optimized network communication in which cellular data traffic is offloaded to other mediums (e.g. WiFi) where possible. If WiFi is only intermittently available, or if its use sees resource competition from other applications/users, then an individual delay-tolerant application may need to decide whether to postpone communication in order to exploit (free) WiFi a bit later, or whether to use cellular connectivity now.

Our work frames the decision regarding when to wait for WiFi versus when to use currently-available cellular connectivity as a Mixed Integer Linear Programming (MILP) scheduling problem. In particular, the goal is to schedule application data streams onto available network channels in order to minimize cost, while abiding by specified constraints. For our experiments, we minimize total cellular data usage (bytecount), although other capping schemes and cost functions could be considered. Through MILP scheduler constraints, we consider multiple data streams, each with different sizes and deadline requirements, and multiple networks, each with different coverage and bandwidth characteristics. A data stream can be a webpage or photo to download, or a video to stream.

While previous research has studied the question of when and how to use WiFi offloading [3, 8, 12], such prior work has largely been heuristic-driven. In contrast, our work develops and experiments with an optimal scheduling framework. Such a formulation has several advantages. First, the framework allows us to efficiently find true minimum cost solutions for a range of assumptions about application data streams, network availability and pricing. Second, the framework can be used to compare heuristic techniques and see how they approach optimum. Third, lessons learned from the framework can be the inspiration for real-world solutions and practical heuristics.

The primary contributions of our work are: First, we *formalize the cellular/WiFi scheduling problem.* Our formulation accounts for each application's delay tolerance, as well as the bandwidth, latency and availability of different communication options. Second, our MILP solver is *efficient.* With our approach, a wide range of delay tolerant network (DTN) and connectivity-choice scenarios can be solved optimally in 10 minutes or less. Third, our strategy of breaking data streams into *scheduling units* affords *better scheduling for short WiFi availability durations.* For example, when WiFi is available 20% of the time with 10 sec average duration, our approach allows a 23% decrease in cellular data usage compared to the unpartitioned case.

2. PROBLEM FORMULATION USING MILP

To explore opportunities afforded by delay tolerance in challenged networks, we formulate the bandwidth allocation problem as an MILP scheduling problem. As Figure 1 and Table 1 show, the scheduler inputs are parameters regarding the application and data characteristics (size, delay tolerance, etc.) and the network characteristics and availability. Its output is a schedule that seeks to minimize cost—in this case a function of cellular data network usage.

2.1 Objective Function and Constraints

Objective Function: Our MILP scheduler seeks to optimize the cost where many different cost models might be

Term	Description	Type
A_s	Availability time of data stream s	Constant
B_n	Beginning time of network n	Constant
BW_n	Bandwidth of network n	Constant
C	Default number of packets per scheduling unit	Constant
D_{header}	Default header size of scheduling units	Constant
$D_{payload}$	Default payload size of scheduling units	Constant
E_s	Deadline to receive/send data stream s	Constant
F_n	Ending time of network n	Constant
H_u	Header size of scheduling unit $u \in U_s$	Constant
L_n	Latency of network n	Constant
M	Cellular network index	Constant
$MTU_{payload}$	Size of maximum payload for IP packets	Constant
MTU_{header}	Size of header for IP packets	Constant
N	Networks	Set
NC	Available network count	Constant
S	Data streams	Set
SC	Total data stream count	Constant
SS_s	Payload size of data stream s	Constant
T_{max}	Time upper-bound	Constant
U_s	Scheduling Units belonging data stream s	Set
UC_s	Scheduling unit count for data stream s	Constant
US_u	Payload size of scheduling unit $u \in U_s$	Constant
$ns_{u,n}$	Scheduling Unit $u \in U_s$ is scheduled to network n	Var *Binary*
$seq_{u,u'}$	Scheduling Unit $u \in U_s$ is scheduled after scheduling unit $u' \in U_k$ ($u \neq u'$, $A_s < E_k$,$A_k < E_s$)	Var *Binary*
τ_u	Schedule time of scheduling unit $u \in U_s$	Var *Real*

Table 1: Formulation variables, constants and sets. Upper-case terms are inputs. Lower-case terms are formulation variables.

relevant: for example, some users pay per byte; other users pay a monthly subscription fee for the first N bytes and are charged only when they exceed that cap. Here, we consider cost as simply the total byte count of cellular data network usage (Eq. 1). This sums header and payload bytes of scheduling units that use the cellular network.

