Spider: Improving Mobile Networking with Concurrent Wi-Fi Connections

Hamed Soroush
University of Massachusetts
hamed@cs.umass.edu

Peter Gilbert
Duke University
gilbert@cs.duke.edu

Nilanjan Banerjee
University of Arkansas
nilanb@uark.edu

Brian Levine
University of Massachusetts
brian@cs.umass.edu

Mark Corner
University of Massachusetts
mcorner@cs.umass.edu

Landon Cox
Duke University
lpcox@cs.duke.edu

ABSTRACT

We investigate attempting concurrent connections to multiple Wi-Fi access points (APs) from highly mobile clients. Previous multi-AP solutions are limited to stationary wireless clients and do not take into account a myriad of mobile factors. We show that connection duration, AP response times, channel scheduling, available and offered bandwidth, node speed, and `dhcp` *joins all affect performance. Building on these results, we present a system, Spider, that establishes and maintains concurrent connections to 802.11 APs in a mobile environment. While Spider can manage multiple channels, we demonstrate that it achieves maximum throughput when using multiple APs on a single channel.*

Categories and Subject Descriptors

C.2.m [**Computer Systems Organization**]: Computer-Communication Networks—*Mobile and Wireless Systems*

General Terms

Concurrent Wi-Fi, Mobile Networks

1. INTRODUCTION

To realize their fullest extent of gains, Wi-Fi systems can aggregate a large number of access points (APs) *concurrently* to achieve improved network characteristics, unlike cellular, where devices are relegated to a single access point assignment. Recent *virtualized* Wi-Fi systems, such as VirtualWi-Fi [1], FatVAP [2], and Juggler [3], have shown that stationary users connected to multiple APs can achieve up to $3x$ greater bandwidth than users connected to a single AP [2]. These systems work by switching between APs on *multiple channels* rapidly, aggregating bandwidth at the client.

We show that multi-AP solutions designed for stationary users, like the ones mentioned above, are not useful at all in truly mobile scenarios. Our results show that at higher speeds, mobile users receive better performance by connecting to multiple APs only if they appear on the same channel. Only at lower speeds can mobile users recover from the throughput loss resulting from `dhcp` joins to APs on separate channels. In the extended technical report of this work [4], we present a general model of this problem that isolates the critical factors that determine an optimal schedule using one or more channels. These factors include the user's speed, the AP's `dhcp` response time, the AP's offered bandwidth, and the attained bandwidth. Based on this model, we find the mentioned *dividing speed* to be about 10 m/s (∼22 mph) in a typical environment; mobile users moving at this speed or faster would have to form concurrent Wi-Fi connections only within a single channel. Furthermore, we empirically show that link-layer association, `dhcp` and, TCP performance are affected negatively by multi-channel solutions.

We present a practical version of a mobile, virtualized Wi-Fi system called Spider that is designed for high-speed mobile users. Evaluation of Spider on a vehicular testbed crystalizes tradeoffs between throughput and connectivity. Spider can be used to manage and schedule joins to APs on multiple channels, and at a serious penalty to achieved bandwidth as our model predicts [1].

2. CHALLENGES

Joining to Multiple APs. Since APs can be instructed by the client to buffer packets, concurrent connections between a static client and multiple APs [1–3] are possible for Wi-Fi. The client falsely claims it is entering *power-save mode* (PSM), implicitly asking the AP to queue the incoming packets, and then communicates with another AP. Given that the backhaul bandwidth is typically smaller than the wireless bandwidth, such a scheme results in higher aggregate throughput if switching delays are kept very short. Static multi-AP solutions are not concerned with the delay incurred by the process of joining to the APs, and they do not need to be. In a static scenario joining process (association and `dhcp`) happens once, and its duration is negligible compared to the total connection time. In a mobile Wi-Fi environment, on the other hand, clients must continuously associate and obtain `dhcp` leases from APs as they become available. In addition, the packets associated with the join process cannot be buffered by the PSM request, and therefore, the client cannot switch away without reducing its chances of getting a `dhcp` lease.

To evaluate how the amount of time spent on the wireless channel affects the success rate of getting a `dhcp` lease, we performed several experiments, each lasting six hours on five vehicles moving around a small town, representing hundreds

[1]Detailed information about our analysis and design are available in our technical report [4] at `http://www.cs.umass.edu/publication/docs/2011/UM-CS-2011-016.pdf`.

SIGCOMM'11, August 15–19, 2011, Toronto, Ontario, Canada.
ACM 978-1-4503-0797-0/11/08.

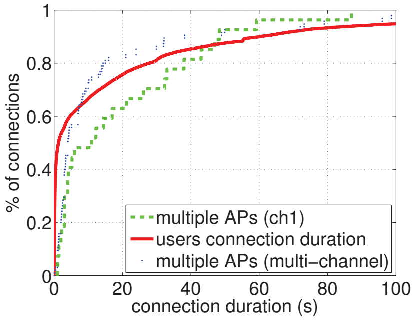

Figure 1: Comparison of connection lengths for wireless users and Spider.

