

# Improving User's Experience through Simultaneous Multi-WLAN Connections

Guillem Cañizares, Boris Bellalta  
Wireless Networking Group  
Universitat Pompeu Fabra, Barcelona  
guillem.canizares@upf.edu, boris.bellalta@upf.edu

## Abstract

In highly-dense IEEE 802.11 deployments, areas covered by multiple Wireless Local Area Networks (WLANs) will be common. This opens the door for stations equipped with multiple IEEE 802.11 interfaces to use several WLANs simultaneously, which not only may improve user experience, achieving a better connection with higher throughput and resilience; but it may also improve the network utilization. In this paper we investigate such a scenario. First, using a test-bed, consisting of a single station equipped with two interfaces and two access points, we observe that the file transfer time between the station and a destination server can be significantly reduced, studying with special attention the case in which both links do not have the same available bandwidth. Then, using a Markovian model that captures the scenario's dynamics in presence of multiple stations, we observe that in addition to improve individual station's performance, we can also improve the utilization of a multi-Access Points network despite increasing the contention level.

## Categories and Subject Descriptors

C.2.5 [Computer Systems Organization]: COMPUTER - COMMUNICATION NETWORKS—*Local Wide-Area Networks*

## General Terms

Design, Experimentation, Evaluation

## Keywords

Multiple 802.11 interfaces, WLANs

## 1 Introduction

Future Wireless Local Area Networks (WLAN) scenarios presume the existence of multiple overlapping IEEE 802.11 WLANs over the same area [3, 4]. In those places under coverage from different Access Points (APs), the use of multiple IEEE 802.11 interfaces –physical or virtual– at the same

time can be a promising solution to improve both the user experience and network utilization.

To the best of our knowledge, we have not found any other paper focusing on a similar scenario. However, there is a large number of works that consider the existence of multiple wireless interfaces in the stations. For example, Brik et al. propose a multiple radio system in [6] to eliminate the handoff latency in WLANs, reducing it to around 30-40 ms using their *MultiScan* approach.

There are also radio virtualization proposals such as the Picasso [7] project. Authors propose a radio design that allows simultaneous transmission and reception on separate spectrum fragments, using a single RF front end and antenna. They provide evidences that their prototype can virtualize a single radio into separate independent frequency slices, and achieve the same cumulative throughput than using the same number of real interfaces. Furthermore, authors in [1] stand that the use of virtual wireless interfaces to connect to multiple networks saves energy, minimizes the physical space, and improves the coexistence in dense deployments.

Based on this, the incorporation of a transport protocol that is able to use different interfaces simultaneously, such as Multipath TCP (MPTCP), would become crucial in order to provide a better user experience. Arzani et al. [2] expose that the use of a congestion controller and packet schedulers are basic for MPTCP performance, while path election and buffer sizes have also a significant impact. The authors of [8] analyze the benefits that MPTCP can provide to a wireless connection. They test single path, 2 and 4 Multi Path connections and observe the impact of the flow size on the average latency, concluding that the latency achieved by MPTCP is comparable to the smallest latency produced by either Wi-Fi or LTE connections in single path, except for small files under MB size.

In this paper, instead on focusing on the performance of multiple paths to simply transmit data, we focus on a more fundamental problem. We investigate if the use of multiple IEEE 802.11 interfaces, and simultaneous connections to multiple APs, can be a suitable solution to improve the overall network performance in dense scenarios, providing some evidences about its potential benefits and drawbacks. Namely,

1. For the case of a single station, we study through experiments the performance gains when multiple IEEE 802.11 interfaces placed in a single node are used si-

multaneously, considering unbalanced links and background traffic.

2. We develop a simple analytical model to quantify the performance gains when multiple interfaces are used in scenarios with multiple stations.

The rest of the paper is organized as follows. Section 2 introduces the Testbed, the Java Application developed and the experiments done. The tests performed and the obtained results are extensively explained. In section 3 we show the analytical results for scenarios with multiple stations. Finally, we present the conclusions in section 4.