Constraints: The MILP solver minimizes cost subject to the constraints in Eqs. 2-6. The first constraint (Eq. 2) ensures that scheduling units cannot be scheduled until the data is actually ready. Likewise, Eq. 3 is a deadline constraint requiring that a scheduling unit's transmission finishes by the scheduling unit's deadline. If the time between availability and deadline is larger than what is minimally required to send the data, then the stream has some *delay tolerance* we can exploit.

Eq. 4 has several parts, each of which constrain some aspect of network scheduling. The first part says that each scheduling unit is scheduled to one and only one network. While using multiple types of connectivity at once is promising, our current formulation does not yet consider it. Future extensions could easily cover this case, but even our current formulation allows *different* scheduling units within the same application to use different types of networks. The second part of Eq. 4 says that a scheduling unit can only be scheduled to a network after the start time when the network connectivity is ready for use. The third part requires that transmission must be finished before this network becomes unavailable. These allow for WiFi connectivity that is only sporadically usable.

Minimize: Total Cellular Data Usage :

$$\sum_{s \in S, u \in U_s} (US_u + H_u) \times ns_{u,M} \tag{1}$$

Subject to:

Availability :

$$\forall s \in S, u \in U_s : \tau_u \geq A_s \tag{2}$$

Deadline :

$$\forall s \in S, u \in U_s, n \in N :$$

$$\tau_u + (US_u + H_u)/BW_n + L_n \leq T_{max} \times (1 - ns_{u,n}) + E_s \tag{3}$$

Networks :

$$\forall s \in S, u \in U_s : \sum_{n \in N} ns_{u,n} = 1,$$

$$\forall s \in S, u \in U_s, n \in N : ns_{u,n} \times B_n \leq \tau_u,$$

$$\forall s \in S, u \in U_s, n \in N :$$

$$\tau_u + (US_u + H_u)/BW_n + L_n \leq T_{max} \times (1 - ns_{u,n}) + F_n \tag{4}$$

Sequencing :

$$\forall s \in S, u \in U_s, u' \in U_s, u' \leq u : seq_{u,u'} = 1$$

$$\forall s \in S, k \in S, u \in U_s, u' \in U_k, u \neq u', A_s < E_k, A_k < E_s :$$

$$seq_{u,u'} = 1 - seq_{u',u} \tag{5}$$

Bandwidth :

$$\forall s \in S, k \in S, u \in U_s, u' \in U_k, n \in N, u \neq u', A_s < E_k, A_k < E_s :$$

$$\tau_{u'} + (US_{u'} + H_{u'})/BW_n$$

$$\leq T_{max} \times (3 - ns_{u,n} - ns_{u',n} - seq_{u,u'}) + \tau_u \tag{6}$$

Figure 1: MILP Formulation: Objective function and constraints.

Eq. 5 ensures the correct ordering of the scheduling units which are scheduled to the same network channel. The first part of it sequences scheduling units belonging to the same data stream. The second part requires that sequence of the scheduling units, $\{u,\ u'\}$ be the inverse of the sequence of $\{u',\ u\}$ since both cannot be the same binary value at the same time. Finally as shown in Eq. 6, this constraint ensures that all bandwidth is dedicated to a scheduling unit when it is using a given network. Disallowing bandwidth sharing between applications is not fully realistic, but it improves solver runtime, and is a reasonable first step.

2.2 Data and Network Characteristics

For each data stream, we define an availability time and a deadline. The availability time is when the data becomes ready for uploading or downloading. The deadline is when transmission should be completed. At a minimum, deadlines are always defined to allow feasible scheduling; that is, they are at least as big as the time required to transmit the bytes over the slowest possible network. Beyond this, application delay tolerance may allow the input deadline to be even later. If such slack is allowed, the MILP solver can exploit it to reduce cost by waiting for free WiFi to become available.

Our scheduler assumes that the data streams are divisible to smaller scheduling units. This allows them to make use of smaller windows of WiFi availability. Each scheduling unit consists of a fixed number of packets of size MTU. As a result, the size of each scheduling unit is fixed, but the final scheduling unit in a stream will be smaller if the data stream size is not an even multiple of scheduling unit size. We assume that header information is appended to each packet and our scheduler models this effect. In addition, in some cases, header information might also be appended to each scheduling unit. Since this header size (Bytes) is, however, negligible compared to the scheduling unit sizes (KBs), we assumed that scheduling unit header sizes are 0.

