Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Distributed Computing

Selfish Strategies to Increase Your Throughput in 802.11 Networks

Bachelor Thesis

Kilian Risse

krisse@ethz.ch

Distributed Computing Group
Computer Engineering and Networks Laboratory
ETH Zürich

Supervisors:
David Stolz
Prof. Dr. Roger Wattenhofer

August 5, 2015

Abstract

Today, more and more devices access the internet through wireless LAN. As the bandwidth of access points is limited, it frequently happens that the throughput is unsatisfactorily small. We assume that a user is connected to the internet via an access point, and that the user wants to increase his throughput using selfish measures. Observe that in the typical use case the bottleneck is not the direction from the user to the access point (upstream), but the opposite direction (downstream). This is due to the fact that an HTTP request is usually smaller than it's response. Hence, to effectively increase the throughput, we focus on improving the downstream. We therefore study how access points distribute incoming packets, and we propose attacks to abuse the behaviour of access points. In particular, we study how the power saving mode U-APSD (supported by an increasing number of modern routers) can be exploited. With U-APSD the router is required to buffer the packets per client, and we show that a user can gain a bigger share of the totally available bandwidth by multiplexing his network interface card.

Contents

Abstract i

1 Introduction 1

2 Scenario 2

2.1 Terminology 2

2.2 A Crucial Question 2

2.2.1 Global Queue 3

2.2.2 Queues Per Client 3

3 Test Setup 5

4 Results 7

4.1 A Guide to Reading the Figures 7

4.2 Technicolor TC7200-U 8

4.3 TP-LINK Archer C5 AC1200 11

4.4 Apple Time Capsule 4th Generation (A1409) 13

4.5 Thomson TWG870U 16

5 Conclusion 19

Bibliography 20

Chapter 1

Introduction

Who has never experienced a slow wireless LAN (WLAN) connection? I am sure all of you have experienced this issue – whether during a lecture or while at work – and wished to have a larger bandwidth. In the typical use case, where we want to browse the internet or fetch some email, the bottleneck of the connection is usually the direction from the access point (AP) to the client: The downstream. This is due to the asymmetric nature of these applications: The required downstream bandwidth is larger than the bandwidth in the opposite direction (upstream). Several ways to increase the upstream bandwidth have been proposed (e.g. by reducing the backoff time), but as the bottleneck is the downstream, these are not very helpful in a congested WLAN. Therefore we focus on the downstream bandwidth and try to increase our downstream share by exploiting the way APs distribute incoming packets to associated clients. As we will show later, different APs distribute incoming packets differently, hence there is no generic way how we can increase our bandwidth. But for each AP we studied, we developed a strategy how we can increase the downstream share. If the AP supports the power saving mode U-APSD [1], it is possible to increase our downstream share without a proxy server; otherwise we need a proxy server.

Chapter 2

Scenario

We assume that there are users connected with an AP via WLAN. We are one of these users and would like to increase our downstream share. This is trivial if the WLAN link is not congested, so we assume congestion (i.e., the AP is overloaded). Further we assume that the WLAN link is the bottleneck of the connection for all users. Therefore, if we can somehow manage to increase our share of the WLAN link, we are able to achieve a larger throughput. This is our objective throughout this thesis.

From now on, unless noted otherwise, we only discuss the downstream.

2.1 Terminology

For the sake of unambiguousness we introduce some terminology:

user a person trying to access the internet

client a device associated with an access point

adversary a selfish user

server a device sending data to a client

Note that a user may act as multiple clients.

2.2 A Crucial Question

In order to increase the bandwidth there is a crucial question we need to answer first: Does the access point buffer packets *per client* or *globally*? Globally meaning that there is only a single queue for all packets. Depending on the answer, we have to choose the appropriate strategy as we will argue in the following.

2.2.1 Global Queue

Definition 2.1. An access point has a *global queue*, if it buffers packets in a single queue (for all clients) and distributes them in a FIFO manner.

Theorem 2.2. *If an access point has a global queue, a client's loss rate is independent of the corresponding server's send rate – and therefore also independent of the achieved throughput.*

Proof. Let $s_i > 0$ denote the send rate of a server, sending data to client i. Now let's assume that we have an empty buffer slot that gets filled with the next packet that arrives at the AP. If we assume that the packets arrive independently, the probability that the free slot gets filled by a packet for client i is $p_i = \frac{s_i}{\sum_{j=1}^{|clients|} s_j}$. Therefore the receive rate for client i is $r_i = \frac{p_i \cdot B}{s_i}$, where B stands for the bottleneck bandwidth. The bottleneck bandwidth is equal for all clients as we assumed that the bottleneck of all connections is the WLAN link. Now if we replace p_i in the last formula by the definition we get

$$r_i = \frac{s_i}{\sum_{j=1}^{|clients|} s_j} \cdot \frac{B}{s_i} = \frac{B}{\sum_{j=1}^{|clients|} s_j},$$

which is independent of s_i. □

Selfish Strategy

We require a proxy server outside of the WLAN. Instead of contacting the target server directly, the proxy is used to duplicate (possibly many times) the packets sent to the adversary. The proxy will send all packets (including the duplicates) to the adversary. As the adversary receives twice as many packets as a different client requesting the same bandwidth, she will also receive twice the bandwidth of the other client. This follows from Theorem 2.2.

As the proxy server duplicates the adversary's stream, the probability that a packet arrives increases. This is better than without duplicating her stream, but the adversary is still unable to receive the entire stream. In order to deal with the packets that were lost, the adversary can additionally ask the proxy to encode packets with an error correcting code, such that she is able to recover the lost packets.

2.2.2 Queues Per Client

Definition 2.3. An access point has *queues per client* if it maintains a dedicated buffer for each associated client. The access point distributes the packets by contiuously iterating over the buffers and sending an equal amount of packets per buffer.

Selfish Strategy

So what can we do in such a scenario? Our strategy is to multiplex our network interface card; we act towards the AP as multiple clients. Then we split the downstreams over these virtual clients. Such hardware is available on the market (e.g. TP-LINK TL-WDN4800) and is usually used to connect to multiple WLANs simultaneously with different MAC addresses. If we assume that each client gets the same share of the bandwidth, which we will show in the following, we can increase our share of the bandwidth and are better off than ordinary clients.

