Evaluation of alternative backoff schemes for 802.11

Bachelorarbeit

zur Erlangung des akademischen Grades
Bachelor of Science (B. Sc.)

Humboldt-Universität zu Berlin
Mathematisch-Naturwissenschaftliche Fakultät
Institut für Informatik

eingereicht von: Lange, Frank
geboren am: 08. August 1989
in: Berlin

Betreuer: Dipl-Inf. Robert Sombrutzki

Gutachter: Prof. Dr. Jens-Peter Redlich
Prof. Dr. Joachim Fischer

eingereicht am: ...... verteidigt am: ......

Abstract

Opposed to wired communication, devices forming a wireless network are all using the propagation medium as a shared and limited resource. Therefore access to the medium needs to be organized. The IEEE 802.11 wireless communication standard therefore defines a *Distributed Coordination Function* (DCF) and its behavior can be summarized as *listen before talk.* Before a node transmits, he senses the shared medium to assure he is not interfering with an already ongoing transmission. If he senses the medium to be idle, he transmits. If however two nodes transmit at the same time, there is a chance of their packets interfering at the receiver. This is called a packet *collision* because in most cases the receiver is not able to decode any information from either packet. Collisions can occur because the involved transmitters cannot detect each other, which is known as the *hidden node* problem and 802.11 defines a special RTS/CTS mechanism to handle such scenarios. If however the transmitters are all in receiving range of each other the likely occuring collision is called *in-range collision.* To avoid such scenarios a *backoff scheme* is deployed which is a vital part of the *Medium Access Control* (MAC) layer.

Using the backoff scheme deployed by 802.11 each transmitter picks a random countdown. The one who first reaches zero gets access to the medium. If the competing transmitters pick large backoff countdowns to reduce their chance of collision they risk wasting precious medium idle time. If they all pick small countdowns the chance of collision increases and medium is wasted since it is polluted with corrupted transmissions and retransmissions. Additionally, since 802.11 defines an upper and lower bound for the possible backoff values but the node density in wireless networks is steadily increasing, alternative backoff schemes need to be deployed that can handle an increased number of nodes competing for the medium.

In this work three existing alternative backoff schemes are evaluated by means of network simulation. Furthermore two variations of a new backoff scheme approach are introduced and evaluated which, in contrast to the other backoff schemes, only use live and ad hoc information for determining the optimal per packet backoff.

Keywords: backoff scheme, wireless networks, 802.11

Contents

1 Introduction

After establishing itself as a "here-to-stay" technology in the last decade, wireless local area networks got an additional popularity boost by the rise of smartphones and tablets. Annually smartphone and tablet sales are expected to surpass the 1 billion mark in 2014 for the first time [1], which makes it save to assume that wireless network devices are almost everywhere.

This is of special interest, because these mobile devices are a perfect fit for the general use case of having a wired backbone network including wireless access Points (APs). These APs allow wireless devices to connect to the wired backbone network. If the backbone network is connected to the internet, users can browse the web whilst roaming the area and indeed this is how most company or campus wifi networks are set up. Furthermore wireless devices can be set up to communicate without any given network infrastructure by setting up their own ad hoc network. This renders them extremely versatile devices for both, expanding an already existing backbone network and forming a new network in a very ad hoc and flexible manner.

(a) Mensa Nord cafeteria [2]

(b) Central library [3]

Figure 1: Crowded spots at the Humboldt-Universität zu Berlin showing different demands of availability, capacity and roaming needs

With the climbing device count per area, also new challenges arise e.g. how to provide the same network experience for each client, independent of his bandwidth appetite. This remains a challenging task and is an ongoing research topic. In the context of wireless networks this seems additionally challenging because wireless devices communicate by exchanging signals without a direct physical connection between them,

Therefore wireless devices must use the surrounding medium as their shared communication channel as opposed to wired communication where said physical link exists and where ongoing transmissions can be transparent to other participating nodes in terms of throughput and link quality. Since the following work is focusing on wireless communication according to the

Institute of Electrical and Electronics Engineers (IEEE) standard 802.11 [4], electromagnetic radio waves in the *Radio Frequency* (RF) spectrum are being used to transmit signals.

Dictated by their physical characteristics these waves propagate the surrounding medium omnidirectional, reaching other wireless devices not intended as receivers or interfering with other waves. These underlying system properties are the reason, why the communication medium is often referred to as a *shared medium* and generally thought of as a *broadcast channel*. Needless to say these system properties get shoved into focus the more devices there are, since they are all competing for a limited and shared resource.

This is the source of several implications, one being that some sort of mechanism that determines when and how a device is allowed to transmit on the shared channel must exist. Because of that, the IEEE 802.11 standard features a whole sublayer devoted to *Medium Access Control* (MAC), the so called MAC-layer. One fraction of this MAC-layer deals with the question of how a device, also referred to as *node*, needs to behave if it wants to transmit using the shared medium. If the medium is unoccupied the node can start transmitting immediately but how should the node behave if it detects an already ongoing transmission? If he transmits at will and his transmission overlaps with an already ongoing transmission, then these two signals will interfere at one or more receivers which means there is a chance that neither of them will be correctly received at all. This results in precious medium capacity being wasted and these signals needing to be retransmitted, occupying even more medium capacity.

The 802.11 standard resolves this issue by defining a specific mechanism called a *backoff scheme*. The backoff scheme defined by the 802.11 standard has been analyzed in detail and different modifications have been proposed to improve their performance with regard to general performance criteria for wireless networks.

These modifications often focus on optimizing a single performance aspect of a wireless network e.g. throughput. It should be interesting to see how they perform when more performance aspects are required, e.g. throughput *and* fairness.

1.1 Aim of this work

The following work analyzes and evaluates alternative backoff schemes for wifi networks by means of simulation using the *Humboldt Wireless Lab* (HWL) [5] simulation framework based on the *NS-2* network simulator [6] and the *Click Modular Router* [7] software routing architecture. Additionally a new backoff scheme approach is proposed that utilizes information gathered from the hardware wifi chip and is benchmarked against other already proposed backoff schemes.