Our network model is simple, but includes key characteristics for evaluating the potential benefit of delay tolerance and data scheduling. Our results here assume that the cellular network is available all the time, but our MILP formulation and constraints can already handle cases when, as in many challenged networks, this is not the case. We further assume that cellular and WiFi networks can be used simultaneously by different applications. This is important since prior work has already demonstrated that concurrent network connections can improve performance, battery life and throughput [13, 11]. Our formulation does not allow a single application's scheduling unit to be split between cellular and WiFi, but such scenarios could be evaluated by our solver simply by inputting different parts of the application data as separate streams. When a data scheduling unit is using a particular network connection, we assume all the bandwidth is dedicated to that data stream. This makes the scheduling problem easier but could be broadened in future work.

3. EXPERIMENTAL METHODOLOGY

Our studies focus on up to four application data streams. Data stream delay tolerances can be very high, particularly in rural or challenged networks [7]. Our experiments conservatively vary them from 0 to 256 seconds, but prior DTNs have considered delay tolerances of an hour or more [8]. Data stream sizes are generated randomly from an exponential distribution with mean size of 2MB. (Single web page sizes averaged almost 700 KB in 2011 [14]. Larger streams would come from audio or video feeds.) Since longer streams can be considered as repetitions of schedules optimal for shorter subsets, variability with data size is not high. We verified this with other experiments covering a larger range of data sizes, but space constraints preclude a detailed presentation. We use a uniform random distribution for when data streams appear in the timeline.

Bandwidths of different cellular networks vary heavily depending on service provider and networking technology. We used equal download/upload speeds, and for most of our results, we set cellular network bandwidth to 1 Mbps [9, 10]. For WiFi, we vary from 1 Mbps to 64 Mbps [15]. The baseline cellular network latency is 200 msec [9] and baseline WiFi latency is 25 msec [5]. MTU size is set to be 1500B (1480B payload + 20B header) [9]. WiFi's availability varies with speed and location. Our experiments consider WiFi availability from 0 to 100%. The duration of WiFi availability is set to average at 10sec [4]. We assume uniform random distributions for network begin/end times.

Our MILP scheduling formulation is written in AMPL and solved using IBM CPLEX 9.1 optimizer [1, 6]. The solver runs on a system with 2.33 GHZ Intel Xeon Quad-Core CPU, 16GB memory and Red Hat Linux 4.1. Since some input parameters are randomly chosen from stochastic distributions, we repeat each experiment 30 times. Section 4 shows the average objective across these runs. We limit optimizer runtime to 10 minutes per run. Within this time limit, 96.4% of the experimental points (954 out of 990) gave an exact/optimal solution. For the remaining 46 schedules, CPLEX reports an average optimality gap of 48.5%; this is the relative difference between the best integer solution found and the lower bound of the best of the remaining nodes [6].

4. RESULTS

To give the reader insights about the scheduler, Figure 2 shows a small example. The following paragraphs discuss experiments using the solver.

MILP Performance and Scheduling Units: This section considers scheduling units, both in terms of their impact on network usage as well as in terms of their impact on MILP solver performance. Scheduler units are an intermediate granularity between network level packets (of size MTU) and full data streams. Scheduling at the granularity of full data streams would be unrealistic, since large chunks of data could not feasibly be scheduled onto short WiFi connectivity times. On the other hand, optimal scheduling of realistic data sizes at the granularity of individual packets would lead to prohibitively-long solver run times. To explore these tradeoffs, we vary scheduler unit size from 128 MTU-sized packets to "infinite" (i.e. unpartitioned data).

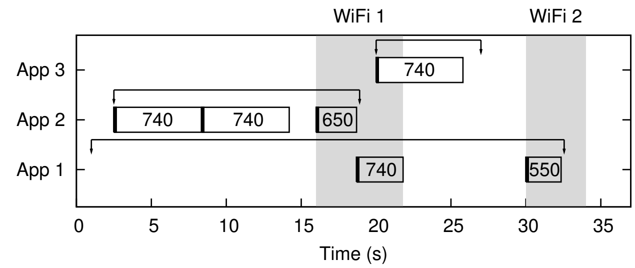

Figure 2: This Gantt chart shows an example 3-application schedule obtained from the solver. While it assumes cellular connectivity is always available, WiFi is available only at the gray-shaded times with 2X cellular bandwidth. The arrowed-windows show when data streams become available and their deadlines. Each rectangle is one scheduling unit, and the number denotes its size in KBs. Black rectangles are the headers. Transmissions using cellular network are shown in white, and those using WiFi are shown in grey.

