Controlling TCP Fairness in WLAN access networks using a Rate Limiter approach

Nicola Blefari-Melazzi, Andrea Detti, Alessandro Ordine, Stefano Salsano.

DIE, University of Rome "Tor Vergata"
{blefari,andrea.detti,alessandro.ordine,stefano.salsano}@uniroma2.it

Abstract— **This paper deals with application level QoS and fairness issues in IEEE 802.11 Wireless LANs. The current implementations of 802.11 use the DCF, which gives the same medium access priority to all stations. However, fairness at the MAC level does not necessarily translate in fairness at the application level, particularly in presence of TCP-based applications, operating in an infrastructured network topology. Unfairness at application level may result in users perceiving unsatisfactory levels of QoS. This paper addresses fairness related phenomena, as experienced by TCP-based applications, and proposes a solution able to offer fair performances to final users with minimal additional complexity. The proposed solution, based on rate control mechanisms, avoid TCP critical unfairness. Our solution operate at the IP level and is fully compatible with existing devices. Moreover, it seems appealing for its easiness of implementation. Also, we analyzed fairness phenomena and evaluated the performance of our solution by means of simulations.**

I. INTRODUCTION

Wireless LANs based on the IEEE 802.11 standard are shared media enabling connectivity in the so-called "hot-spots" (airports, hotel lounges, etc.), university campuses, intranet access in enterprises, as well as "in-home" for home internet access among others.

In all of the above-mentioned scenarios, WLAN coverage is based on "*Access Points"*, devices that provide the mobile stations with access to the wired network. These scenarios are often called "infra-structured" WLANs to distinguish them from the "ad-hoc" WLANs, where mobile stations talk to each other without using Access Points.

In the above scenarios, it is crucial to maintain fairness among the TCP connections competing for access to the shared media of the WLAN. By fairness among multiple TCP connections, we mean that any TCP engine would be capable of starting a connection with negligible delay, as well as achieving and maintaining a reasonable throughput. The latter, of course, depends on other competing TCP connections. Viewed this way, TCP fairness is, then, a mandatory pre-requisite for enabling a satisfactory quality service for upper layer applications. However, we also have to specify that we are not requesting a "perfect" fairness, i.e., a perfectly balanced sharing of resources among all TCP connections (which can be seen as a "second order" objective). Rather, our main aim is to avoid the scenario of "*critical unfairness*" that is characterized by complete starvation of some TCP connections or, even, the inability of some TCP connections to start altogether.

We introduce two critical unfairness cases. In the first case, downstream TCP connections are penalized with respect to upstream ones. This is explained as follows: packets belonging to multiple downstream TCP connections are buffered inside the Access Point wireless interface. Note that the Access Point does not enjoy a privileged access to WLAN capacity, with respect to user terminals. Hence, a single station transmitting upstream packets will get the same priority as that of the Access Point which needs to transmit downstream packets heading towards many stations. Thus, downstream TCP connections suffer because of the arising congestion and corresponding packet losses happening in the download buffer at the Access Point. These losses in conjunction with TCP congestion control mechanism cause the starvation of downstream connections. This is defined as "critically" unfair.

The second case arises from the interaction of multiple TCP connections in the upstream direction. In this case, the Access Point wireless interface has to transmit TCP ACK packets traveling downstream towards stations in the WLAN. Also, in this case, we have a bottleneck because the Access Point can not access the medium with a priority higher than other stations. Hence, the Access Point buffer will be congested leading to severe loss of TCP ACK packets. Due to the cumulative nature of TCP ACKs, few connections will be able to "survive" and open their window, while the majority of connections will get starved. Note that this situation is not specific of our scenario; it can happen in whatever environment characterized by heavy losses of ACK packets. This case is also another example of "critical unfairness".

It is worth-mentioning that the 802.11 standard also includes a different access control mechanism, called Point Coordination Function (PCF). An extension of the basic 802.11, namely the draft standard 802.11e, provides further mechanisms to control the allocation of the WLAN resources. Both the PCF and the 802.11e could be used to improve the fairness perceived at the application level. Unfortunately, the current status is that the large majority of existing WLAN cards and devices support neither the PCF functionality nor the 802.11e. Considering the lack of deployment of PCF and 802.11e, we focus on strategies to achieve TCP fairness by using the widely deployed DCF mechanism. Moreover, we focus our attention only on techniques that can be implemented within the Access Point (or in a nearby router), without requiring changes in the 802.11 standard, nor any enhancement to mobile stations.