Since alternative backoff schemes often feature pre-applied simulation to predetermine the optimal backoff for any given situation, it will be interesting to see whether or not the proposed backoff scheme can compete while only using live and ad hoc information.

1.2 Structure of this work

First, in section 2, the basic mechanics relevant to backoff behavior of the IEEE 802.11 standard MAC-layer are introduced and explained. In section 3.1 possible weaknesses of the 802.11 standard backoff behavior are exposed and the general metric for evaluating any given backoff scheme performance is defined. In section 4, three already proposed alternative backoff schemes are being presented and explained. Additionally my own approach which yielded two slightly different backoff schemes is presented and the two backoff schemes are introduced. In section 5 all introduced backoff schemes are evaluated individually and in detail by means of network simulation and section 6 concludes the work.

2 802.11 basics

The IEEE 802 standard family is a collection of multiple specifications targeting the realization of a *Physical Layer* (PHY) and a MAC layer for ceveral classes of local area network devices. This means each specification targets the lower two layers of the *Open Systems Interconnection* (OSI) model [8, p.41-43]. Each of the IEEE 802 standards contains a MAC and a PHY component, with the MAC component regulating how to access the medium and the PHY component detailing how data is actually received and transmitted [9, p.21].

The specific standard this work focusses on is the 802.11 standard, which targets wireless devices. In its initial version released in 1997 it defines one MAC-layer and three PHY-layers, utilizing *Frequency Hopping Spread Spectrum* (FHSS), *Direct Sequence Spread Spectrum* (DSSS) and difuse infrared respectively as PHY-layer receive and transmit mechanisms. It utilizes the RF range around 2.4 GHz defined by the *Industrial, Scientific and Medical* (ISM) radio band [10].

Over the years the basic standard has been augmented, resulting in the 802.11a, 802.11b, 802.11g and 802.11n versions where additional PHY-layer mechanisms like *Orthogonal Devision Frequency Multiplexing* (OFDM) and an additional frequency range in the 5 GHz spectrum were introduced. The currently released 802.11 products at the time of writing are using the newest 802.11ac standard which defines a single link throughput of at least 500 *Megabit per second* (Mbit/s) [4].

2.1 WLAN structures

The 802.11 standard defines two basic ways of logically connecting wireless devices to form a WLAN: *infrastructure Mode* and *ad hoc mode.* In the infrastructure mode the nodes opperating in the constructed WLAN are seperated into two groups, APs and clients. The minimal requirement for constructing a WLAN using the infrastructure mode is having at least one single AP to which one or more clients can be *associated* with. In this infrastructure mode, shown in Fig.2, every data packet a client sends will first be send to the AP he associated with and the AP will then decide how to forward the packet any further. Perhaps the AP is connected to a wired backbone network which is connected to the internet. Then the AP can forward the client packet to the next wired router, thus allowing the client to communicate using the internet.

If however the client wants to communicate with another node in the same network he cannot simply transmit to his neighbouring node even if a direct link technically does exist. He has to address his neighbouring node logically as the intended receiver and send his data packets to his AP first, which then forwards the packets to the other client.

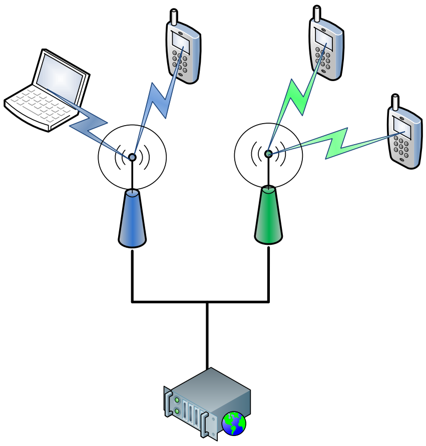

Figure 2: 802.11 Infrastructure scenario with two access points being connected to a wired backbone network with access to the internet

Wireless nodes opperating in ad hoc mode don't make that distinction between being a client or an AP, because in ad hoc mode every participating node acts as a client. This results in the fact, that every node can directly communicate with every other node in its receiving range. The downside of this approach is, that all nodes in the network must be in receiving range of each other. A single node can only directly receive or send packets and cannot forward them since he's not acting like an AP. At least that is, what the 802.11 standard defines for the lowest OSI layers.

This does not however limit the capabilities of any higher OSI layers, e.g. the network layer, where there still can be routing i.e. forwarding of packets to other nodes or intended receivers respectively.

Adding a routing component to the lower OSI layers however leads to wireless networks which are described as *multi hop ad hoc networks* or more often as *Mobile ad hoc Network*s (MANETs). In a MANET every participating node acts as a client and an *Access Point* (AP). This means there is no static underlying network infrastructure which is why MANETs are often characterized as being self-organized. This makes them extremely versatile and flexible which is why the most prominent use cases describe scenarios in which the underlying network infrastructure has broken down or never even existed, i.e. in a disaster area or in a war zone.

Such a wifi network structure implies ceveral constraints that may not exist otherwise, i.e. limited global knowledge about the whole network for each node, no centralized coordination, error-prone communication caused by intereference of other devices or just simply high mobility of devices.

Since there is no global coordinator, otherwise controlled access to the propagation medium is key, which fits perfectly with the MAC-layer defined in 802.11.

2.2 802.11 MAC

The 802.11 MAC-layer is targeting the *Data Link Layer* (DLL) or layer two of the OSI model and its task is to manage access to the propagation medium shared by all devices in a wifi network. Although 802.3 (Ethernet) of the IEEE 802 standards family already defines such a layer for wired communication, 802.11 needed to define its own for wireless communication because of the way 802.3 handles *collisions*.

A collision occurs when two transmitted signals interfere at the receiver. As a result the receiver cannot decode any useful information and the packets need to be retransmitted (preferably without interfering again). IEEE 802.3 tries to detect such collisions using the *Carrier Sense Multiple Acces with Collision Detection* (CSMA/CD) mechanism. With CSMA/CD an interfering signal can be detected at the transmitting side of the wire. Since using wired communication, one can assume the signal strength to be the same at the receiving end as well as at the transmitting end. With that additional information the sender can infer that if he could detect a collision at his side of the wire it also occured at the receiver [11, p.69-70].