## 2 Test-bed and Experiments

### 2.1 Test-bed

Figure 1 shows the testbed we used for the experiments. It consists of a server (HP Compaq 600 pro) and a client (DELL Latitude 5580) connected through a router and two wireless access points -AP1 and AP2- (TP-Link AC1750). The client uses two wireless interfaces (Intel Dual Band Wireless-AC 8265 and Realtek TP-Link TL-WN822N). Each wireless access point is located 1 meter away of the client, being separated the same distance between them. The wireless system operates using the 802.11n standard. The Server and both APs are directly connected to the router. Each one of them belong to different subnets.

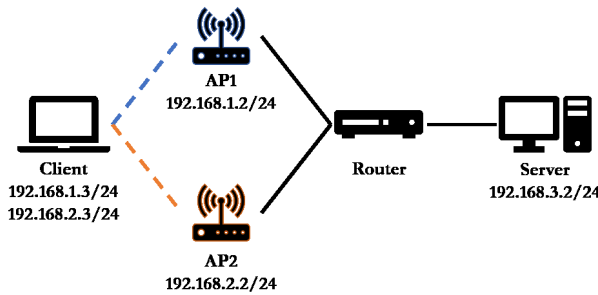


Figure 1. Testbed

Ubuntu 16.04.3 LTS (Xenial Xerus) is installed on both client and server. The two APs use channels 1 and 11 respectively, minimizing any interference between them. The average signal strength measured at the client's device is  $-33$  dBm in both interfaces. The RTT between the client and the server is less than 5 ms for both of them. Our testbed is not connected to the Internet to avoid the presence of external traffic that could have an impact on the results.

Figure 2 illustrates the client operation. The application, besides generating data, distributes the packets to the corresponding interfaces through a scheduler. The scheduler splits the file in different chunks and distributes them between all the interfaces based on the instantaneous available bandwidth of each link. For this to happen, information about the state of the network is sent to the application.

### 2.2 Application

We have developed a Java Application (APP) which connects the client and the server through a multi-socket connection. An independent thread is used to control the data transmission for each active wireless interface. Interfaces are

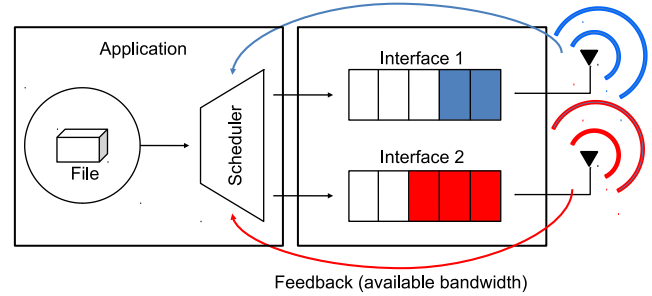


Figure 2. Illustration of the client operation

bound to the correspondent IP address and server port number. The APP sends files from the client to the server, and returns the file transfer time. The files can be sent using one or two interfaces following different split ratios, thus generating multiple data chunks, and distributing them between both interfaces. Using a  $n$  split ratio means that the file is divided in  $n$  chunks. Then, the first chunk is sent through the slow link, and the remaining  $n - 1$  are sent through the fast link.

### 2.3 Experiments

We focus on the file transfer time when files of different sizes are sent from the client to the server. We designed a test bench that included a total of ten files which size increases from 1 MB to 128 MB. Every test –i.e., a single file upload– is reproduced ten times in order to obtain reliable results. Three different experiments were done:

1. **Upload Test:** test files are split 50/50 (i.e.,  $n = 2$ ) and uploaded to the server. The total file transfer time using two interfaces is evaluated and compared to the case in which a single interface is used.
2. **Different Link Throughputs Test:** test files are uploaded to the server when the achievable throughput in one of the paths is significantly lower than in the other. The total file transfer time using a 50/50 splitting ratio is compared to the case in which the best splitting ratio is used.
3. **Background Traffic Test:** test files are uploaded to the server in different network conditions, as one of the links carries background traffic sent from the client to the server. The total file transfer time using a 50/50 splitting ratio is compared to the case in which the best splitting ratio is used.

### 2.4 Results

In this section we present and discuss the obtained results in the tests presented above.

#### 2.4.1 Upload Test

In this first test, files are first uploaded to the server using only a single link (the slower one, Link 2). Then, a 50/50 split is performed to the file and each file chunk is uploaded to the server through a different interface. Figure 3 shows the file transfer time when a single and two interfaces are used. For instance, for the 128 MB size, we observe that the use of 2 interfaces reduces the transfer time more than 3 minutes.