Let B denote the total bandwidth available. Then $b = \frac{B}{|users|}$ is the fair share of the bandwidth that each client has available; if a client requests $b_{req} \leq b$ he is supposed to receive the whole b_{req}. In order to show that all clients have at least b available we do a case destinction:

Case 1: All client buffers are always full. As the access point distributes the packets in a round robin fashion, all clients receive the same amount of packets in a given time. Therefore all clients receive b.

Case 2: At least one client's buffer is not always full. Therefore this client requested less than b. As long as his buffer still contains packets, these are distributed as in the previous case. When there are none left, the access point will 'skip' this buffer and continue with the others. Therefore he receives his whole b_{req} and all other clients will receive slightly more than b.

If we multiplex our card k times, what is the lower bound for our bandwidth? Let b_k denote the bandwidth we get by multiplexing our card k times and let n be the number of users. Note that b depends on the number of *users* and not the on number of *clients*.

$$\underbrace{\left(\frac{B}{n+(k-1)}\right)}_{\text{= the new "fair" share}} \cdot k = \frac{n \cdot k}{n+k-1} \cdot b \leq b_k \tag{2.1}$$

Equality holds iff all users requested more than b. The advantage is not a factor k because whenever we multiplex our card once more, the bandwidth each client receives is reduced.

Chapter 3

Test Setup

To observe the packet distribution of an access point we used the following setup: We connected n computers to the access point by WLAN: The clients. Further we connected another n computers with the access point by Ethernet. The latter serve as our servers and each one sends a UDP stream to a client; there is always one server associated with one client. The setup for $n = 4$ is displayed in Figure 3.1. The servers send their packets evenly distributed over time as otherwise the access point would get flooded with packets at the beginning of every second and we would be measuring the size of the buffer in the access point. The size of the payload of each UDP packet sent by a server is 1000 bytes. This will most likely not give us maximum throughput in Mbit/s, due to the maximum transmission unit (MTU) being larger. But as we are only interested in the loss rate, which we will compare with other clients, and as they all receive equally sized packets, this should not influence the results.

For each test we average over eight runs of the following procedure: The servers send their streams for 100 seconds to the clients. After the time has elapsed we collect the number of packets received per client. This can then be used to calculate the average loss rate per client.

As the test network is not perfectly isolated from the environment, it happens that other wireless networks interfere. This can easily be spotted as usually one of the test runs has a much lower throughput. We will then repeat the test by running the test another eight times and average over the new runs.

In order to decide whether an access point has a global queue or client queues, we will perform the following test series: One server sends a small stream of 8 Mbit/s to a client. All other servers send equal streams that increase after each test. If we observe that for all tests in the series all clients have equal loss rates we conclude that we have a global queue, as it aligns with the characteristics of the global queue model. If the small stream has a significantly lower loss rate than the others, we conclude that the access point has client queues or something similar. This series will be conducted with different amount of clients to test whether there is a limit in the number of client queues.

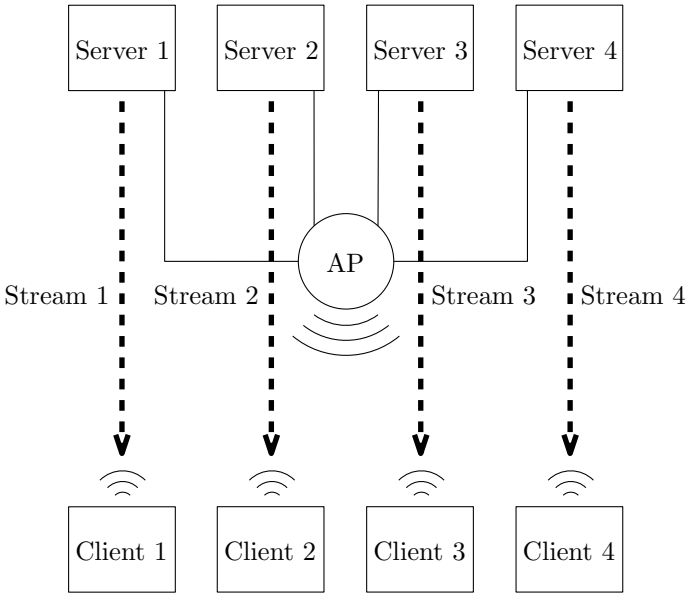


Figure 3.1: Test setup for $n = 4$. The servers are connected to the access point by Ethernet, the clients by WLAN.

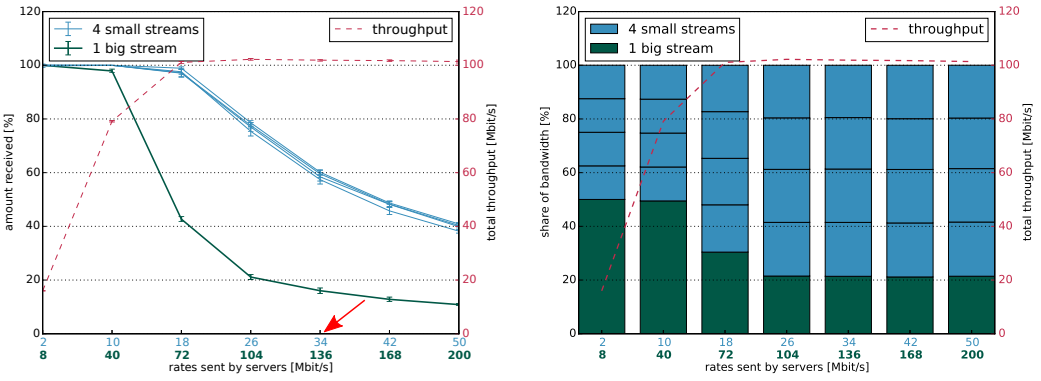
Chapter 4

Results

We studied four access points in detail. In the following sections, we discuss the results for each access point seperately.