Using wireless communication however renders this implicit knowledge unuseful because the signal strength at the transmitter can be significantly different from the received signal strength at the receiver which makes it impossible for the transmitter to infer whether a collision has occured at the receiver by just using his own signal strength as an indicator. This is why 802.11 defines a new mechanism called *Carrier Sense Multiple Acces with Collision Avoidance* (CSMA/CA). Rather than trying to detect collisions, 802.11 tries to avoid them in general.

It is important to note however, that if two packets do collide at the receiver, there is a chance that the *Signal to Nois and Interference Ratio* (SNIR) of one packet is higher than the other, which results in the packet with the higher SNIR still being received correctly. This effect is called *PHY-Layer capture.*

Furthermore the 802.11 standard defines two logical variants of transmissions, a *broadcast* and a *unicast* transmission. If the transmission is a broadcast transmission then there is no single intended receiver, since the whole idea behind a broadcast message, is to reach multiple receivers with a single message without targeting them all individually. This also means that there is no acknowledgement mechanism using broadcast transmission. This restricts backoff behavior because although a node wanting to trans-

mit a broadcast message but detecting the medium to be busy does pick a backoff countdown for his (initial) broadcast transmission, without *Acknowledgement*s (ACKs) there is no notion of retransmitting packets and adapting the backoff interval which could decrease the chance of collision for further broadcast transmissions.
In contrast to that, unicast transmissions only target a single receiver who will acknowledge any received data packet. This allows for the transmitter to realize when a packet did get lost, because the ACK is missing which makes him infer a collision. He can then retransmit the packet using the backoff mechanism to reduce the collision likelihood and can further adapt his backoff interval if he is not receiving an ACK for his retransmissions.

Although a retransmission for the transmitter, it polutes the medium like any other transmission and is therefore considered as such. This means that the same backoff mechanism applies, as with any other transmission. It is important to note however, that 802.11 defines different sizes of time intervals that can occur between any two data frames. With these so called *Inter Frame Spacing*s (IFSs) it is possible to give ACKs a higher precedence over any data frame. Therefore ACKs are not subject to any backoff mechanism and can be transmitted immediately by the receiver.

2.3 802.11 backoff mechanism

Similiar to 802.3, a node wanting to transmit first senses the medium to detect if there is an already ongoing transmission. This behavior can also be summarized as *listen before talk* and is key to 802.11's CSMA/CA. If the medium is sensed to be *idle* the node can start transmitting right away. If however the medium is sensed to be *busy* this means there is an already ongoing transmission taking place. In that case the node defers its transmission, he *backs off*.

If however multiple nodes want to transmit but defer their transmission because of an already ongoing transmission, the end of that transmission marks a synchronization point. If they all start transmitting when the medium is idle again, their packets are likely to collide. So to safely backoff its transmission a node chooses a random number from the whole number interval [0, *CW*) and counts it down to 0. Here, *CW* denotes the current *Contention Window* size and therefore a momentarily upper bound for the backoff interval. To count down the randomly picked number, the node continues sensing the medium. If he senses an idle slot he decrements his backoff counter. If on the other hand he senses a busy slot, he freezes his countdown until the next idle slot is encountered. When the backoff countdown finally reaches 0, the node starts to transmit. A *slot* in this context refers to an abstract time interval defined by 802.11. In the different 802.11 versions the actual duration of a slot ranges from 9 μs in the newer versions to up to 50 μs in original version from 1997.

If a collision occurs or the ACK packet gets lost due to other reasons, e.g. a weak signal, the transmitting node assumes a collision occured. This means that, at some point during his countdown process either right at the beginning or along the way, some other node picked a backoff that resulted in both backoffs reaching zero at the same time. So to avoid another collision, both double their backoff interval, thereby reducing the chance of further collisions. If a node finally receives an ACK for a retransmission, he *resets* his backoff interval to the starting size specified by 802.11.

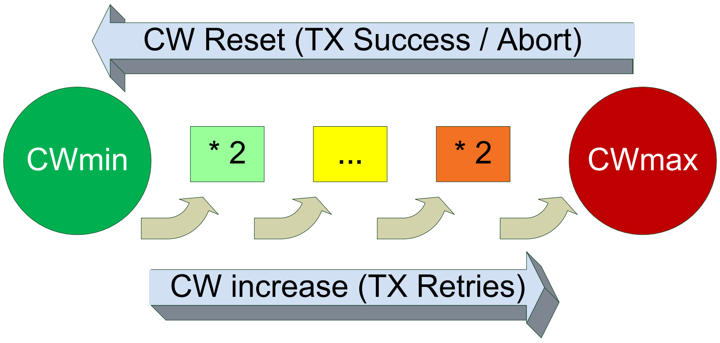

Figure 3: Basic 802.11 dynamic contention window size

This means that over the course of an unicast transmission the backoff interval changes dynamically as shown in Fig.3. For that matter 802.11 defines specific boundaries for the backoff interval. For the initial transmission of a packet, a node choses his backoff from $[0,\ CW)$ where CW takes the value of CW_{min} which is the 802.11 defined constant of 32. For any retransmissions of that packet he doubles CW until it reaches a maximum value of CW_{max} which is also a constant defined by 802.11 and amounts to 1024, utilizing powers of two throughout the entire backoff scheme. If CW reached CW_{max} it keeps that value for any further retransmissions until either a successful transmission occurs and the CW gets reset to CW_{min} or the maximum retry count is reached and the overall transmission is aborted. The maximum number of retries per packet amounts to 7 and is also a specific constant defined by 802.11 [4].

In summary, the crucial MAC mechanisms deployed by 802.11 are *listen before talk* CSMA/CA instead of CSMA/CD and a so called *Binary Exponential Backoff* (BEB) scheme based on backoff values of powers of two. Since in a *Wireless Local Area Network* (WLAN) there is no centralized management facility, every node has to deduce his right to access the medium on its own, using just the described MAC mechanisms 802.11 offers. This is why the composition consisting of CSMA/CA and the BEB is termed *Distributed Coordination Function* (DCF) by 802.11.