In this paper, we propose a solution aiming at avoiding critical unfairness (i.e., starvation) and at enforcing a fair sharing of radio bandwidth between the Access Point and the mobile stations. Our approach is based on a rate-limiter, implemented via a Token Bucket Filter (TBF) [1]. The rate-limiter operates on the overall aggregate of uplink packets suitably dropping them as a function of its parameters. The TCP congestion control mechanisms, that are automatically enabled when losses are detected, reduce the transmission windows and consequently the number of transmitted packets. Thus, by setting the TBF parameters, it is possible to suitably control the overall uplink rate.

We will show that our solution are indeed very simple to implement and that the rate control is very effective in avoiding the starvation of TCP connections and resource wasting. In addition, our approach can provide the operator of the WLAN with a tool to controlling the sharing of WLAN resources between upstream and downstream applications.

II. SYSTEM MODEL AND PROBLEM DEFINITION

Our scenario is shown in Fig. 1. A number of wireless stations are connected to an Access Point and exchange information with a host in the high-speed fixed network. We consider N_{dn} wireless stations downloading information from a wired host and N_{up} wireless stations uploading information to a wired host. We simulated this environment by means of the NS-2 simulator package [2].

As shown in Fig. 1, the wired host can be connected to the Access Point via a Fast Ethernet LAN link, or via a generic duplex link with capacity *C* and one-way propagation delay *D*. The former represents the case in which the Access Point and the wired host are in the same Local Area Network ("local wired host"). The latter represents the case in which the wired host is remotely located somewhere in the Internet ("remote wired host"). We can set the Round Trip Time (RTT) between the wireless stations and the remote wired host to arbitrary values by choosing a proper value of *D*. In the same way, we can emulate a bottleneck in the Internet connection toward the remote wired host by properly setting the capacity *C*.

The most important simulation parameters that we adopt in this work are: IP packet size 1500 bytes, Maximum TCP congestion window 43 packets (64 KBytes), TCP version: Reno [3].

We also assume that TCP sources have always information to transmit (commonly denoted as "greedy source" and representing a worst case). As for the downlink buffer, when not otherwise specified, the downlink buffer size is 100 packets, which, according to [4], is a typical value for commercial equipments.

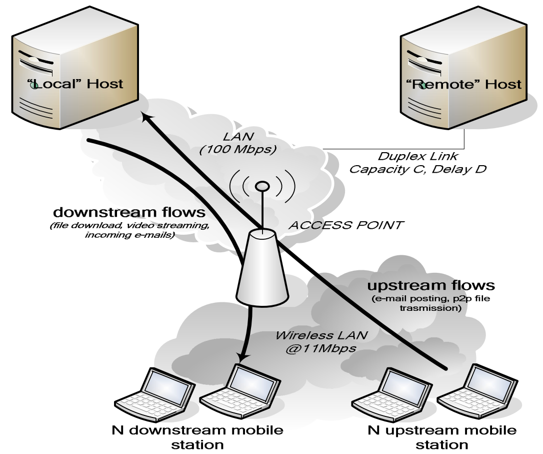

Fig. 1: Reference simulation scenario

The merit figures that we consider are the total upstream throughput R_{up_tot} (i.e., the sum of the IP-level throughputs of upstream TCP connections), the total downstream throughput R_{dn_tot} (i.e., the sum of the IP level throughputs of downstream TCP connections) and the total throughput R_{tot}. Assuming the presence of both upstream and downstream active "greedy" TCP connections, we state that critical unfairness can take place in two basic circumstances: interaction between upstream and downstream TCP connections (see Fig. 2) and interaction between a set of upstream TCP connections (see Fig. 3).

