

OSCAR: A Deployable Adaptive Mobile Bandwidth Sharing and Aggregation System

Karim Habak*, Khaled A. Harras†, Moustafa Youssef‡

*Georgia Tech, †CMU-Qatar, ‡E-JUST

*karim.habak@cc.gatech.edu, †kharras@cs.cmu.edu, ‡moustafa.youssef@ejust.edu.eg

ABSTRACT

The exponential increase in mobile data demand coupled with the rapid deployment of various wireless access technologies have led to the proliferation of multi-interface enabled devices. As a result, researchers focused on exploiting the available interfaces on such devices in both solitary and collaborative forms. Unfortunately, the proposed systems, that exploit these interfaces, face a formidable deployment barrier. Therefore, in this work, we present OSCAR, a system that exploits multiple network interfaces on modern mobile devices. OSCAR provides a set of mechanisms for sharing bandwidth across multiple collaborating devices and enables this collaboration by providing users with sharing incentives. We present an overview of the system coupled with the OSCAR scheduler. We evaluate OSCAR via implementation on Linux and compare our results to the current optimal achievable throughput. Our preliminary results shows that in the throughput maximization mode, we provide up to 150% enhancement compared to using only one interface, without any changes to legacy servers.

1. INTRODUCTION

The Federal Communications Commission (FCC) has indicated an expected *data tsunami* predicting a 25-50× increase in mobile data traffic by 2015 [1]. This expected explosive demand for mobile data, along with expensive data roaming charges and user expectation to remain connected in all places at all time, are creating novel challenges for service providers and researchers to solve. A potential approach for solving some of these challenges is exploiting all communication interfaces available on modern mobile devices.

Currently, researchers focus on exploiting all the available interfaces on mobile devices in both solitary and collaborative forms [3]. In the solitary form, they aim to exploit Internet connectivity on any of the available interfaces by distributing application traffic across them to achieve higher throughput and/or minimize energy consumption [2, 5, 6]. In the collaborative form, the goal is to have mobile devices utilize their neighbors' potentially under-utilized bandwidth in addition to their own direct Internet connections. These approaches either deal with a small scale collaborative community managed by a single authority [9], or utilize proxy servers to handle and guarantee such collaboration [8]. Overall, solu-

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

MOBIQUITOUS 2014, December 02-05, London, Great Britain

Copyright © 2014 ICST 978-1-63190-039-6

DOI 10.4108/icst.mobiquitous.2014.258056

tions proposed to date face a high deployment barrier, and have focused on bandwidth maximization without paying sufficient attention to energy or cost efficiency.

In this work, we present the OSCAR collaborative bandwidth aggregation system that fulfills the following requirements: (1) Being deployable since it does not require changes to legacy servers, applications, or network infrastructure. (2) Exploiting available network interfaces in solitary and collaborative forms. (3) Being able to adapt to real-time Internet characteristics and the system parameters to achieve efficient utilization of these interfaces. (4) Providing users with incentives for sharing their bandwidth. (5) Leveraging incremental adoption and deployment to further enhance performance gains.

We evaluate OSCAR via implementation on Linux Operating System and show its ability to increase the overall system throughput while achieving its cost and energy efficiency targets. Results show that, with no changes to the current Internet architecture, OSCAR reaches the throughput upper-bound, providing up to 150% enhancement in throughput compared to the current Operating Systems without any change to legacy servers. Our results also demonstrate OSCAR's ability to maintain cost and the energy consumption levels in the user-defined acceptable ranges.

2. SCENARIO AND SYSTEM OVERVIEW

When John is having a meal in a food court, or waiting for the train at the station, he watches youtube videos, listens to podcasts, and uses Facebook to get his social network feeds; using his tablet equipped with WiFi, Bluetooth, and 3G interfaces. John connects to the congested free WiFi hotspot because 3G roaming charges are too expensive. All the data for the above applications go through this heavily congested interface, leading to a very unpleasant experience. Meanwhile, Mark and Alice, who are sitting next to John, are using a flat-rate 3G plan and a private WiFi hotspot subscription, respectively. John's experience can be greatly improved by scheduling different applications' traffic through his available network interfaces using these underutilized connections of Mark and Alice. Figure 1 depicts such a scenario, based on OSCAR, in which John's scheduler directly uses the expensive 3G only for light weight important data, while leveraging underutilized bandwidth from his neighbors using his bluetooth and WiFi interfaces.