There is one other coordination function defined by 802.11 on top of the DCF, called *Point Coordination Function* (PCF) which only works when using the infrastructure mode setup because APs are used as point coordinators, providing a centralized coordination functionality [9, p.39]. Unfortunately the PCF is virtually never used in practice due to the circumstance that APs belonging to one wifi network can not regulate any traffic of near, perhaps even overlapping wifi networks [8, p.304].

3 Backoff analysis

So far the basic backoff mechanism of 802.11 has been described. This standard mechanism is the basis of the following work. It has been thoroughly tested since the standard was released and several minor and major modifications have been proposed. In the following sections the problems of the 802.11 backoff mechanism shall be highlighted, related work in the form of a selected set of proposed modifications shall be introduced, analyzed and evaluated by means of network simulation.

3.1 Backoff scheme performance

With the growing bandwith appetite of end user devices, optimizing the *sum throughput* of a backoff scheme is essential to improving the overall network performance. Optimizing the summarized throughput alone however, could yield unsatisfactory results. Imagine a network where two nodes have far more throughput possibilities than all the other nodes, e.g. by using a more complex *Modulation Coding Scheme* (MCS). Maximizing only the overall summarized throughput would consist of decreasing the backoff intervall of these two nodes, while strongly increasing the backoff intervals of all the other nodes, thus resulting in only these two nodes exchanging traffic, while all the other (slower) nodes are desperately waiting for a chance to transmit. This would result in a maximized summarized throughput of the overall network but the single node throughputs would differ dramatically. Which is why *fairness* is also crucial for evaluating the performance of a backoff scheme. Because this work only analyzes scenarios in which all nodes use the same MCS and packet size, only their backoff interval and therefore the amount of medium access determine the overall network fairness. This means a *fair* setup would include equal backoff intervals for all participating nodes, so that all nodes get an equal chance of medium access. If, however, one would only optimize for (medium access) fairness, one could think of simply using an enormous backoff interval to minimize the chance of collisions. This way every node would have an equal chance of being granted access to the medium with virtually no fear of collision, resulting in nodes picking way to high backoffs and spending most of their time decrementing their backoff counter rather then transmitting, therefore wasting precious medium idle time and contradicting with the throughput optimization criteria.

With that being said, finding the sweet spot between throughput and fairness is key to any given backoff scheme performance and therefore the main focus when evaluating any given backoff scheme.

3.2 802.11 backoff weaknesses

As described earlier, the standard 802.11 backoff scheme uses fix constants as lower and upper bounds for the backoff intervall from which a node chooses its backoff. This means that no matter how big the network or how much neighbours a node has, he will choose his backoff from [0, CW_{min}) for his first deferred transmission, with CW_{min} set to 32. This means that even when there are only a few nodes, e.g. 2 or 3, they are choosing way to high backoffs and therefore are wasting medium idle time. Likewise, if the maximum backoff each node can possibly choose is 1024 then for all networks that feature single collision domains with 10 or more contending nodes, there is a high chance of two nodes choosing the same backoff. Being able to choose a backoff higher than 1024, if the scenario requires it, could be beneficial to the overall network performance. In essence, choosing these interval limits accordingly could yield performance boosts, when done wisely.

A second possible weakness dictated by 802.11 is the contention window reset after a successful transmission. If a node had previously increased his contention window size due to the backoff mechanism he will blindly reset his contention window size to the minimum of CW_{min} after each successful transmission. This means that for each transmission following a successful one, the node will pick a relatively small backoff, thereby risking new unnecessary collisions because it does not take into account why he previously had to increase his contention window size in the first place. This can lead to scenarios where nodes reset their contention window although they and their neighbours would actually benefit from picking a genuinely higher backoff, e.g. in highly intermeshed networks or just dense parts of a network.

4 Alternative backoff schemes

In the following chapter different alternative backoff schemes shall be introduced and their approaches shall be explained and discussed at an abstract level. After the proposed backoff schemes that can be found in the literature are being discussed, my proposed approach of a backoff scheme shall be introduced so that in the following chapters it can be evaluated in detail and compared to the most promising contenders from related work.

4.1 Related work

To resolve the known issues of the standard 802.11 backoff scheme, multiple alternative backoff schemes have been proposed.

In [12] the authors propose a backoff scheme called *Multiplicative Increase Linear Decrease* (MILD) where, in case of a transmission failure, the transmitting node still doubles its *Contention Window* (CW) but on a successful transmission does not reset his current CW back to CW_{min} but rather decreases it linearly by subtracting a fix constant. This is supposed to cure 802.11's nervousness in case of a successful transmission and resembles a more pessimistic approach because nodes increase their contention window quite generous but only slowly decrease it in case of a successful transmission and therefore carefully approach the optimal contention window size from higher backoff values.

In [13] the authors propose a backoff scheme called *Pessimistic Linear Exponential Backoff* (PLEB) where in case of a transmission failure the transmitting node starts increasing his CW by doubling it but after a certain threshold switches to only a linear increase. On a successful transmission the node resets his CW to CW_{min} just like 802.11.

In [14] the authors propose a fibonacci-inspired backoff scheme where in case of a transmission failure the n-th retry picks a backoff according to the n-th fibonacci number.

All these proposed schemes have in common that they are willing to break with the binary exponential backoff defined by 802.11 and which is implemented by off the shelf hardware. This implies that the only way to evaluate the performance of said schemes is to use network simulation and even then the used network simulator needs to be changed quite a lot since it is implementing the 802.11 MAC-Layer using BEB right out of the box.

Because of those evaluation limitations the following work focusses on backoff schemes which stick to the basic constraints the hardware and 802.11 enforce. That means that all of the following backoff schemes stick to a BEB based on powers of two and try to achieve greater performance by finding new ways of choosing CW_{min}, CW_{max} and when and why to change from CW_i to CW_{i+1}.

In that regard the authors of [15] propose a backoff scheme, where 802.11's BEB is still being used but the value for CW_{min} is determined by the formula:

$$CW_{min} := 8.5 * N - 5$$

where N depicts the number of neighbours of the transmitting node. The formula was obtained by means of simulation where every value for CW_{min} was tested against any number of neighbouring nodes and is supposed to make it possible for nodes to reflect their congestion situation in the backoff they are choosing, without being bound to fix constants as with 802.11. This scheme will be further referenced as the *Neighbours* (Nbs) scheme.