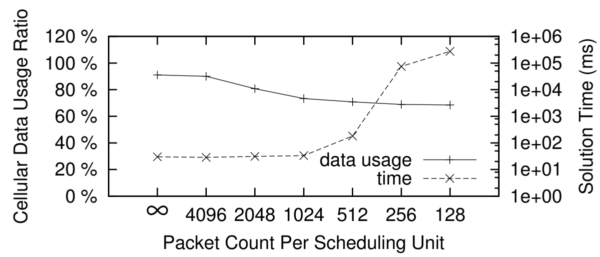

Figure 3: As packet count per scheduling unit decreases, scheduling unit sizes decrease. This causes the cellular data usage to decrease by 23% compared to the ∞ case where data streams are not partitioned to smaller scheduling units. On the other hand, small scheduling units increase scheduling complexity, causing solution times to increase. 512 is a good tradeoff point.

Figure 3 shows the effect of scheduling unit size on cellular data usage and solution time. These results are for an average data stream size of 2MB and WiFi ratio of 20%. "∞" shows the result for the no-partitioning case in which a single scheduling unit comprises an entire data stream. Compared to a case where no WiFi is ever available, even the unpartitioned case offers a 9% decrease in cellular data usage. Because this case has the fewest scheduling options to consider, its average solution time is very short: 30 ms. Considering smaller scheduling units (fewer packets per unit) offers more cost savings. This is because small WiFi durations are utilized better with smaller scheduling units. Cellular data usage reaches 68% at 128 packet count. While solution time increases as the scheduling unit size decreases, it remains manageable. At 128 packets per scheduling unit, average solution time is 274 s.

Overall, controlling scheduling unit sizes allows us to find tradeoffs with fast-enough solution times and realistic network usage. Going from 128 packets to 512 packets per scheduling unit, we can decrease the average solver time by more than 1500X with only a 2% cost increase. Therefore,

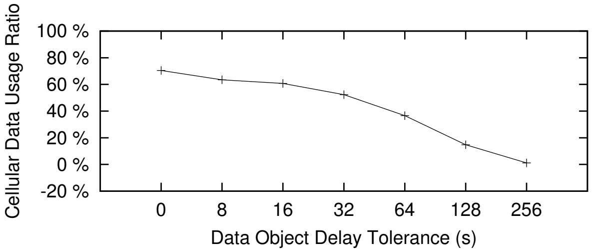

Figure 4: Increasing application delay tolerance decreases cost (cellular data usage). For 256 s delay tolerance, cost approaches 0, since data streams can almost always wait for WiFi.

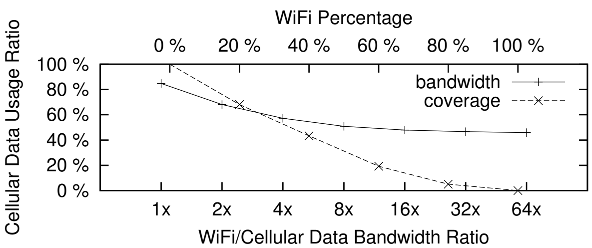

Figure 5: Changes in WiFi ratio have greater effect on cellular network usage when WiFi coverage is low. As WiFi bandwidth increases, cellular data usage first decreases sharply. After a certain point, additional WiFi bandwidth without additional coverage time is of no further use because data stream sizes and arrival and deadline times will eventually limit the maximum amount that WiFi can be exploited.

for the rest of the paper, scheduling units are 512 MTU-sized packets. With this unit size, 96.4% of the simulations give an exact solution under 10 minutes.

Effect of Delay Tolerance: A key goal of our research is to explore how application delay tolerance can be exploited to save on the cost of cellular network usage. In particular, Figure 4 shows how cellular network data usage varies with application delay tolerance varying from 0 to 256 s. A delay tolerance of 0 s means that the application's data requirements fit exactly into the time required to transmit them over the cellular network; if a higher-bandwidth WiFi network is available, some slack may be present even while meeting the same deadline. As we increase the delay tolerance, the application's data stream deadline is shifted further. (To prevent deadlines larger than the simulation period, the simulation period is 300 s in this section.)