Figure 1 shows that we have five communicating entities in our scenario. Firstly, *client devices* equipped with multiple network interfaces varying in their available bandwidth, energy consumption rates, and cost per unit data. Each interface has different *paths* to the Internet, either directly or through neighboring client devices sharing their connectiv-

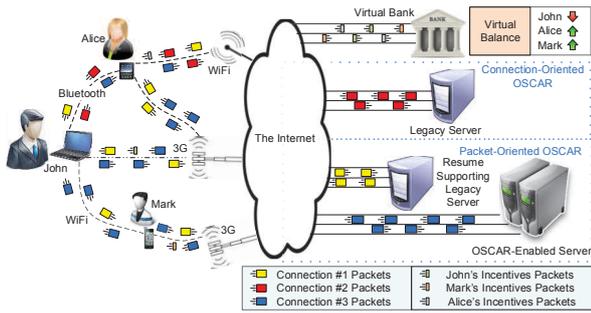


Figure 1: OSCAR scenario.

ity options. Secondly, a *virtual bank* representing a trusted third party that handles payment or sharing reputation incentive mechanisms between client devices. Thirdly, *legacy servers* which are typical non-modified Internet servers. When communicating with these servers, OSCAR traffic schedulers residing on clients use a *connection-oriented* mode to schedule different connections to the available paths where a TCP connection can be assigned to only one path. Fourthly, *connection resume-supporting legacy servers*, such as HTTP servers that support resuming the connections. OSCAR leverages these servers to enhance the performance by switching to a *packet-oriented mode*, where each packet or group of packets can be independently scheduled on a different path. Finally, *OSCAR-enabled servers*, represent servers that may adopt and run OSCAR in the future, to help clients with highly efficient *packet-oriented* scheduling.

3. OSCAR SCHEDULER

3.1 System Model

We assume a mobile device with m different paths to the Internet. Each path represents a way to connect to the Internet either by using the Interface's direct connectivity or by using the interface to communicate with one of its neighbors sharing its Internet connectivity. Each of those paths has its effective bandwidth b_j and cost per unit data c_j , which can be the service provider usage cost or the cost paid to the neighbor to use their connectivity. In addition, each path uses one network interface and it has an energy consumption rate a_j , where a_j equals the difference in power consumption between the active and idle states of the used interface. The data rate of each path interface is denoted as r_j . The device runs a set of connections that share these interfaces and varies in their characteristics.

Our scheduling unit is a connection or a packet. We refer to a standard network connection as a *stream* to avoid confusion with the scheduling unit. Scheduling decisions are taken when a new stream (number of streams active in the system is n , including the new stream) is requested from an application. OSCAR automatically determines whether the operation mode is connection-based ($S_n = 1$), or packet-based ($S_n = 0$) when the other end is OSCAR-enabled or supports the resume mode. In the former case, the scheduler's goal is to determine to which path it should be assigned (sets $x_{nj} = 1$ for only one path j). In either case, the percentage of packets to be assigned to each path, i.e. paths relative packet load (w_j), should be re-calculated based on the current system load (\mathcal{L}). Our OSCAR scheduler has three modes of operation based on user preferences: (1) throughput maximization, (2) energy minimization, and (3) cost minimization. Due to space constraints, we only present a brief overview of the *throughput-maximization* mode and refer the readers to the accompanying technical report [4] for detailed description of the three modes of operation.

3.2 Optimal Scheduling

In this section, we provide a brief overview of our throughput-maximization scheduling mode. The decision variables are: (1) If $S_j = 1$, which path to assign the new stream n to (variable x_{nj}) and (2) the new values for w_j , $\forall j : 1 \leq j \leq m$. The scheduler's goal is to maximize the system throughput under certain energy and cost constraints.

Objective Function: The objective of the scheduler at any decision instance is to maximize the overall system throughput (\mathcal{T}) which is equivalent to minimizing the time needed to finish the whole system load. The objective function can be written as follows:

$$= \text{Minimize } \max_j \left(\frac{\sum_{i=1}^n (\mathcal{L}_i (1 - S_i) w_j) + \sum_{i=1}^n (\mathcal{L}_i S_i x_{ij})}{b_j} \right)$$

Where the left summation represents the packet-oriented mode load (each term, $\mathcal{L}_i (1 - S_i) w_j$, is the number of bytes from the packet-oriented stream i load assigned to path j) and the right summation is the connection-oriented mode load. Note that any stream i will be either connection-oriented ($S_i = 1$) or packet-oriented ($S_i = 0$) and thus will appear in only one of the two summations. Dividing the sum of two loads by the available bandwidth on that path (b_j) gives the time needed for path j to finish its load.

