Computer Networks 57 (2013) 3067–3080

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

An optimal deployable bandwidth aggregation system ☆

Karim Habak [a,*], Moustafa Youssef [a,b], Khaled A. Harras [c]

[a] Wireless Research Center, Egypt-Japan University of Science and Technology (E-JUST), Alexandria, Egypt
[b] Alexandria University, Alexandria, Egypt
[c] Computer Science Department, School of Computer Science, Carnegie Mellon University Qatar, Qatar

ARTICLE INFO

Article history:
Received 12 November 2012
Received in revised form 6 April 2013
Accepted 18 July 2013
Available online 30 July 2013

Keywords:
Bandwidth aggregation
Multiple network interfaces
Throughput
Optimization
Multihoming

ABSTRACT

The explosive increase in data demand coupled with the rapid deployment of various wireless access technologies have led to the increase of number of multi-homed or multi-interface enabled devices. Fully exploiting these interfaces has motivated researchers to propose numerous solutions that aggregate their available bandwidths to increase overall throughput and satisfy the end-user's growing data demand. These solutions, however, do not utilize their interfaces to the maximum without network support, and more importantly, have faced a steep deployment barrier. In this paper, we propose an optimal deployable bandwidth aggregation system (*DBAS*) for multi-interface enabled devices. We present the *DBAS* architecture that does not introduce any intermediate hardware, modify current operating systems, modify socket implementations, nor require changes to current applications or legacy servers. The *DBAS* architecture is designed to automatically estimate the characteristics of applications and dynamically schedule various connections and/or packets to different interfaces. We also formulate our optimal scheduler as a mixed integer programming problem yielding an efficient solution. We evaluate *DBAS* via implementation on the *Windows OS* and further verify our results with simulations on NS2. Our evaluation shows that, with current Internet characteristics, *DBAS* reaches the throughput upper bound with no modifications to legacy servers. It also highlights the significant enhancements in the response time introduced by *DBAS*, which directly enhances the user experience.

1. Introduction

The Federal Communications Commission (FCC) recent notice on the *data tsunami problem* predicts a 25–50× increase in mobile data traffic by the year 2015 [2,3]. Fortunately, the widespread deployment of various wireless technologies coupled with the increase of multi-interface enabled devices are providing users with many alternatives for sending and receiving data. Simultaneously leveraging these interfaces by potentially aggregating their bandwidths can lead to higher throughput, improved end-user experience, efficient resource utilization and, more importantly, a solution for the world-wide *data tsunami problem*.

Unfortunately, current devices and operating systems do not exploit the true potential of these interfaces as they allow only the usage of one interface at a time. Researchers have addressed the multi-interface bandwidth aggregation problem over the years with solutions and techniques implemented at different layers of the protocol stack with a large focus on solutions at the transport and network layers [4–19]. These solutions, however, have faced a steep deployment barrier and do not utilize these interfaces to their maximum without network support.

☆ An earlier version of this paper has appeared in the proceedings of the 5th IFIP International Conference on New Technologies, Mobility and Security (NTMS) 2012 [1].

* Corresponding author. Tel.: +1 (404) 645-4232.
E-mail addresses: karim.habak@ejust.edu.eg (K. Habak), moustafa.youssef@ejust.edu.eg (M. Youssef), kharras@cs.cmu.edu (K.A. Harras).

1389-1286/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.comnet.2013.07.012

In order to have successful bandwidth aggregation systems for multi-interfaced devices, it is inevitable for these systems to be: (1) Easily deployable without requiring changes to legacy servers, applications, or the addition of new hardware such as proxy servers and routers, (2) Optimally exploiting all the available network interfaces to their maximum, and (3) Able to leverage the available applications, interfaces, and traffic characteristics, for more effective scheduling decision making.