Increasing delay tolerance increases the chances for a packet to make use of a WiFi channel; this in turn substantially decreases the cellular data usage. The graph shows the most rapid decrease until 128 seconds, making the cellular data usage ratio as low as 13%. After that, the probability that a packet will have the chance to use WiFi is high enough and for 256 the ratio decreases almost to 0. This curve shows us that for the WiFi/cellular network characteristics we used for experimentation, delay tolerance is one of the most crucial parameters. APIs and system support for applications to express their delay tolerance can be one of the highest-leverage ways of improving cellular vs. WiFi adaptation.

Effect of WiFi Ratio and Bandwidth: Figure 5 shows how different fractions of WiFi coverage and bandwidth influence the amount of cellular data network used. From 0% to 100% coverage, WiFi connections vary from never available to always available. For this experiment, all data streams have a delay tolerance of 0 and WiFi bandwidth is 2X cellular. As expected, increasing WiFi coverage significantly decreases cost as measured in cellular network usage. The cost decrease is much sharper for low coverage; with 60% coverage, the cellular usage has already dropped to only 20% of the communicated data. Since 100% WiFi coverage may not be an option especially for developing regions, our scheduler can usefully plan when to use it.

Similarly, varying WiFi bandwidth from 1 Mbps (equal to cellular network bandwidth) up to 64 Mbps (64X cellular network bandwidth), cellular network usage first decreases sharply, and then reaches a steady state of roughly 45% usage ratio. The size, arrival times and deadline times of data streams eventually place a limit on where the WiFi bandwidth can be exploited. Beyond this, further increases in bandwidth offer no benefit. Overall, increasing WiFi bandwidth has diminishing returns while increasing delay tolerance is always advantageous.

5. CONCLUSION

This work proposed and experimented with an optimal cellular/WiFi network scheduling framework. Our framework finds a minimum cost schedule for multiple-application data streams on different networks. Exploiting modest delay tolerance has high leverage in reducing per-byte cellular costs. Furthermore, breaking data streams into smaller scheduling units offers faster solver times and better scheduling for short WiFi durations. We also showed how cellular data usage varies under varying WiFi coverage, bandwidth and duration scenarios. Our results offer context for dynamic and heuristic-driven proposals in this topic of considerable and increasing real-world importance.

6. REFERENCES

[1] AMPL: A modeling language for mathematical programming. `http://www.ampl.com/`.

[2] Data plans from AT&T. `http://www.att.com/shop/wireless/plans/data-plans.jsp?WT.srch=1&fbid=qJecLdOu-5Q`, 2012.

[3] A. Balasubramanian et al. Augmenting mobile 3G using WiFi. MobiSys, 2010.

[4] V. Bychkovsky et al. A measurement study of vehicular internet access using in situ Wi-Fi networks. MobiCom, 2006.

[5] E. Cuervo et al. MAUI: making smartphones last longer with code offload. MobiSys, 2010.

[6] IBM ILOG ODM enterprise developer edition. `http://pic.dhe.ibm.com/infocenter/odmeinfo/v3r4/index.jsp`, 2010.

[7] S. Isaacman and M. Martonosi. The C-LINK system for collaborative web usage: A real-world deployment in rural Nicaragua. NSDR, 2009.

[8] K. Lee et al. Mobile data offloading: How much can WiFi deliver? Co-NEXT, 2010.

[9] R. Ludwig et al. TCP over Second (2.5G) and Third (3G) Generation Wireless Networks, 2003.

[10] B. Nadel. 3G vs. 4G: Real-world speed tests. `http://www.computerworld.com/s/article/9201098/3G_vs._4G_Real_world_speed_tests`, 2010.

[11] E. Nordstrom et al. Serval: An end-host stack for service-centric networking. In *NSDI*, 2012.

[12] A. Rahmati and L. Zhong. Context-for-wireless: context-sensitive energy-efficient wireless data transfer. MobiSys, 2007.

[13] H. Soroush et al. Spider: improving mobile networking with concurrent wi-fi connections. SIGCOMM, 2011.

[14] Average Web Page Size Septuples Since 2003. `http://www.websiteoptimization.com/speed/tweak/average-web-page/`, 2011.

[15] Wi-Fi / WLAN channels, frequencies and bandwidths. `http://www.radio-electronics.com/info/wireless/wi-fi/80211-channels-number-frequencies-bandwidth.php`.

[16] C. Ziegler. Sprint falls in line, caps "unlimited" data at 5GB. `http://www.engadgetmobile.com/2008/05/19/sprint-falls-in-line-caps-unlimited-data-at-5gb/`, 2008.