Constraints: The following constraints must be satisfied: (1) To maintain the cost efficiency, the average cost per unit data should not exceed the threshold set by the user; (2) To maintain the energy efficiency, the average energy consumed per unit data should not exceed the threshold set by the user; (3) For a new connection-oriented stream, it should be assigned to only one path; and (4) For the available packet-oriented streams, their total load should be distributed over all paths. we omit the mathematical formulation of these constraints due to space constraints

Solution: In general, this problem is a mixed 0-1 Integer Programming problem, which is an NP-complete problem. However, it has a special structure that allows for an efficient solution. In particular, we have two cases: if the new stream that triggered the scheduling decisions is packet-based ($S_n = 0$) and if it is connection-based ($S_n = 1$). In the former case, $x_{nj} = 0 \forall j$. The problem becomes a standard linear programming problem. Hence, it can be solved efficiently to determine the different values of w_j . In the later case, we need to determine the binary variables $\forall_j x_{nj}$ such that only one of them equals 1 and others are 0. Our algorithm sets each one of them to 1 and then solves the resulting linear programming problem to find $\forall_j w_j$. The value that achieve the best objective is then selected as the optimal decision.

4. PERFORMANCE EVALUATION

In this section we evaluate the performance of OSCAR via implementation on Linux Operating System using the Click modular router[7]. To evaluate the performance of the OSCAR, we use a testbed that consists of six nodes: an OSCAR-enabled server, two legacy servers with only one of them supporting the resume functionality, a main client, a neighboring device sharing its bandwidth, and a traffic shaper, which connects the clients (the client device and the neighboring device) to the servers. Both clients are enabled with multiple network interfaces. On the main client, we run different applications that vary in terms of the number of connections per second they open (β), the average connection data demand (λ). The client is connected to the traffic

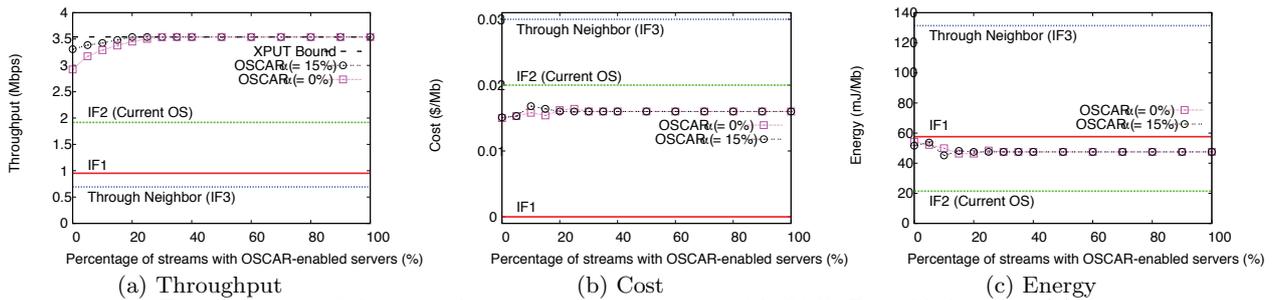


Fig. 2: Impact of changing the percentage of streams with OSCAR-enabled servers (γ).

shaper node through two interfaces: IF_1 and IF_2 . It is connected to the neighboring device via IF_3 at the client and IF_4 at the neighbor. The neighboring device is connected to the traffic shaper through IF_5 . The neighbor shares its available bandwidth with incentive cost of $0.03\$/Mb$. Each server is connected to the traffic shaper using a single high bandwidth link (L). We note that the combined bandwidth of IF_1 , IF_2 and IF_5 is less than each server bandwidth to test the true impact of varying the interface characteristics and scheduling strategies. We define $\gamma \in [0, 100]$ as the percentage of connections that have the OSCAR-enabled servers as their destination. When $\gamma = 0$, all connections are with legacy servers, when $\gamma = 100$ all the connections are with OSCAR-enabled servers. Table 1 summarize the characteristics of the interfaces used in our evaluation.