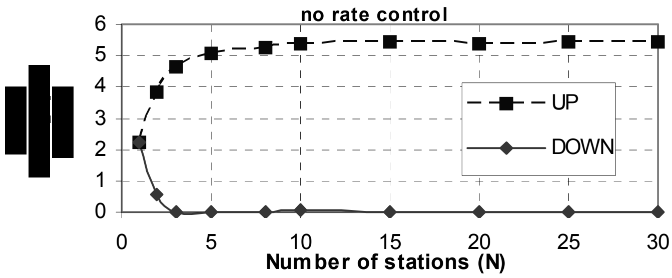

Fig. 2: Upstream and downstream throughput without rate control mechanisms

Fig. 2 shows the total upstream throughput and the total downstream throughput as a function of *N*. When *N*=1 the overall bandwidth is fairly shared. The throughput of downstream connections dramatically decreases as *N* increases and is almost equal to zero when *N*=4. Thus, downstream connections do not succeed in perceiving their "right" bandwidth share, even with a moderate number of upstream connections.

We state that the main cause of starvation, and unfairness, is the packet loss occurring in the downlink buffer. The packet loss in the downlink buffer may attain large values because of the DCF access mechanisms whose task is to fairly share the available capacity among all active entities, mobile stations, and Access Point alike. Since the Access Point does not enjoy a privileged access to WLAN capacity with respect to users' terminals, and since it has to handle more traffic with respect to a single station, it is more likely that its downlink buffer becomes congested, with respect to the buffering resources of mobile stations.

We now investigate why this loss leads to TCP starvation of some connections. We start by looking at downstream connections. For such connections, a packet loss in the downlink buffer means a TCP segment loss; TCP segment losses trigger congestion control mechanisms which, in turn, cause a decrease of the TCP throughput. In addition, both at the beginning of a connection and after the occurrence of several segment losses, the TCP congestion window is small in order to prevent the use of fast retransmit mechanisms. Hence, most of the losses are recovered by means of the Retransmission Time Out (RTO) mechanism. Since the RTO doubles after each consecutive loss (and consecutive losses are likely in the above conditions), downstream connections experience long idle period, and even throughput starvation, as shown in Fig. 2.

Let us now turn our attention to upstream connections. In this case, a packet loss at the downlink buffer means the loss of a TCP ACK. For large values of such loss probability several consecutive ACKs of the same connection may be lost. The impairments caused by consecutive ACK losses worsen as the TCP congestion window decreases [5]. For instance, assuming ideal conditions, with ACK losses being the only cause of performance degradations, and assuming a congestion window equal to W packets, the sender will find itself in the Retransmission Time Out state, and thus reduces its throughput, only if W ACKs are lost. As a consequence, the greater W, the rarer are RTO events. If we consider that the TCP congestion window increases when ACK segments are received, the probability of RTO events is maximized at the start of the connection. On the contrary, these events are always less likely to occur as the congestion window increases, and disappear once the window gets higher than a critical threshold (e.g., five packets). This chain of events is the cause of the behavior illustrated in Fig. 3.

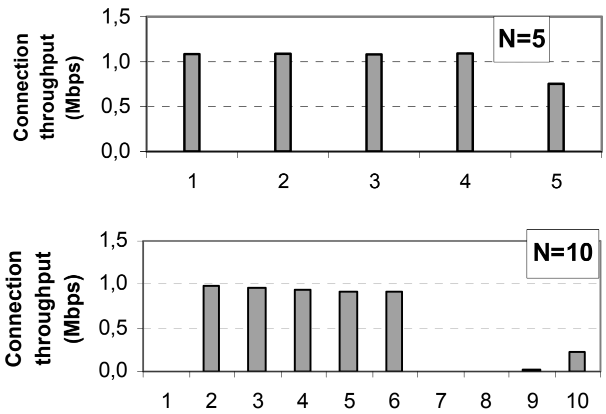

Fig. 3: Throughput of up connections ("local wired host scenario"); N=10

It is worth noting that upstream connections experience starvation for larger values of loss probabilities, as compared to downstream connections. For instance, in our scenario, the downstream starvation occurs when the loss probability is greater than 20% (and N=3), whereas upstream connections suffer this full service outage for loss probabilities greater than 50% (and N=10).

Fig. 4 shows a parameter that can be used to appraise the unfairness: the ratio between the standard deviation (σ_{up}) and the mean value ($R_{up}=R_{up\_tot}/N$) of the throughput of upstream connections. If the ratio σ_{up}/R_{up} is equal to zero, then we have perfect fairness; otherwise the greater the ratio, the greater the unfairness. In Fig. 4, the ratio σ_{up}/R_{up} sharply increases for N>5. To solve the unfairness problem, a possible solution would be to choose a size of the downlink buffer equal to B_{noloss} (a buffer size large enough to completely avoid loss phenomena in the AP downlink buffer). This choice would have also the advantage of guaranteeing to all connections the same throughput. However, such a large buffer size would have severe drawbacks on the value of the overall Round Trip Time experienced by TCP connections.