4.1 A Guide to Reading the Figures

There are two types of figures. You can see them next to each other in Figure 4.1. Both figures represent the same data. In this figure we tested the AP with five different streams. The left figure shows how many percent of the data sent to the clients were received. The clients are grouped by the rate the servers attempted to send to them; i.e. if the servers sent equal bandwidth to the clients they have equal color. In both figures the total achieved throughput is displayed as a dashed line. Notice that the throughput has it's own scale on the right side of the figure.



(a) An ordinary user competing against an adversary, who multiplexes her data on four small streams (clients).

(b) As soon as the network is saturated, all clients have an equal share of the bandwidth. As the adversary multiplexes her stream four times, she has four times the share of the ordinary user.

Figure 4.1: Adversary competing against an ordinary user.

If you look at the marked position in the figure, you can see that the client, who is trying to receive a bandwidth of 136 Mbit/s, receives roughly 15%, while the others, each trying to receive 34 Mbit/s, receive about 60% of their stream. The total throughput is at roughly 100 Mbit/s. The throughput has stagnated for the previous tests and shows that the WLAN is saturated. This test shows that an adversary who multiplexes her data can achieve a lower loss rate than an ordinary user. Note that the ordinary user (the big stream) attempts to receive the same amount as the four small streams in total, i.e. the ordinary user requested the same bandwidth as the adversary did.

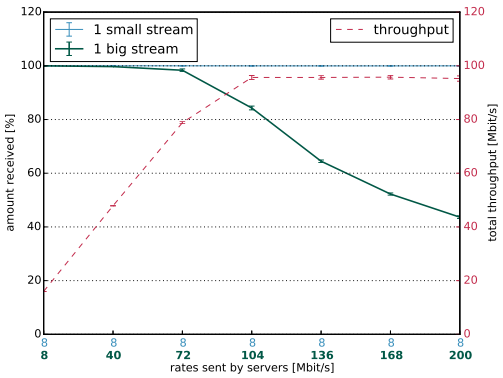
The right figure shows the share each client receives of the total bandwidth. Again, the streams that attempt to receive the same bandwidth are equally colored. This figure complements the other figure as we can see the overall distribution of the bandwidth. The errorbars were omitted for representational reasons – they clutter the figure and contain no additional information.

Figure 4.1b shows how each stream gets the same share of the bandwidth as soon as the WLAN is saturated. This is dependent on the AP. Again, if we assume that the small streams belong to the adversary, she gets four times the ordinary user's bandwidth. In the next section we will look at this AP in more detail.

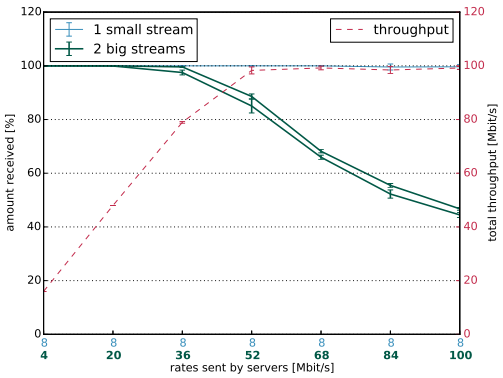
4.2 Technicolor TC7200-U

Our results show that this access point has client queues. Figure 4.2 shows a test series where a single small stream with 8 Mbit/s competes against a number of big streams. The small stream never experiences loss, while the big streams have considerable loss. This is a perfect fit for the client queue model. But Figure 4.2d does not fit into the picture; it indicates that there is a limit in the number of client queues. But if you look at Figure 4.3, you see that the AP has more than four client queues. We do not know how to explain the behaviour of the AP in Figure 4.2d.

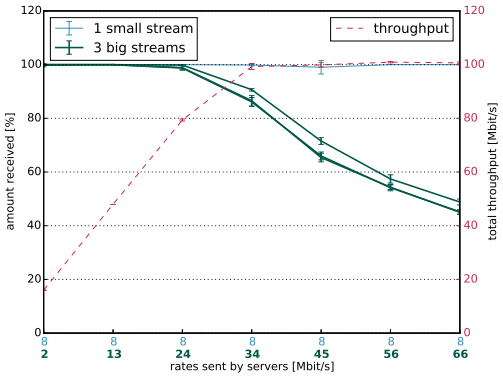
Figure 4.3 shows the advantage an adversary may gain by distributing her load onto several clients. The adversary distributes her load over multiple clients (the small streams) and competes against the ordinary users (the big streams). Note that the ordiary users and the adversary all attempt to receive the same bandwidth. Figure 4.3a and Figure 4.3d show that the advantage gets diminished as the number of users increases; the adversary needs to multiplex her card more agressively.