We therefore present an *Optimal Deployable Bandwidth Aggregation System (DBAS)* for multi-interface enabled devices. Our system is based on a middleware that lies below the application layer, requiring no changes to either the OS kernel or the applications. One of the core functionalities of this middleware is to schedule different connections and/or packets to different interfaces. We propose *connection-oriented* and *packet-oriented* scheduling techniques that aim to maximize the overall system throughput. We also formulate an optimal scheduling technique that allows the user to achieve the maximum throughput. We show that this is a mixed integer programming problem that can be efficiently solved due to its special structure. Furthermore, *DBAS* leverages the resume functionality common in many protocols [20] to enable packet-level scheduling with current legacy servers to achieve higher performance gains. Finally, as our system is gradually adopted, *DBAS* leverages the existence of any DBAS-enabled server for further performance gains.

We evaluate our system via implementation. To both test the scale of our system and validate our implementation results, we conduct further evaluation via simulations on NS2 [21]. Currently, our *DBAS* prototype is deployed using a standard *MS Windows* installation, highlighting its deployability and ease of use. We use the overall system throughput as well as the different applications response times as our main evaluation metrics. We compare the results with the current operating system's approach as well as other baseline schedulers. Our results show that *DBAS* significantly increases the overall system throughput and minimizes the applications response time, which enhances the user experience. In addition, with no changes to the current Internet architecture, *DBAS* reaches the throughput upper-bound.

The remainder of this paper is organized as follows. We first present a motivating scenario for *DBAS* in Section 2. We present the *DBAS* architecture and components in Section 3. Section 4 discusses our different scheduling techniques and formulates our optimal scheduler. Details of our *DBAS* implementation is presented in Section 5. We evaluate our scheduling techniques in Section 6. Section 7 discusses the related work and how *DBAS* compares to them. Finally, Section 8 concludes the paper and provides directions for future work.

2. Motivating scenario

Most mobile devices today are equipped with multiple heterogeneous network interfaces. For example, John's mobile device is equipped with Wi-Fi, WiMax and 3G. When John is sitting in a mall, waiting for a train at the station, or having a meal, he typically watches YouTube videos, listens to podcasts, uses Facebook and Twitter to get his social network feeds, and opens CNN and BBC to read the news. He connects his mobile device to available Wi-Fi hotspots while being connected to his 3G and WiMax networks.

Unfortunately, his current operating system assigns all his connections to Wi-Fi only. Consequently, John has to wait until the YouTube video is buffered in order to watch it continuously without disruption; meanwhile other applications are slowly retrieving their content over the same interface. With the high contention for Wi-Fi from other users, the available bandwidth is degraded and John disconnects his device from Wi-Fi to utilize his high bandwidth 3G or WiMax connections. Although, this increases his throughput, the available bandwidth is not sufficient to smoothly retrieve the content, in addition to the added overhead of restarting all his connections as well. However, his YouTube video will continue buffering from the point it reached when John was using Wi-Fi since YouTube servers support connection resume.

From this scenario, we observe that John's needs can be satisfied by concurrently utilizing his available network interfaces. Since John has multiple connections running concurrently on his device, these connections can be scheduled across the different interfaces in order to enhance his experience. Such scheduler should take the network interface characteristics, bandwidth heterogeneity, and different application characteristics into account when scheduling different connections to the available interfaces. Furthermore, the scheduler should be able to utilize the fact that some servers support the resume functionality in order to seamlessly migrate connections assigned to an interface that got disconnected to another. The scheduler can also leverage this resume functionality to schedule the data at a finer granularity in order to increase John's overall throughput.

3. A deployable bandwidth aggregation system

In this section, we provide an overview of the *DBAS* architecture and components which are depicted in Fig. 1. We consider a client device, equipped with multiple network interfaces, and connected to the Internet. Each interface has its own parameters in terms of bandwidth, latency, and loss ratio. The device is running multiple applications with different communication characteristics. In its default *connection-oriented* mode, *DBAS* schedules different connections to the interfaces such that a connection can be assigned to only one of the available interfaces. Once assigned to an interface, all the packets of this connection utilize the same interface. *DBAS*, therefore, achieves bandwidth aggregation without requiring any support from or changes to the legacy servers. On the other hand, if the end server happens to be DBAS-enabled or the application layer protocol used at the other end supports connection resume, our system leverages this fact to further enhance the performance by switching to a *packet-oriented* mode, where each packet or group of packets can be scheduled