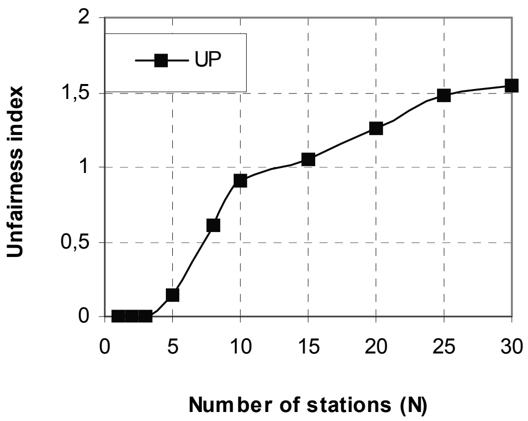

Fig. 4: Ratio σ_{up}/R_{up} for upstream and downstream connections ("wired host scenario")

Another possible solution [4] (denoted in the following as *Lossless* rate control) aims at controlling upstream and downstream rates so that no loss occurs in the downlink buffer. This is done by suitably modifying the window size advertised in TCP packets (such modification happening in the Access Point). However, we argue that, from an implementation point of view, this solution adds complexity since the device implementing this solution must operate on a packet-by-packet basis and parse TCP headers, in order to modify the receiver advertised window. Moreover, it is also necessary to estimate, in real-time, the number of TCP flows crossing such device.

In addition, as we will show in Section III.1, this solution may have worse performance in terms of throughput than our solution in the "remote scenario".

III. RATE CONTROL BASED SOLUTION

We envisage the use of a Limiter based Rate Control, to be implemented within the Access Point or in a router of the access network. In addition, we require that this mechanism can operate without any modifications in actual Wi-Fi mobile stations, based on the DCF defined in the 802.11 standard.

Our approach purposely introduces packet losses that stimulate the TCP congestion control mechanisms. The rate-limiter operates at the IP level, before the AP uplink buffer (see Fig. 5). Packets are dropped by the rate limiter with the aim of indirectly controlling the rate of uplink flows via the TCP congestion control. The rate limiter is a token bucket filter characterized by two parameters: 1) the rate of generating tokens into the bucket, R (expressed in Mb/s), and 2) the bucket size B_{bucket} (expressed in Mbit). The TBF generates tokens at rate R and puts them in the bucket. The rate limiter (and thus the AP) forwards arriving uplink packets only if there are tokens available in the bucket, otherwise uplink packets are dropped. Thus, the token bucket operates only as a dropper, i.e., it does not try to reshape non conforming packets and it does not need to queue packets. This makes its practical implementation very simple.

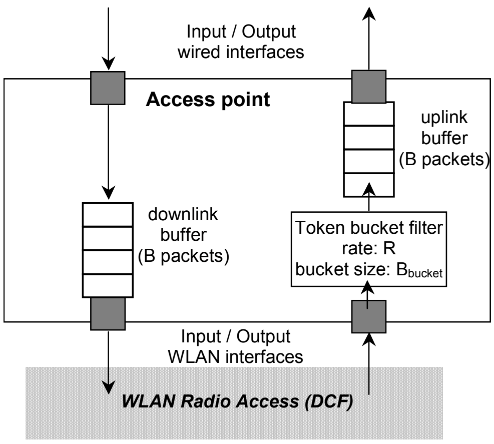

Fig. 5: Rate control solution based on a rate-limiter

In order to chose the right (for the performance point of view) values of R and B_{bucket} parameters we performed a simulation analysis; as a results we set R=2.3 Mb/s and B_{bucket}=200 packets (of 1500 bytes). For more details see [6].

We recall that our aim is to avoid starvation and to provide a fair access possibility to all applications. By using the TBF parameters given above, we evaluated by means of simulations the performance of the WLAN with and without the rate limiter. The results are given in Fig. 6, which shows the total upstream and the total downstream throughput, comparing the performance of the system without rate control to our rate control solution. The unfairness index in the same two cases is shown in Fig. 7. We stress that we obtained the important result of avoiding the so-called critical unfairness, observed in the "no rate control case".