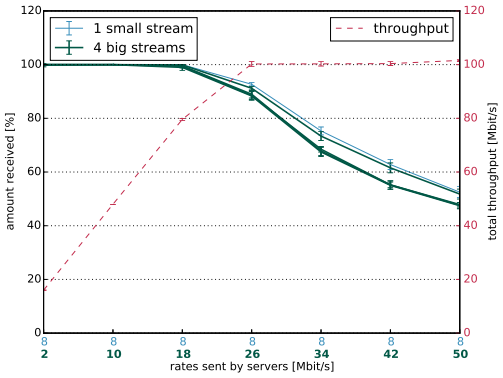
(a) small stream vs. one big stream



(b) small stream vs. two big streams

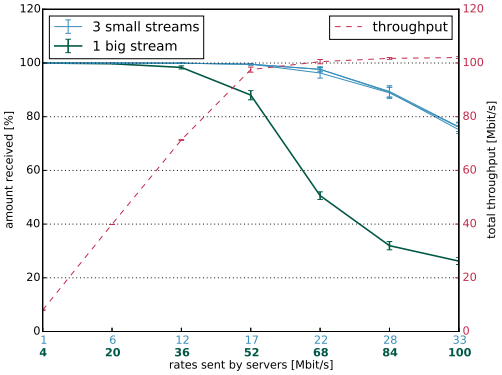


(c) small stream vs. three big streams

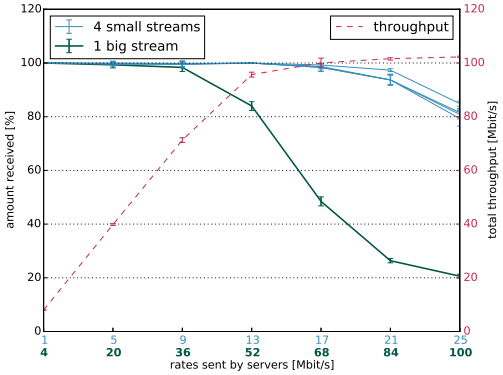


(d) small stream vs. four big streams

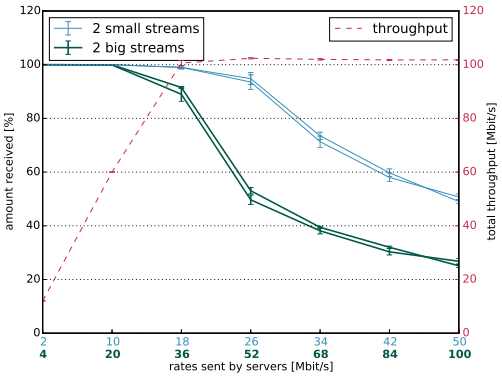
Figure 4.2: The behavior of Technicolor TC7200-U when we have a small stream competing against a number of big streams. The first three figures match our queues per client model.



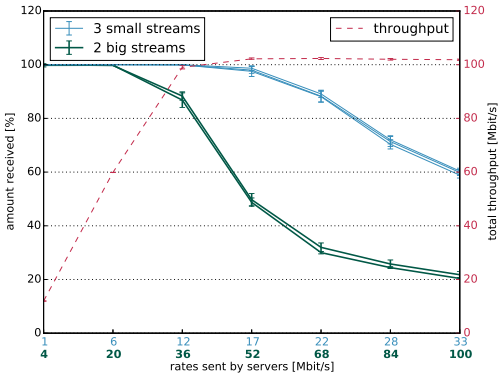
(a) an ordinary user vs. the adversary, multiplexing her card three times



(b) an ordinary user vs. the adversary, multiplexing her card four times



(c) two ordinary users vs. the adversary, multiplexing her card two times

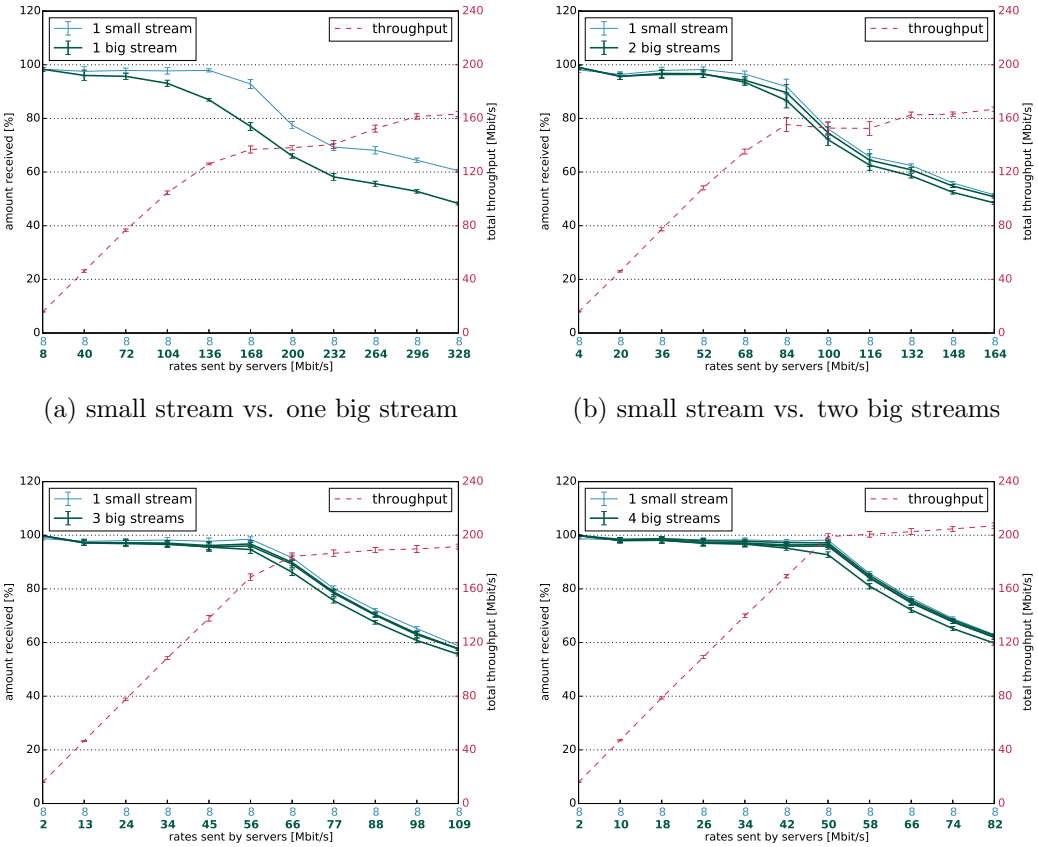


(d) two ordinary users vs. the adversary, multiplexing her card three times

Figure 4.3: In case of the Technicolor TC7200-U, an adversary gains an advantage over the ordinary users if she distributes her load onto several clients.

4.3 TP-LINK Archer C5 AC1200

In contrast to the first access point we studied, we will now look at an access point that has a global queue. In Figure 4.4 the small stream has roughly the same loss rate as the big streams, which shows that the AP has a global queue. The loss rates are not always identical but there is a substantial difference to the previous AP, see, e.g. Figure 4.2. The total throughput was not always the same. It seems to depend on the number of clients; the more clients, the higher the total throughput.