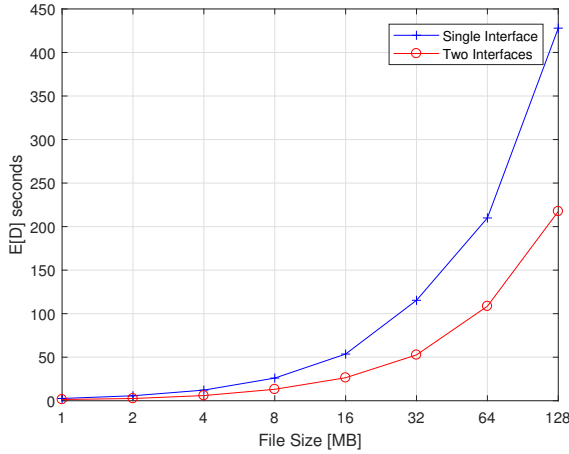


Figure 3. File Transfer Time (First Test)

We observe that the ratio between the file transfer time when a single and two interfaces are used is always around 0.5. This means that the file transfer time using two interfaces lasts closely half of the time using a single interface.

#### 2.4.2 Different Link Throughputs Test

In this second test, files are uploaded to the server in different network conditions. Link 1 throughput calculations estimate an average of 2 Mbps, while Link 2 works 6 times faster, around 12 Mbps. Calculations are computed using *iperf*<sup>1</sup> and *ifstat*<sup>2</sup> tools. Files are first split 50/50 to observe how the presence of different transmission rates affect the file transfer time. Then, a second split ratio is applied in order to achieve a similar time in both links. Two different file sizes are tested in this scenario: 16 MB and 64 MB.

Figure 4 shows the results for the second test. The client transmits the 16 MB file using both links, sending through each one 8 MB –i.e., half of the file– in a 50/50 split. It can be seen that the chunk transmitted through the slower link has a transfer time far larger than the time of the chunk transmitted through the fast link. To reduce this difference, a second splitting ratio is applied. This time, we split the file using a splitting factor of  $n = 6$ , sending the 83.3% of the file through the fast link, while the remaining 16.7% is sent through the slow link. We observe that using this fairer split we achieve a 71% reduction in the file transfer time. However, Link 2 transfer time is incremented by 67%, even though this increment does not harm the total connection time. Similar results are obtained for the 64 MB file. As in the previous case, using the best splitting ratio –i.e.,  $n = 8$ – reduces the file transfer time in a 81% and increments Link 2 by 34% respectively.

#### 2.4.3 Background Traffic Test

In this third test, TCP throughput values for both links are between 10 and 12 Mbps. However, one of the two links also carries background traffic. The background traffic is generated using the *iperf* tool and uses the UDP protocol. Two

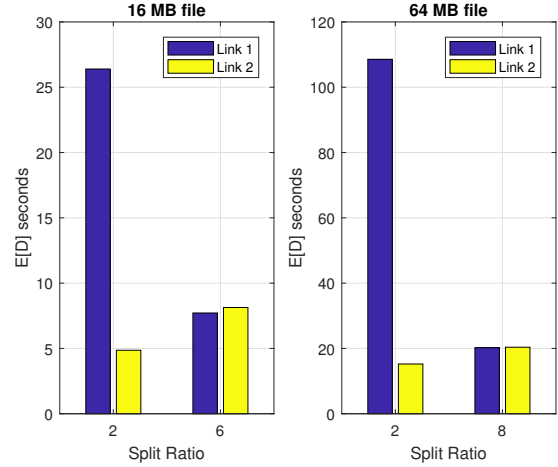


Figure 4. File Transfer Time for different file sizes (Second Test)

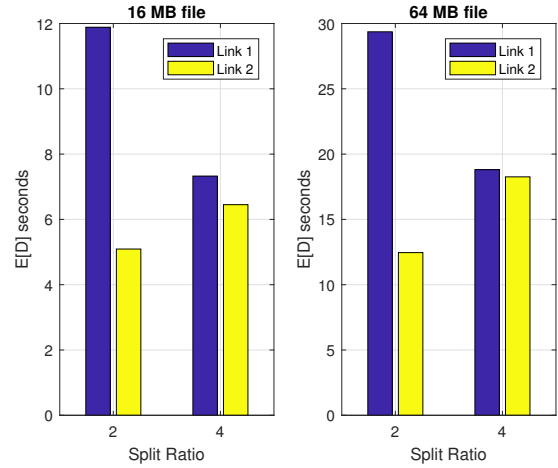


Figure 5. File Transfer Time for different file sizes and background traffic loads (Third Test)

different file sizes are considered: 16 MB and 64 MB, and a 10 MB flow of background traffic is used for the tests.