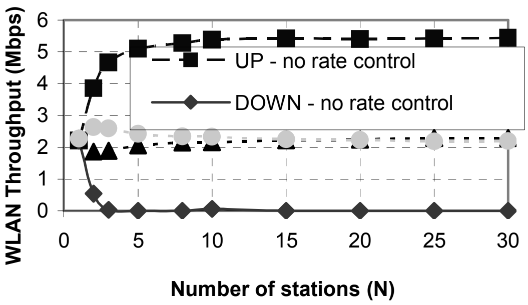

Fig. 6: Upstream and downstream throughput

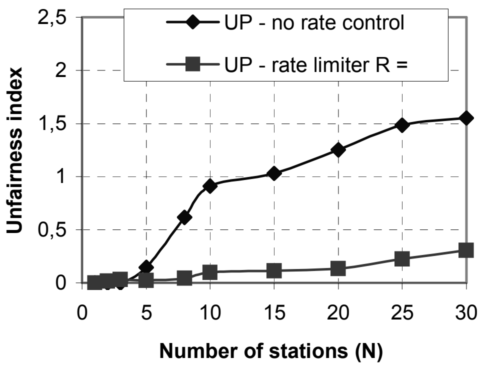

Fig. 7: Upstream fairness

III.1 Validation of the Rate Limiter for Different RTT

If the server is connected to the WLAN via an high speed Local Area Network, the round trip time is not an issue. In this case we can either enlarge the size of the AP downlink buffer up to B_{noloss} or adopt the solution proposed in [4] (neglecting the implementation complexity point of view). In both cases we obtain very good fairness and throughput performance. Instead, if we have to deal with large values of the Round Trip Time, our proposed rate limiter solution may work better then the *Lossless* rate control. As a matter of fact the case of large RTT is left as an open issue in [4].

Thus, we consider now the "remote scenario" (see the right part of Fig. 1) and terminate the TCP connections on a remote host in the fixed network, considering different values of the RTT of the link between the Access Point and the wired host. Fig. 9 shows the total, upstream and downstream throughput as a function of the number of stations, for an RTT value of 400 ms, comparing two solutions: the *Lossless* rate control proposed in [4] and our rate limiter. For an RTT of 200 ms (reported in Fig. 8) the results are still comparable with the scenario where the wired host is locally connected to the Access Point. The only difference appears when the number of stations is high (N=30): the lossless rate control is not able to reach the maximum throughput. For an RTT of 400 ms, the performance of the *Lossless* rate control is definitely worse, as it never reaches the maximum WLAN throughput. This is due to the limitation on the TCP Congestion Windows of the TCP connections.

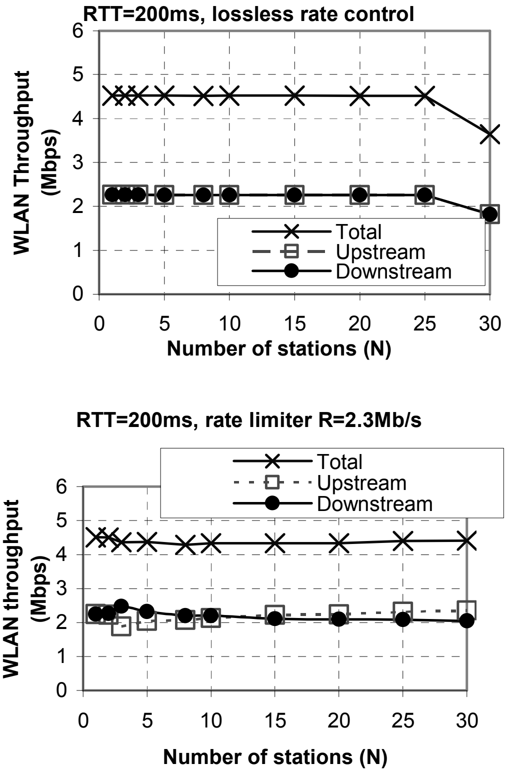

Fig. 8: Throughput versus N in the "remote wired host" scenario (RTT=200); comparison between lossless rate control and rate limiter