We evaluate OSCAR using two classes of applications: browsers ($\lambda_{HTTP} = 22.38KB$) and FTP applications ($\lambda_{FTP} = 0.9498MB$). The connection establishment rate follows a Poisson process with mean (β) connections per second ($\beta_{HTTP} = 13\text{ con/sec}$, and $\beta_{FTP} = 1\text{ con/sec}$).

4.1 Results

In this section, we evaluate the performance of OSCAR using three metrics: throughput, energy consumption per unit data, and cost per unit data. We vary both the percentage of connections with OSCAR-enabled servers (γ) and the percentage of resumable connections (α). We compare OSCAR against the theoretical throughput upper bound. Figure 2 shows the effect of increasing the percentage of streams established with OSCAR-enabled servers (γ) on the performance of the OSCAR scheduler for different values of α (the percentage of resumable streams to legacy servers). We set the energy consumption and cost limits to their maximum limits (131.36 Joule/Mb and $0.03\text{ \$/Mb}$ respectively) to highlight the throughput gains that can be achieved by OSCAR. Based on Figure 2(a), we share the following observations: (1) Even when $\gamma = 0$ and $\alpha = 0$ (i.e. only working with legacy servers with **no** resume support), OSCAR can enhance the throughput by 150% as compared to the current OSs. (2) When γ and α are low, most of the streams are connection-oriented, rendering the scheduling decision coarse grained; once the stream is assigned to a path, all its packets have to go through this path until termination. This reduces the scheduling optimality. (3) For $\alpha = 0\%$,

Table 1: Experimental Interfaces Characteristics.

Network Interface	Power (mWatt)	Cost (\$/Mb)	D. Rate (Mbps)	BW (Mbps)
IF_1 (WiFi)	634	0	11	1
IF_2 (3G)	900	0.02	42	2
IF_3 (Blue.)	95	0	0.7232	0
IF_4 (Blue.)	95	0	0.7232	0
IF_5 (WiFi)	726	0	11	1

the system reaches its throughput upper bound when we have only 30% of the streams connecting to OSCAR-enabled server ($\gamma = 30$). (4) This need of OSCAR-enabled servers decreases as α increases, till it reaches 0% when $\alpha = 35\%$.

Figure 2(b) shows that OSCAR significant increase in throughput comes with even better cost for the user. This can be explained by noting that the current OSs use the interface with the maximum throughput, which happens to be the costly 3G interface in our case. OSCAR, on the other hand, mixes the different interfaces, leading to lower cost. In addition, Figure 5(c) shows, as we relaxed the constraints on the energy consumption, that OSCAR uses more energy consumption than the current operating systems to achieve its superior throughput gains.

5. CONCLUSION

We proposed OSCAR, a system that exploits multiple network interfaces on modern mobile devices, and presented an overview of it as well as its scheduler. We evaluated OSCAR using our prototype implementation. Our preliminary results shows that in the throughput maximization mode, we provide up to 150% enhancement compared to using only one interface, without any changes to legacy servers.

6. REFERENCES

- [1] FCC's national broadband plan. <http://www.broadband.gov/>.
- [2] K. Habak, K. Harras, and M. Youssef. OPERETTA: An optimal energy efficient bandwidth aggregation system. In *IEEE SECON*, 2012.
- [3] K. Habak, K. A. Harras, and M. Youssef. Bandwidth Aggregation Techniques in Heterogeneous Multi-homed Devices: A Survey. *ArXiv e-prints 1309.0542*, 2013.
- [4] K. Habak, K. A. Harras, and M. Youssef. OSCAR: A Collaborative Bandwidth Aggregation System. *ArXiv:1401.1258*, Jan. 2014.
- [5] K. Habak, M. Youssef, and K. Harras. DBAS: A Deployable Bandwidth Aggregation System. *IFIP NTMS*, 2012.
- [6] K. Habak, M. Youssef, and K. A. Harras. An optimal deployable bandwidth aggregation system. *Computer Networks*, 2013.
- [7] E. Kohler, R. Morris, B. Chen, J. Jannotti, and M. F. Kaashoek. The click modular router. *ACM Trans. Comput. Syst.*, 2000.
- [8] P. Sharma, S.-J. Lee, J. Brassil, and K. G. Shin. Aggregating bandwidth for multihomed mobile collaborative communities. *IEEE TMC*, 2007.
- [9] D. Zhu, M. Mutka, and Z. Cen. QoS aware wireless bandwidth aggregation (QAWBA) by integrating cellular and ad-hoc networks. In *QSHINE*. Citeseer, 2004.