(a) small stream vs. one big stream

(b) small stream vs. two big streams

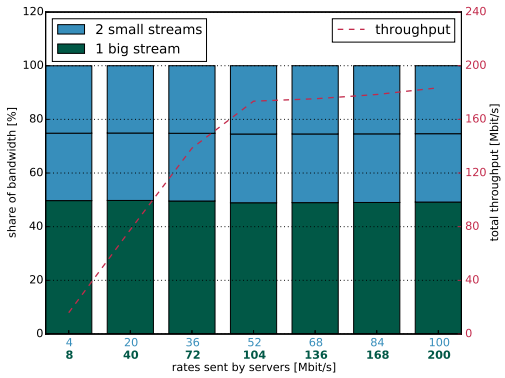
(c) small stream vs. three big streams

(d) small stream vs. four big streams

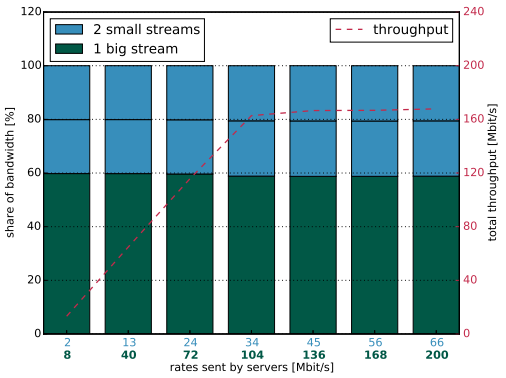
Figure 4.4: The TP-LINK Archer C5 AC1200 has a global queue. The difference between the small stream and the big stream is probably due to the fact that the small stream has a larger buffer relative to its send rate; a bigger share of the stream can be stored in the buffer.

As we know that we have an AP with a global queue, we are going to test whether the selfish strategy, as outlined previously, works. As a reminder: By

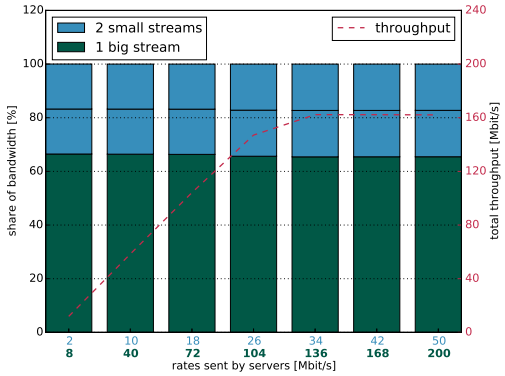
requesting a larger bandwidth than the ordinary users, the adversary hopes to increase her share of the bandwidth. In Figure 4.5 you can see that the adversary gets a larger share of the bandwidth than the ordinary users – our strategy works. Keep in mind that the adversary is the big stream. But as can be seen in Figure 4.5d, the more clients there are, the more aggressively the adversary needs to multiply her stream in order to achieve the same share of the bandwidth as previously.



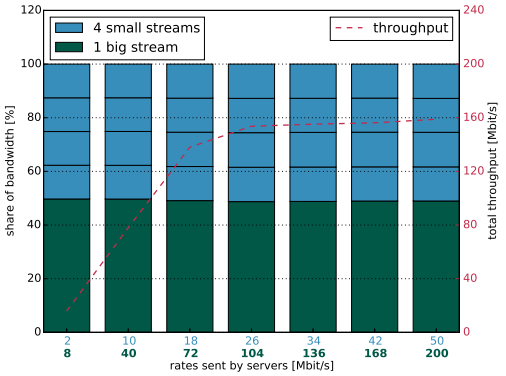
(a) adversary doubling her traffic vs. two ordinary clients



(b) adversary tripling her traffic vs. two ordinary clients



(c) adversary quadrupling her traffic vs. two ordinary clients

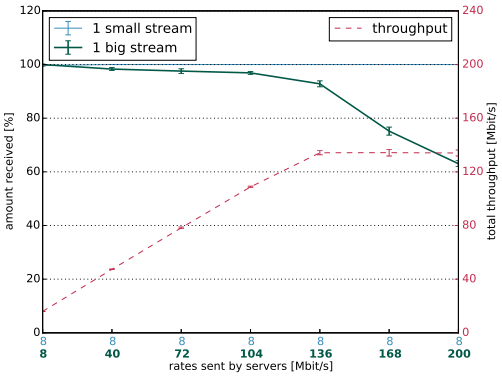


(d) adversary quadrupling her traffic vs. four ordinary clients

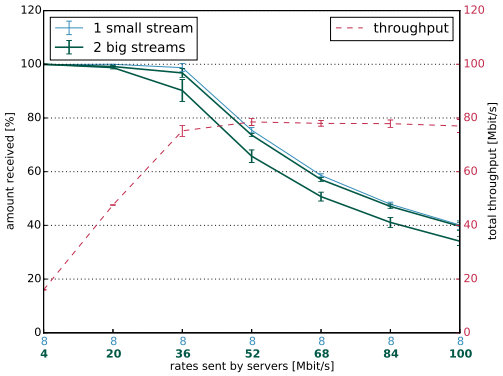
Figure 4.5: For the TP-LINK Archer C5 AC1200, the adversary can increase her share of the bandwidth with the help of a proxy that duplicates her stream. The big stream depicts the adversary.

4.4 Apple Time Capsule 4th Generation (A1409)

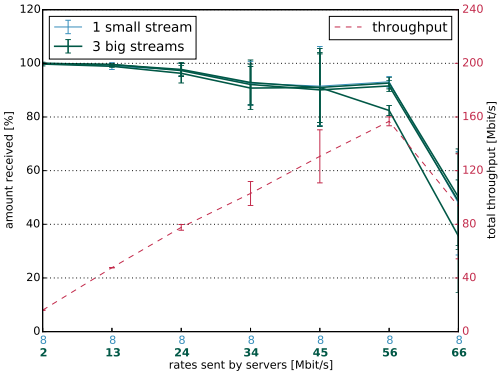
This AP does not quite fit one of our models. In Figure 4.6a the AP behaves like an AP with client queues. But as you look at other plots in Figure 4.6, you see that the AP behaves differently with more streams; like an AP with a global queue. This behaviour could be explained by an AP with a maximum of two client queues. Now take a look at the subplots (a) - (c) of Figure 4.7. These show a distinct difference in the loss rate between the small and the big streams. So even though we have more than two streams, the AP behaves as if it had client queues.