In [16] a backoff scheme is proposed where the transmitting node uses a lookup table to determine which CW_{min} to use for his current transmission. The table was predetermined by means of simulation and is the same for all nodes and immutable. To obtain the table every combination of selected rate, packet size and number of neighbouring nodes was simulated against any CW_{min} thus making it a three dimensional lookup to choose the approriate CW_{min}. For the rest of this work, this scheme will be refereced as the *maximum Throughput* (max. TP) scheme.

Both schemes have in common that pre-applied simulations are being used to determine the "optimal" backoff, which means the backoff that yields the best results regarding a balance between throughput and fairness. All a node has to do live, to pick a backoff, is to either simply look it up in a table or compute it using a compact formular.

It is important to note however, that although both schemes strive to reduce the amount of work a node has to do, to choose his backoff in a live scenario, by heavily using pre-applied simulation, they both still need information that can only be gathered live because they both rely on the fact that each node knows how many neighbouring nodes he has. Detecting how many neighbours a node has at a current point in time is a weak spot that can yield many discussions because one needs to define which node under which circumstances gets recognized as a neighbour. This will again be thorughly discussed in the chapter 4 where each scheme gets evaluated individually.

In [17] the authors propose a different approach which in the following work will be referenced as the *Learning* scheme. Instead of using pre-applied simulation to prepare the backoffs to choose from, they view CW_{min} as an evolving value that – over time – reaches the "optimal" value and then oscillates around it. The key idea is that the last used CW_{min} is remembered for the next packet. Therefore one could say that nodes "learn" what the optimal backoff for their current situation is and remember how they got there by at least remembering their last transmission. This is supposed to cure 802.11's unforgiving short term memory which blindly resets its CW size to CW_{min} after every successful transmission and for every new packet.. To achieve that, the authors propose, that for an unsuccessful transmission the CW size still gets doubled but for a successful transmission, instead of being reset, it only gets halved. To see how this yields oscillation, imagine a node which has chosen a CW size of 64 for his current packet which means he randomly chooses his backoff in the intervall $[0, 63]$. Further imagine that this transmission failes and he doubles his CW size to 128. If 128 turns out to be closer to the "optimal" backoff value this next transmission is likely to succeed, which would then trigger a halved CW size of 64 again which, as we have already seen, would likely result in an unsuccesful transmission, triggering a doubling of the CW to 128 again and so forth. This oscillation is supposed to happen when CW_{min} has reached some value close to the optimal backoff and is also supposed to yield better results in situations where the theoretical optimal backoff would *not* be a power of two, e.g. 82 or 100 in the example above.

It is important to note however, that when using this backoff scheme CW_{min} can only change step by step, doubling or halving it with every new transmission. The previously described schemes *Nbs* and *max. TP* both did not depend on previously chosen backoffs when picking a backoff for a new packet. If a node using *Nbs* or *max. TP* had just used a CW_{min} of 64 for his last packet, he could now choose a CW_{min} of 1024 for his next packet without any hesitation *if* his backoff table or formula would tell him to do so. When using the *Learning* scheme, even if a node miraculously knew, that 1024 would be the best backoff for his current situation, if his current CW_{min} is 64, he can only gradually double it with every transmission, which could take a considerable amount of time when the amount of his neighbours has reached a point, where it naturally occurs that he can't access the medium for even a second or two because of the medium congestions that comes with an increased number of neighbouring nodes.

This means that if, for some reason, some nodes "leave" the network, because their battery died, they simply moved to a different position or because they don't want to transmit, then all it takes for the *Nbs* and *max. TP* scheme is to realize that the number of neighbours has changed. This would result in a different table lookup for example and would immediately result in picking a new and hopefully more appropriate backoff for this new situation. The *Learning* scheme however, would now again need time "learn" what the new approriate backoff is.

In favour of the *Learning* scheme one should note, that unlike the previously introduced schemes *Nbs* and *max. TP* it doesn't need any statistics. The only information it relies on, is how many retries were needed to transmit the current packet. For every retry, the CW gets doubled, for every success it gets halved, simple as that.

4.2 Proposed approaches

To extend the roster of backoff schemes I would like to propose another, different approach to choosing the approriate backoff. In [18, p.49–53] Thomas Hühn used, among other things, the information contained in the hardware registers of the Atheros wifi chips to improve the rate selection and power control of wireless mesh nodes. Inspired by the use of those registers one could imagine a backoff scheme using these registers to infer which backoff might be approriate for the current situation. Although the Atheros wifi chips feature several registers, as can be seen in [19], only four of them shall be relevant for the proposed backoff approach. The "AR5K_PROFCNT_CYCLE" register gets periodically incremented for *every* cycle the wifi chip ever does and will for the rest of this work be referenced as the *cycle count* register. The AR5K_PROFCNT_TX and AR5K_PROFCNT_RX register are being incremented for every cycle the wifi chip spends sending or receiving and will therefore be referenced as the *tx* register and *rx* register respectively. The AR5K_PROFCNT_RXCLR register gets incremented for every cycle the chip could detect the medium as "busy" and will therefore further be referenced as the *busy* register. This means that whenever the chip itself transmits or receives, the *busy* register gets incremented, since the medium is "busy" at that time. However this also means, that whenever the wifi chip itself is not transmitting or receiving and the *busy* register still gets incremented, then the medium is occupied by some other signal source. This could either be some other node within receiving or just carrier sensing range but this could also indicate an interfering signal source which could indicate *Adjacent Channel Interference* (ACI) or even an external non-wifi device like a microwave or a baby monitor.

In summary the really interesting registers are the *rx*, *tx* and *busy* registers because the cycle count register is only used as a reference to calculate what percentage of time the wifi chip spent in which mode. This can be achieved by reading and reseting the registers periodically, e.g every second, and therefore *busy* divided by *cycle count* depicts what percentage of the last second the wifi chip spend detecting *any* signal on the medium, either his own or someone else's. The interesting question now is, how to use these registers as useful information for choosing an approriate CW_{min}.