Figure 5 shows the results obtained in this third experiment. The background traffic is injected in Link 1, reducing the available bandwidth for the file transmission. Similarly to the previous experiment, we can see that the file transfer time is improved using a fairer split ratio that takes into account the amount of background traffic. This time, we reschedule the file packets using a splitting factor of  $n = 4$ , thus, sending the 25% of the file only through the now busy Link 1, while the remaining 75% is sent through the idle link. The total transfer time for 16 MB file is reduced by 30%, and 36% in the 64 MB file case. We also see in both cases that Link 2 transfer time is slightly increased.

### 3 Performance Analysis with Multiple Stations

Once we have experimentally validated that the simultaneous use of two 802.11 interfaces associated to two differ-

<sup>1</sup> <https://iperf.fr/>

<sup>2</sup> <http://gael.roualland.free.fr/ifstat/>

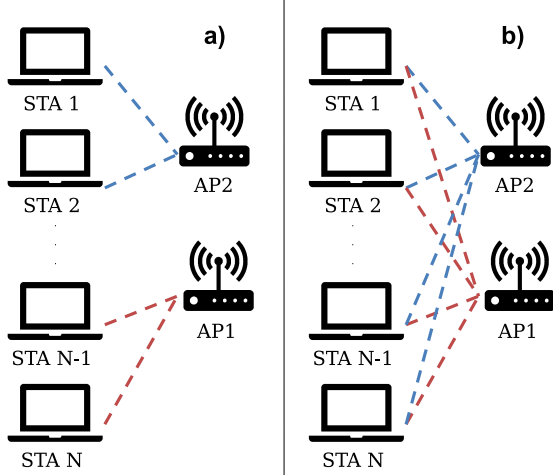


Figure 6. The two scenarios considered in the analysis

ent APs results in significant performance gains in terms of file transfer time when we have a single user in the network, we study if such gains can be extended when there are multiple stations sharing both APs.

Let us consider the two scenarios depicted in Figure 6. In scenario a), stations are equipped with a single interface, and are fairly distributed between the two APs. In case of an even number of stations, the AP to which the last station is associated is randomly selected. In scenario b), all stations are equipped with two interfaces, and are simultaneously associated to both APs. We assume that all interfaces (at the APs and stations) use always a 64-QAM modulation and a 3/4 coding rate, which for packets of size  $L = 12000$  results in a transmission duration of 0.253 ms. Since the basic access scheme is employed, the duration of a collision is the same as the duration of a successful transmission.

We consider that the two APs operate in different channels (i.e., 1 and 11, as in previous section), and all interfaces associated to a given AP are able to listen the transmissions from the others interfaces associated to the same AP. A station receives new files of average size  $F$  bits to upload with rate  $\lambda$  when it is idle. Then, it becomes active and starts transmitting the new file until it is completed. The file transmission rate,  $\mu$ , depends on the instantaneous throughput provided by the network, which depends on both the number of interfaces available at the station and the number of active stations in the network.

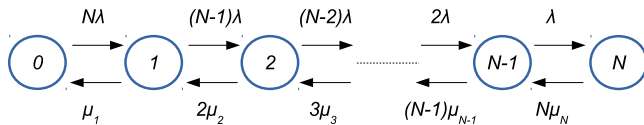


Figure 7. Markov chain used to characterize the number of active stations

By assuming both inter-file arrivals and file transmission times are exponentially distributed, the described system can be modelled using the Markov chain shown in Figure 7,