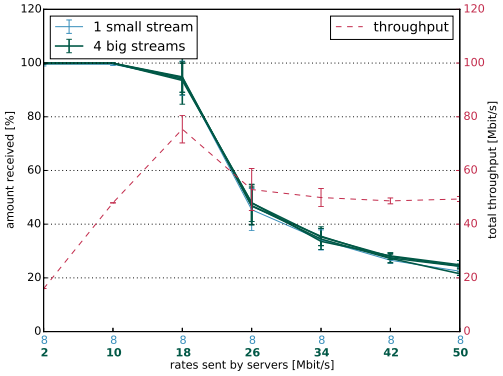
(a) small stream vs. one big stream



(b) small stream vs. two big streams



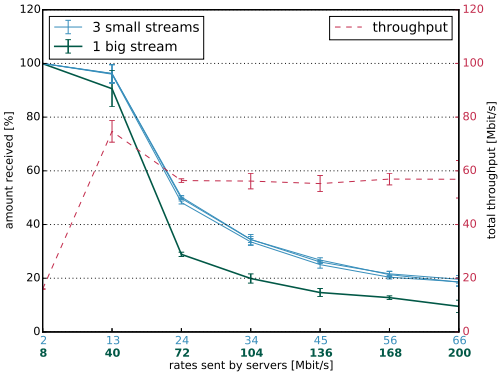
(c) small stream vs. three big streams



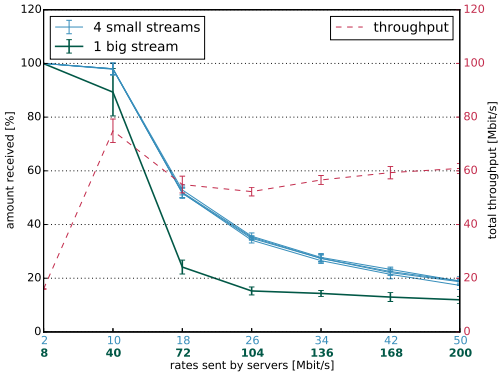
(d) small stream vs. four big streams

Figure 4.6: Figure (a) indicates that the Apple Time Capsule 4th Generation has client queues. But in the three other scenarios the AP acts as if it had a global queue.

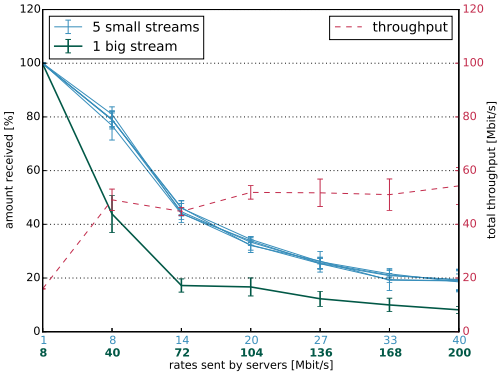
So we need a new model for this AP. For this purpose we let a *dominant stream* be a stream that is at least as large as the sum of all other streams. For example



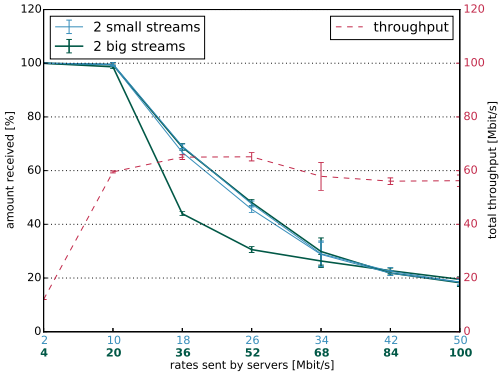
(a) three small streams vs. one big stream



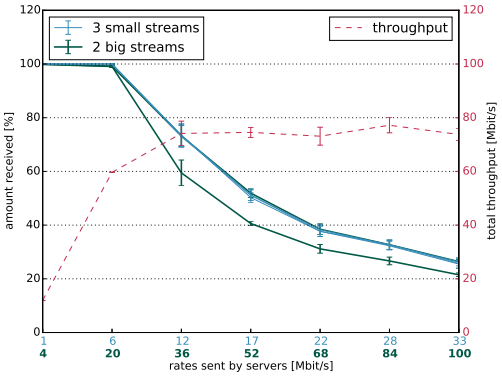
(b) four small streams vs. one big stream



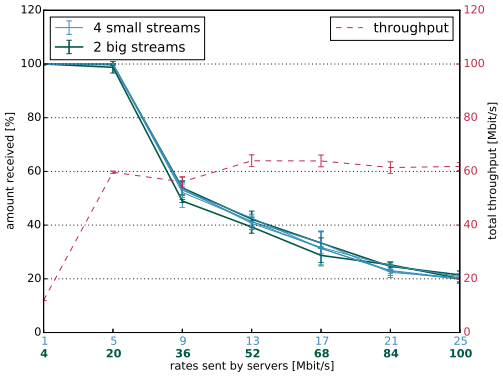
(c) five small streams vs. one big stream



(d) two small streams vs. two big streams



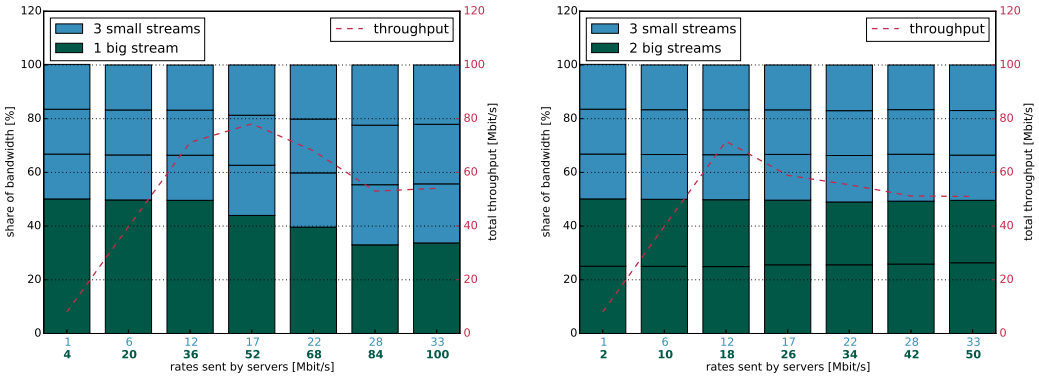
(e) three small streams vs. two big streams