The first approach centers around the idea of *only* using the *busy* register and will therefore further be referenced as the *Busy aware* scheme. Since the *busy* register of a node not only shows the node's own medium access but also every other medium access that reaches this node, it's a perfect fit to indicate the actual channel load at this node. So to utilize this as a backoff scheme, a specific target channel load is given to every node which is the same for all nodes. Each node tries to reach the specifed target channel load by calculating the channel load he observed using the *busy* and *cycle count* register as depicted above. If the detected channel load is below the provided target, he will halve his CW_{min}. By decreasing his CW_{min} he reduces the time he has to wait until he can access the medium. Since the backoff is only decremented while the medium is detected as idle, less medium idle time is spend decrementing and therefore the channel load is increased. Which is acceptable because he had just detected that his received channel load was below what was provided as a target channel load. If he detects his received channel load to be above the provided target load, he will double his CW_{min}, thereby increasing the time he has to wait for medium access which decreases the channel load. Again this is exactly what is supposed to happen since his received channel load was above the provided target.

This approach borrows heavily the idea of interpreting CW_{min} as an steadily evolving value that was proposed by the *Learning* scheme. However, it extends this idea because now, in a general sense, a node can decide on how to change his backoff based on two seperate values: what he wants and what he actually got. Providing these two values makes it now possible for any backoff scheme to provide behavioral mechanisms to deal with the possible differences of these values. If, for example, a node detects that he has gotten way less than he actually should have got it is now up to the backoff scheme to determine the follow up behavior. In case of the *Busy aware* scheme the want/get is represented as the channel load, given by a target and the currently measured channel load.

In case the node actually measures less channel load than the given target channel load The *Busy aware* scheme decides to be more "aggressive" by decreasing the backoff which represents the increased demand of medium access of a node. As with the *Learning* scheme, the *Busy aware* scheme is also independent of any pre-applied simulation. It is also independent of the number of neighbouring nodes but it does however need the live statistics of register contents.

Another approach of utilizing said hardware registers centers around using the *tx* and *cycle count* register which is why, as with the *Busy aware* scheme, this backoff scheme will further be referenced as the *Tx aware* scheme. Borrowing again the idea of an evolving CW_{min} and balancing expectation with reality, a node using this scheme needs the number of his neighbouring nodes to infer what percentage of the medium he is entitled to. If a node detects that he has n neighbouring nodes, he can infer that, by including himself, $n+1$ nodes are currently sharing the medium. With that information and fairness in mind he can further deduce that he is entitled to

$$mediumshare := \frac{100}{n+1}\%$$

of the medium where n depicts the amount of neighbouring nodes. So by dividing *tx* by *cycle count* he gets the direct feedback of what percentage of the last time interval he actually did spend transmitting. Again, any backoff scheme now could decide how to deal with any mismatch of these values.

A node using the *Tx aware* scheme decides that if the detected transmit percentage is lower than the percentage each node is entitled to if they were to share the medium equally, it will halve his current CW_{min}, thereby increasing his chance of medium access. Which is acceptable since either the node spent too much time decrementing his backoff counter, wasting precious medium idle time or some other node did use an unfair amount of the shared medium. In both cases decreasing the CW_{min} for future packets represents the demand of medium share of this node and enforces fairness. If however the node detects that his transmit percentage is above what he is entitled to, he will double his current CW_{min} thereby reducing or delaying his chance of medium access and guaranteeing being fair to his neighbouring nodes. In contrast to the *Learning* and *Busy Aware* scheme but as well as the *Nbs* and *max. TP* scheme this scheme needs the correct number of neighbours to determine the target transmit percentage.

4.3 Scheme Summary

Five backoff schemes have been introduced. Two of which represent a new approach of realizing a backoff scheme, namely *Busy Aware* and *Tx Aware.* Both use certain hardware registers of the Atheros wifi chip to gather information on how to choose an approriate CW_{min}. In the following section these two schemes shall be evaluated and benchmarked against the three backoff schemes that were also introduced, namely the *Neighbours*, *maximum Throughput* and *Learning* scheme.

Even without further performance evaluation of said schemes two basic approaches of composing an alternative backoff scheme for 802.11 can already be recognized. One way includes using pre-applied simulation to test a certain range of transmission parameter settings to determine a CW_{min} that yields both, promising throughput while still being reasonibly fair in the process. Both, the *Nbs* and *max. TP* scheme use this approach but despite the simulation overhead beforehand, both still rely on live statistics to utilize the “knowledge” advantage they both got from the pre-applied simulations, namely the current number of neighbouring nodes. This introduces a possible disadvantage since retrieving the correct number of neighbouring nodes can be error prone, e.g. when not all neighbouring nodes want or get to send during the (possibly periodic) measurement phase.

The other approach does not rely on simulation but rather bets on finding or rather “learning” the best backoff over time, relying only on live statistics to do so. This eliminates the overhead of simulating every possible parameter combination via simulation but risks a time overhead which is needed to first learn *any* sensible backoff setting for the current situation. This time overhead is needed again if the situation, to which the backoff has just been adjusted to, changes suddenly. Furthermore the performance of these ad hoc schemes is bound to the available statistic mechanism, e.g. readable register contents. If the execution environment, e.g. hardware, simulator, does not support the needed mechanism, any backoff scheme's performance based on that exact feature is bound to drop.

It should be interesting to see how these two approaches perform when being evaluated using network simulation in the following chapter.

5 Evaluation

In the following chapter the previously introduced backoff schemes *Neighbours*, *maximum Throughput*, *Learning*, *Busy Aware* and *Tx Aware* shall be evaluated and their individual advantages and disadvantages shall be illustrated and discussed.

5.1 Implementation and simulation setup

To evaluate the introduced backoff schemes, network simulation was being used which consisted of the network simulation framework that is currently being used at the HWL [5]. This simulation framework makes use of the *NS-2 network simulator* [6] and the *Click Modular Router* framework [20] [7]. To describe a simulation in this setup, the *Click* language is being used to describe the arrangement of software routing elements as a graph. It allows to describe the flow of a packet through the *Click* element stack by "routing" it from one *Click* element to the next one. All the introduced backoff schemes were implemented as such *Click* elements. To allow access to the described hardware registers needed by the *Busy Aware* and *Tx Aware* scheme the NS-2 network simulator was extended to simulate said registers and allow access to read and reset them accordingly.