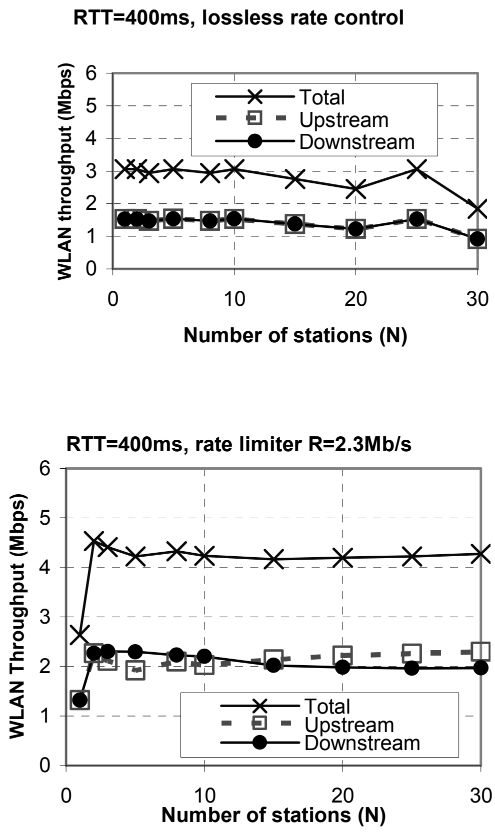

Fig. 9: Throughput versus N in the "remote wired host" scenario (RTT=400); comparison between lossless rate control and rate limiter

The well know throughput limit of the window based protocols (Throughput≤W/RTT), where W is the window size, comes into play and reduces the achievable total throughput. On the other hand, the rate limiter solution is not affected by the increase in the Round Trip Time, both in terms of the total throughput and in terms of upstream/downstream fairness. Fig. 10 provides another insight on the RTT problem, as it reports the total throughput versus RTT for the "no rate control", lossless rate control, and the rate limiter solution for a fixed number of connections (N=15). The throughput in case of lossless rate control starts to decrease from RTT≅250 ms.

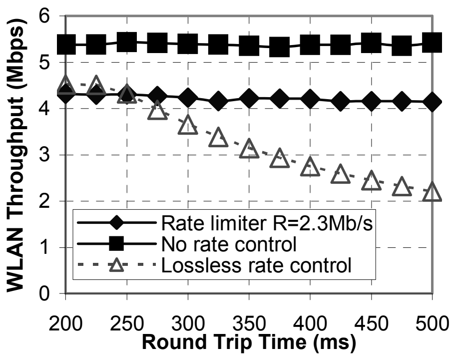

Fig. 10: Total throughput vs. RTT, N=15 ("remote wired host" scenario)

IV. CONCLUSION

In this paper, we addressed fairness issues in a wireless access network based on the IEEE 802.11b standard, operating in DCF mode at 11 Mbps. We have shown that without suitable countermeasures, connections use the available capacity in a very unfair way. We also identified "critical unfairness" that corresponds to complete starvation of TCP connections. We proposed a solution based on a "rate limiter", operating on the uplink traffic. The rate limiter is implemented by means of a token bucket filter and it indirectly controls the aggregate rate of upstream TCP connections by relying upon the TCP congestion control mechanisms. Our proposed rate limiter mechanism avoids critical starvation in all considered scenarios and is independent of RTTs.

REFERENCES

[1] S. Shenker, J. Wroclawski, "Network Element Service Specification Template", RFC 2216, September 1997
[2] The Network Simulator - ns-2, http://www.isi.edu/nsnam/ns/
[3] W. Stevens, "TCP Congestion Control", RFC 2001
[4] S. Pilosof, R. Ramjee, Y. Shavitt, P. Sinha, "Understanding TCP fairness over Wireless LAN", IEEE INFOCOM 2003, March 30-April 3, 2003, San Francisco, USA.
[5] H. Balakrishnan, V. N. Padmanabhan, and R. H. Katz, "The Effects of Asymmetry on TCP Performance", ACM Mobile Networks and Applications (MONET) Journal, Vol. 4, No. 3, 1999
[6] N. Blefari-Melazzi, A. Detti, I. Habib, A. Ordine, S. Salsano: "TCP Fairness Issues in IEEE 802.11 Networks: Problem Analysis and solutions Based on Rate Control", University of Roma Tor Vergata, technical report. Available in: ftp://TW05:TW05@160.80.81.106/tw05/TCP-fairness-long.pdf.