(f) four small streams vs. two big streams

Figure 4.7: The sum of the small stream's attempted throughput is equal to the attempted throughput of a single big stream. These tests were conducted on an Apple Time Capsule 4$^{\text{th}}$ Generation.

the ‘big stream’ in Figure 4.7a is a dominant stream. Further a stream gets *punished*, if its client receives a considerably lower percentage of its stream than all other streams. The ‘big streams’ in Figure 4.6a and the subplots (a) - (c) in Figure 4.7 are dominant streams, which get all punished. Further we observe from tests (c) - (f) in Figure 4.7, that if no dominant stream is present, no stream gets punished. There is still a difference between the streams, but this difference is much smaller than in scenarios (a) - (c). Therefore we conjecture that this AP has a global queue and additionally, it punishes dominant streams. In Figure 4.8 you can observe that this seems to hold; i.e., if there is no dominant stream the clients receive their share dependent on the requested bandwidth, whereas if there is a dominant stream, that stream gets punished. This model is not perfect, as it cannot explain the difference in the received rate for the subfigures (d) and (e) in Figure 4.7. But in most other cases it describes the behaviour quite well. So what shall the adversary do if she connects to such an AP? First she needs to create an additional client and distribute her load equally onto her two clients. Now the adversary can simply follow the strategy from the global queue. As she distributed her stream onto two clients, neither stream may become dominant (assuming that there are other clients associated with the AP).



(a) The big stream is dominant.

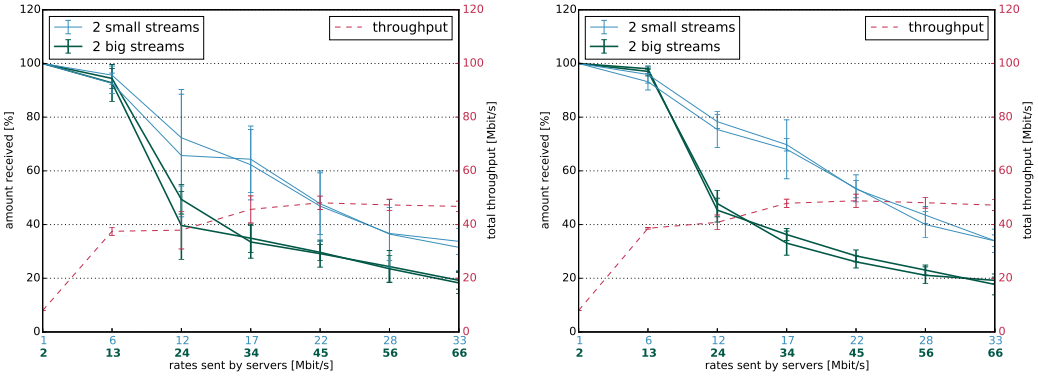
(b) The big streams are not dominant.

Figure 4.8: In case of the Apple Time Capsule 4$^{\text{th}}$ Generation, the dominant stream gets punished.

What troubles us further is the total throughput of this AP. Take a look at Figure 4.6. In subfigures (a) and (c) the AP has a total throughput well above 100 Mbit/s. But in all other cases (not just this figure) the total throughput never exceeds 80 Mbit/s. We have no explanation for this effect.

4.5 Thomson TWG870U

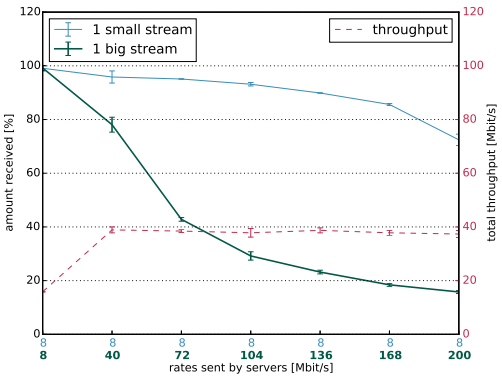
With the testing procedure described in the previous chapter, we were unable to get good results for this AP; i.e., no results that had reasonable standard deviation. In most cases the problem was that two or three runs, out of the eight, performed very poorly; either they had a low overall throughput or some clients received almost no packets. We repeated the tests multiple times but the results were always the same. We suspect that the AP contains bugs which lead to this behaviour. Therefore we adapted the testing procedure: We chose the five best runs out of the eight such that the average standard deviation is minimized. Figure 4.9 displays the effect of this selection procedure. All following figures will be averaged over the five best test runs.



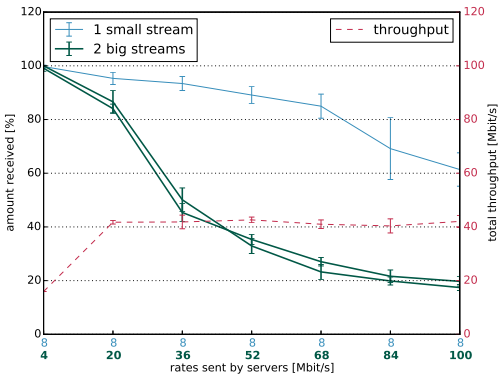
(a) (b)

Figure 4.9: Plot (a) shows the average over all eight test runs, whereas (b) displays the average over five test runs chosen out of the eight, such that the average standard deviation is minimized. These tests were run on a Thomson TWG870U.

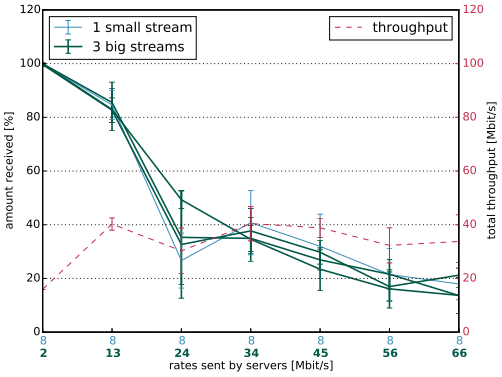
The AP has client queues as can be seen in subfigures (a) and (b) of Figure 4.10. Subfigures (c) and (d) indicate that the AP has only three client queues. But Figure 4.11 shows that the AP posesses more than three client queues. Again, we have no explanation for the behaviour of the AP in scenarios (c) and (d) in Figure 4.10.