To stresstest each scheme, the main setup when simulating consisted of a single collision domain comparable to Fig.4. This means that unless statet otherwise, there was only one receiving node and the number of transmitting nodes changed from simulation to simulation but was constant for a single simulation run.

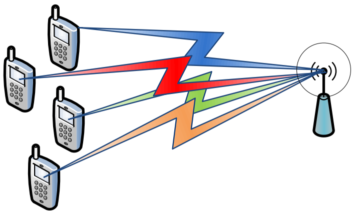

Figure 4: Single Collision Domain scenario: multiple devices being associated with a single access point

For every number of transmitting nodes, the simulation was repeated 20 times with a different seed for the *Pseudo Random Number Generator* (PRNG) used by NS-2. The single collision domain scenario supposed to represent the worst case scenario for any given backoff scheme: a group of transmitting nodes being positioned very dense and nearby each other. All transmitting nodes are also not just in carrier sensing but of course in full receiving range of each other and are therefore all competing for the same resource: the shared medium. Every transmitter is trying to transmit

packets with a *MAC Service Data Unit* (MSDU) size of 1500 bytes and is trying to send every 12 ms using a data rate of 1 Mbit/s which is the lowest data rate 802.11 has to offer but is also considered to be the most stable MCS. This means every transmitter alone is trying to completely saturate the communication channel, which is often referred to as sending *backlogged* in the literature.
To accommodate for the *PHY-Layer Capture* effect described in section 2.2, *Shadow-Fading* is used as part of the path-loss model used by the NS-2 simulator. By using *Shadow-Fading* the signal strengths of packets as received by the receiver are determined by a normally distributed random process with a given mean and standard deviation (in *dB*). Depending on the standard deviation setting, this can lead to packets, transmitted by two nodes with the same distance to the receiving node, being received with different signal strengths by the receiver. This effect would not have occured when using the *Two-Ray Ground* radio model for transmitting nodes with the same distance to the receiver.

To summarize all simulation parameter, an overview can be found in Table 1.

Parameter	**Setting**
RX Nodes	1
TX Nodes	2–30
Repetitions per no. TX	20
Simulation Duration	60 seconds
MSDU Size	1500 Byte
MCS	1 Mbit/s
Transmit duration per packet	12.3 ms
Send interval	12 ms (backlog)
Path-Loss	Shadowing
Channel Stats Interval	1 second

Table 1: Simulation settings overview

5.2 Pre-applied simulation schemes

Keeping the same order as in section 4 the first schemes to be evaluated are the *Neighbours (Nbs)* and *max.TP* scheme. They both use pre-applied simulation to determine the most promising CW_{min} before the actual live scenario.

5.2.1 Neighbours

This scheme used pre-applied simulation testing all feasible CW_{min} against any number of neighbours which resulted in a compact formula that now, in a live scenario, determines the CW_{min} using only the number of neighbouring nodes. Fig.5 shows the mean of the overall number of packets that got transmitted which shows the overall summarized throughput when converted into kbit/s. Also the standard error of each number of transmitters is plotted as errorbars. This shows, that the scheme with its formula deliveres the promised results.

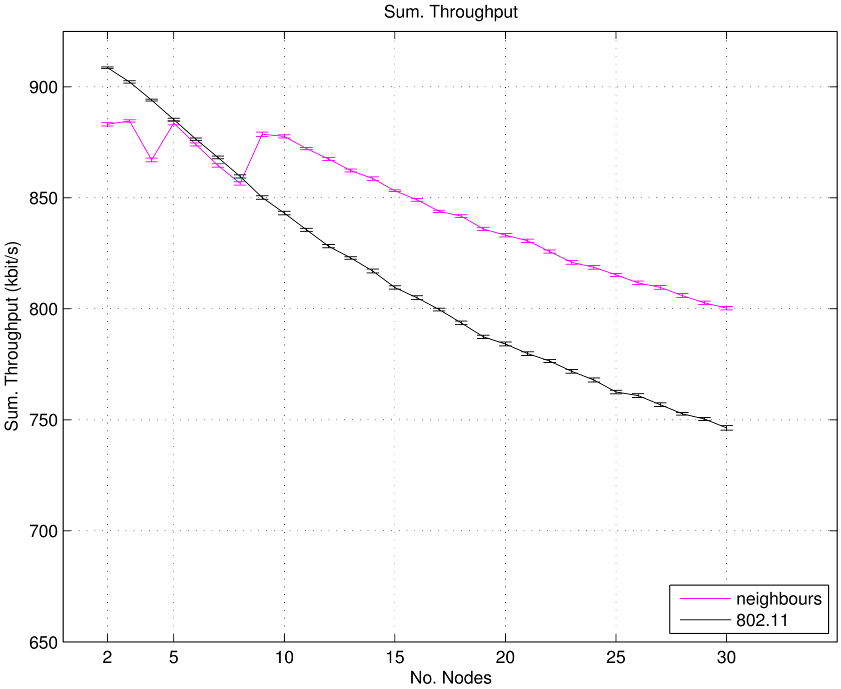

Figure 5: Mean of the summarized throughput using the *Nbs* scheme. Standard error plotted as errorbars for each set of simulations.

The drops in throughput are the symptom of using only powers of two as CW_{min}. Although the formula gives different values for six, seven or eight transmitting nodes, they all pick the same CW_{min} since that is the approriate power of two. Thus resulting in dropping throughput, since the number of competing nodes increases, but so does their chance of collision as well, since CW_{min} stays the same.

As stated in section 3.1 throughput alone without considering fairness however, is not sufficient to evaluate any backoff scheme performance. So to measure fairness, *Jain's Fairness Index* [21] is being used which is calculated by for any given set of throughput streams $x_1, .., x_n$ by using:

$$J(x_1, x_2, .., x_n) = \frac{(\sum_{i=1}^{n} x_i)^2}{n * \sum_{i=1}^{n} x_i^2}$$

This delivers results ranging from $1/n$ in the worst case to 1 in the best case and is at a maximum when all nodes were able to send an equal amount of packets.