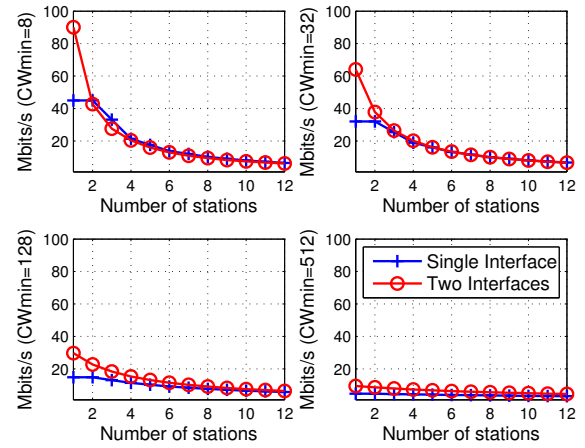


Figure 8. Single station throughput when the number of active stations in a WLAN increases

where states represent the number of active stations, forward transition rates represent the rate at which stations become active, and backward transition rates represent the rate at which files are transmitted. The equilibrium distribution of such a Markov chain is given by

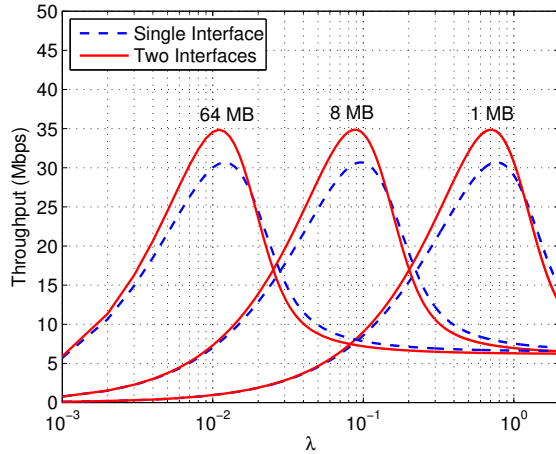
$$\pi_i = \frac{\prod_{j=1}^i \frac{(N-j+1)\lambda}{j\mu_j(M)}}{1 + \sum_{z=1}^N \prod_{j=1}^z \frac{(N-j+1)\lambda}{j\mu_j(M)}} \quad (1)$$

where  $M$  is the number of interfaces of each station. Backward transition rates ( $\mu$ ) are given by  $\mu_i(M) = \frac{S_i(M)}{F}$ , with

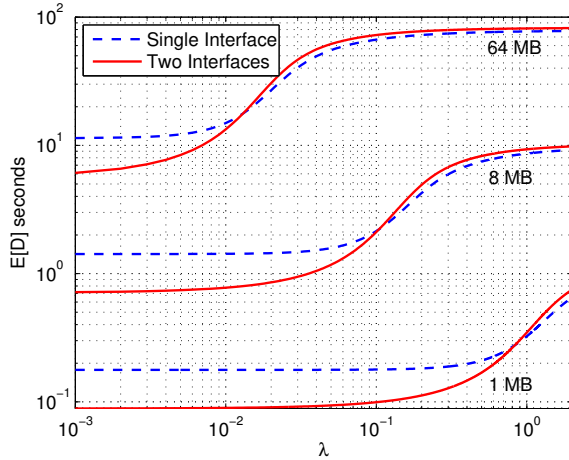
$$S_i(M) = \begin{cases} \frac{1}{2} (B_{AP1}(\lceil \frac{i}{2} \rceil) + B_{AP2}(\lfloor \frac{i}{2} \rfloor)) & M = 1 \\ B_{AP1}(i) + B_{AP2}(i) & M = 2 \end{cases} \quad (2)$$

the average throughput achieved by a single station when there are  $i$  active stations in the network and  $M$  interfaces are used.  $B_{APn}(u)$  is the saturation throughput of a single station associated to AP  $n$  computed using Bianchi's 802.11 throughput model [5].