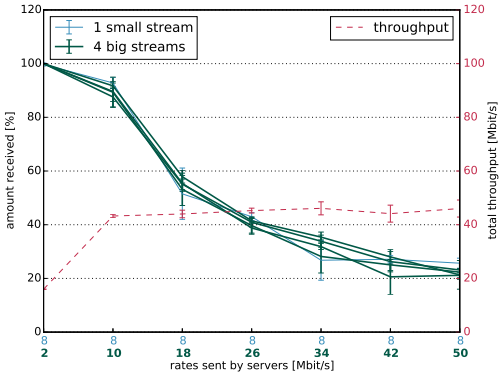
(a) small stream vs. one big stream



(b) small stream vs. two big streams

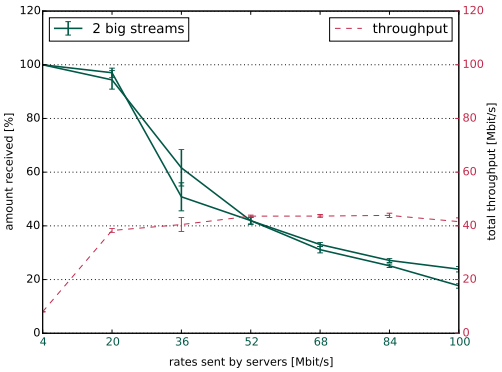


(c) small stream vs. three big streams

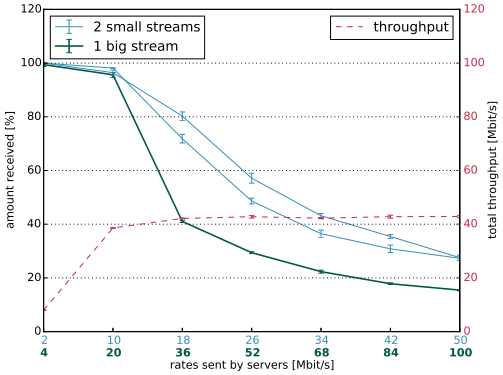


(d) small stream vs. four big streams

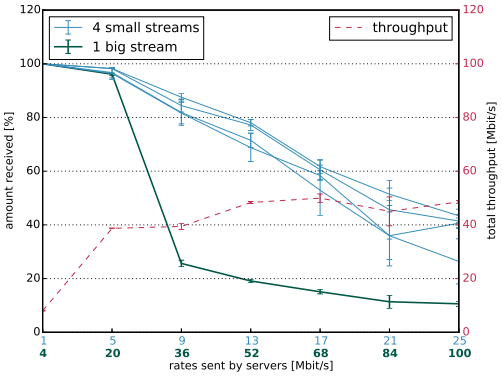
Figure 4.10: In scenarios (a) and (b) the Thomson TWG870U performs as expected for the queues per client model and in the other scenarios as expected for the global queue model.



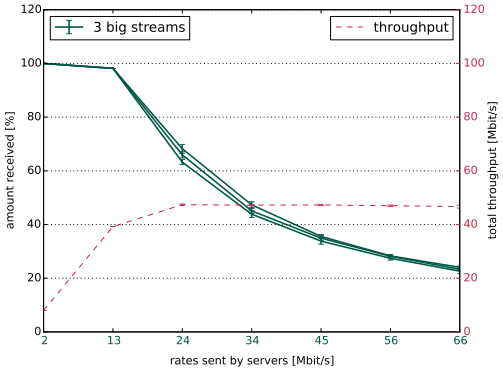
(a) two ordinary users



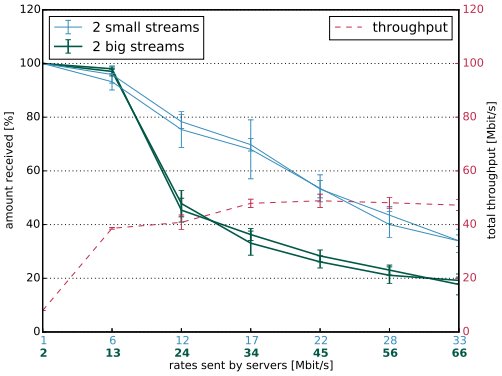
(b) an ordinary user vs. the adversary multiplexing her card two times



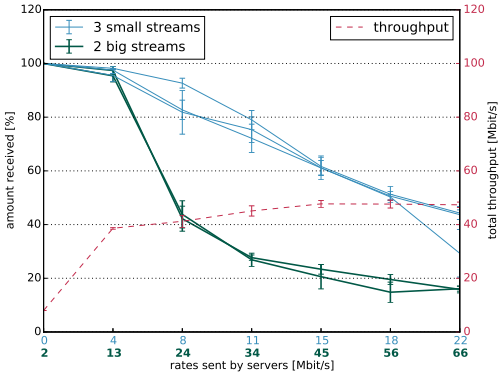
(c) an ordinary user vs. the adversary multiplexing her card four times



(d) three ordinary users



(e) two ordinary users vs. the adversary multiplexing her card two times



(f) two ordinary users vs. the adversary multiplexing her card three times

Figure 4.11: These figures show the advantage an adversary can achieve. Figures (a) and (d) show the behaviour of the Thomson TWG870U if no adversary is present. The subsequent figures show the advantage an adversary may gain by multiplexing her network interface card.

Chapter 5

Conclusion

Access points are more distinct in their implementations than we expected at first. Sometimes we were unable to come up with a reasonable model for the behaviour of certain APs. But we expect that the implementations of the APs get more uniform, as more devices support the power saving mode U-APSD. As U-APSD requires the AP to buffer packets per client, this can be abused by multiplexing the network interface card. We initially wanted to write a proof of concept but failed due to unexpected problems with the WLAN card: There are issues with the driver that prevented us from connecting with the AP multiple times. But as our test results show, such a strategy is not yet guaranteed to succeed with every AP. However, once U-APSD is widely adapted, we believe that such selfish measures allow an adversary to significantly improve her throughput, as our test results indicate.

Bibliography

[1] *IEEE Standard for Information technology–Local and metropolitan area networks–Specific requirements–Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications - Amendment 8: Medium Access Control (MAC) Quality of Service Enhancements*, Std., Nov 2005.