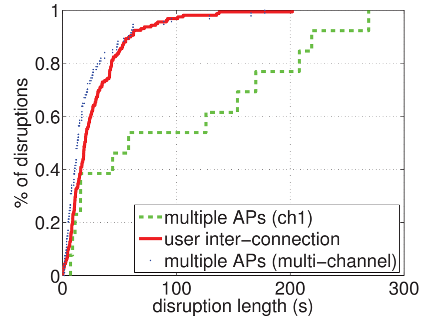

Figure 2: Comparison of disruption lengths for wireless users and Spider.

of trials. Since join delays are affected by link-layer timeouts, we reduced them from a standard of $1s$ to $100ms$ in these experiments. Detailed results of these experiments are presented in [4]. Our results show that reducing both `dhcp` and link-layer timeouts has a significant effect on performance, and that link-layer associations are robust to channel switching. However, while reconfigured `dhcp` timers are a boost to performance, we could not make `dhcp` robust to low fractions of scheduled time. This result suggests that the driver's time cannot be divided among more than two channels at 50% each in a mobile setting where the duration of time in range of an AP is limited.

Sustaining "Concurrent" TCP Connections. In a practical setting, if the channel schedule is skewed towards spending a large fraction of the time on a single channel, TCP connections on an orthogonal channel can timeout, potentially strangling performance. There is an inherent tension between the probability of successfully associating with APs on one channel and sustaining TCP connections on another channel. The results of our indoor experiments [4] show that when the total scheduling time is fixed, the TCP throughput increases monotonically with amount of time spent on AP's channel. However, when the total scheduling time is varied instead, the throughput increase is non-monotonic. This behavior is a result of increasing the total schedule which increases the amount of time *spent away* from the channel which can lead to TCP timeouts.

The Dividing Speed. In addition to these experiments, we have developed and validated a model in [4], that predicts the probability of obtaining a `dhcp` lease from an AP as a function of the amount of time spent in range in truly mobile scenarios. Based on the model, we have formulated an optimization framework to determine schedules that maximize aggregate throughput for a multi-AP solution. Our framework suggests that for highly mobile networks (where the average node speed is greater than $10m/s$), the best policy to maximize bandwidth is to stay on a *single* channel.

3. SOLUTION: SPIDER

Based on the results of our model and experimental analysis briefly presented in Section 2, we have designed and implemented Spider, a system that leverages concurrent 802.11 connections to improve performance in highly mobile networks. Our implementation is a freely available, open source Linux kernel module. Unlike previous workSpider designed for static scenarios that slice time across *individual* APs, Spider schedules a physical Wi-Fi card among 802.11 channels as our analysis suggests is most appropriate to do. Use of per-channel queues in Spider's driver allows it to communicate with *all* the APs on the same channel simultaneously with no switching overhead. In [4], we show that selecting multiple APs while maximiz-ing a given system utility function is NP-hard and use a practically efficient heuristic for that purpose. We have evaluated Spider on a vehicular platform in two different cities [4].

But can open Wi-Fi solutions such as Spider cater to connectivity needs of mobile users? To answer this question, we performed a study using data from a permanent Wi-Fi mesh we deployed in our downtown. The mesh consists of 25 nodes and covers an area of about 0.50 km^2. We collected performance data on all TCP flows from 161 wireless users for an entire day. Although, all users might not be mobile, the data provides us with a plausible baseline. We compare the traffic needs of wireless users with those provided by Spider based on two key metrics: (1) distribution of the duration of TCP connections, and (2) distribution of inter-connection time. Fig. 1 compares the TCP flow lengths gathered from actual users using our mesh network and Spider in its multi-channel and single-channel modes. The figure shows that Spider can support all the TCP flows that users need. Additionally, in Fig. 2 we compare the time between two connections for the mesh users and disruption time for Spider. When Spider uses multiple channels and multiple APs, it experiences disruptions comparable to what real users can sustain.

These results present Spider as a plausible complement to cellular data services. However, more data on mobile user's connectivity needs and network usage pattern is required in order to find out the degree to which Spider can *align* itself with the needs of each individual user. Conducting this study forms part of our future work with Spider.

4. REFERENCES

[1] R. Chandra, P. Bahl, and P. Bahl. MultiNet: Connecting to Multiple IEEE 802.11 Networks Using a Single Wireless Card. In *Proc. IEEE INFOCOM*, March 2004.

[2] S. Kandula, K. C.-J. Lin, T. Badirkhanli, and D. Katabi. FatVAP: aggregating AP backhaul capacity to maximize throughput. In *Proc NSDI*, pages 89–104, 2008.

[3] A. Nicholson, S. Wolchok, and B. Noble. Juggler: Virtual Networks for Fun and Profit. In *IEEE Trans. Mobile Computing*, 2009.

[4] H. Soroush, P. Gilbert, N. Banerjee, B. N. Levine, M. D. Corner, and L. Cox. Spider: Improving mobile networking with concurrent wi-fi connections. *UMass UM-CS-2011-016 Technical Report*, 2011.