Figure 8 shows the value of  $B_{APn}(i)$  when the number of active stations in the network increases for  $CW_{\min} = 8, 32, 128$  and  $512$ , and  $CW_{\max} = 2^5 CW_{\min}$ . We can observe that despite the relative throughput when using two interfaces is higher for larger  $CW_{\min}$  values than in the case of using a single one, the highest throughput is achieved by a  $CW_{\min} = 8$  in both cases. For  $CW_{\min} = 8$ , however, the use of two interfaces only results in a higher throughput when there is a single station active in the network. Otherwise, the higher collision probability when all stations use their two interfaces (i.e., it is the same as having twice the number of active stations in each AP compared to the case when a single interface is used) results in a slightly lower throughput. Therefore, the use of two interfaces will only be an interesting solution if the amount of time there is only one active station in the network is able to compensate otherwise. To dig on that situation, we plot the average per-user throughput and file transfer time when  $\lambda$  increases for three different file sizes. Figures 9 and 10 show that using two interfaces



**Figure 9.** Expected per-user throughput,  $E[S] = \sum_{i=0}^N \pi_i S_i(M)$



**Figure 10.** Expected file transfer time for different file sizes. The file transfer time is calculated by applying Little's formula,  $E[D] = \frac{\sum_{i=0}^N i \pi_i}{\sum_{j=0}^N (N-j) \lambda \pi_j}$

we are able to obtain a higher average per-user throughput and a lower file transfer time compared to the case only one interface is used for a reasonably range of  $\lambda$  values.

#### 4 Conclusions

In this paper, we have investigated if the use of multiple IEEE 802.11 interfaces is able to improve the user experience and network utilization in scenarios with multiple overlapping WLANs. To do that, we have first evaluated experimentally the time reduction when a station uploads a single file to a server using two IEEE 802.11 interfaces at the

same time, compared to the case a single interface is used. We have also studied the impact of the file splitting ratio between the two interfaces, showing the importance to find a splitting ratio that balances the amount of data send through each link proportionally to its available bandwidth.

We have complemented the experimental results by analyzing the network performance in presence of multiple contending stations using a Markovian model. The analysis done shows that the use of multiple interfaces can also be a feasible solution to improve the performance of a multi-AP network when there are several active stations, despite the higher contention that appears when multiple interfaces are used.

This paper shows promising but just preliminary and exploratory results in a topic that has to be further investigated in the next years. Next steps include testing MPTCP in high-density WLANs, including node mobility and intermittent connectivity to multiple and miscellaneous APs. Moreover, we plan to extend the analysis done to completely describe the potential gains of such an approach, and develop new protocols able to get the most of it as well.

#### 5 Acknowledgments

This work has been supported by a Gift from the Cisco University Research Program Fund (CG#890107, Towards Deterministic Channel Access in High-Density WLANs), a corporate advised fund of Silicon Valley Community Foundation.

#### 6 References

- [1] Y. Al-Hazmi and H. de Meer. Virtualization of 802.11 interfaces for wireless mesh networks. In *Wireless On-Demand Network Systems and Services (WONS), 2011 Eighth International Conference on*, pages 44–51. IEEE, 2011.
- [2] B. Arzani, A. Gurney, S. Cheng, R. Guerin, and B. T. Loo. Impact of path characteristics and scheduling policies on MPTCP performance. In *Advanced Information Networking and Applications Workshops (WAINA), 2014 28th International Conference on*, pages 743–748. IEEE, 2014.
- [3] B. Bellalta. IEEE 802.11ax: High-efficiency WLANs. *IEEE Wireless Communications*, 23(1):38–46, 2016.
- [4] B. Bellalta, L. Bononi, R. Bruno, and A. Kassler. Next generation IEEE 802.11 Wireless Local Area Networks: Current status, future directions and open challenges. *Computer Communications*, 75:1–25, 2016.
- [5] G. Bianchi. Performance analysis of the IEEE 802.11 distributed coordination function. *IEEE Journal on selected areas in communications*, 18(3):535–547, 2000.
- [6] V. Brik, A. Mishra, and S. Banerjee. Eliminating handoff latencies in 802.11 WLANs using multiple radios: Applications, experience, and evaluation. In *Proceedings of the 5th ACM SIGCOMM conference on Internet Measurement*, pages 27–27. USENIX Association, 2005.
- [7] S. S. Hong, J. Mehlman, and S. Katti. Picasso: Flexible RF and spectrum slicing. *ACM SIGCOMM Computer Communication Review*, 42(4):37–48, 2012.
- [8] C. Paasch, S. Ferlin, O. Alay, and O. Bonaventure. Experimental evaluation of multipath TCP schedulers. In *Proc. of the 2014 ACM SIGCOMM workshop on Capacity Sharing*, pages 27–32, 2014.