Fig.6 shows that, although using a different packet size and MCS and not being explicitly optimized for fairness in their initial simulation presented in their publication, the *Nbs* scheme surpasses 802.11 for ten and more transmitters in the same collision domain when using the simulation settings depicted above in Table 1.

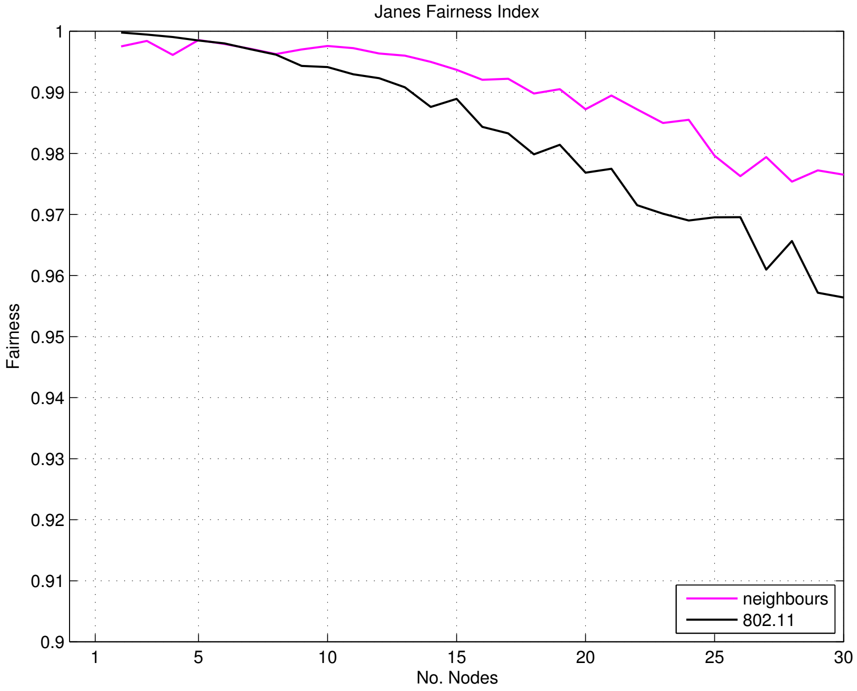

Figure 6: Janes Fairness Index of the Nbs scheme

5.2.2 Maximum throughput

Next is the *max. TP* scheme which also uses pre-applied simulation but rather than a closed formula represents the backoff to choose in form of a lookup table. Fig.7 shows that the summarized throughput is higher than using plain 802.11 but also higher as when using the *Nbs* scheme which could indicate that using more parameters than just the amount of neighbouring nodes can yield better results.

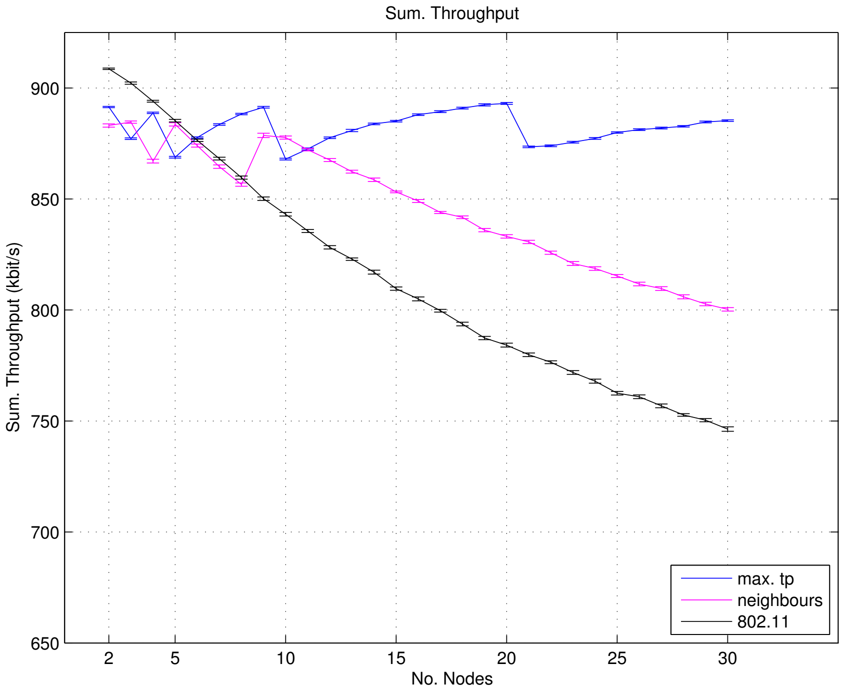

Figure 7: Mean of the summarized throughput using the *max. TP* scheme. Standard error plotted as errorbars for each set of simulations.

Again the drops in throughput can be explained by the table lookup mechanism and using powers of two. It is interesting to note however, that inverse to the *Nbs* scheme, *max. TP* always picks a "defensive" or "pessimistic" backoff. This can be seen in Fig.7 because when using five transmitting nodes the CW_{min} picked by *max. TP* is larger than necessary and does not change for six, seven, eight or nine transmitters. It does however increase the throughput for an increasing number of transmitters because where five nodes can not fully utilize the medium with a genuinely high CW_{min}, using that same CW_{min} for six or seven transmitters allows more transmitters access to the medium while keeping the likelihood of collisions low. When reaching ten transmitters the effectiveness of this CW_{min} starts to decline and therefore *max. TP* jumps to the next CW_{min} being again overly pessimistic for ten nodes and thus yielding a drop in overall throughput.

Compared to the *Nbs* scheme throughput, one can observe the inverse effect because the *Nbs* scheme picks the more "aggressive" or "optimistic" CW_{min} yielding the highest result for five transmitters but declining if the number of transmitters increases. Until a "jump" occurs and the CW_{min} dictated by the formula is represented by a different power of two.

Again the table lookup behavior and jumps can be seen in Fig.8 where the average CW_{min} choosen by *max. TP* is shown. As one can see, for more than 20 nodes a CW_{min} of 2048 is used, which is even higher than the CW_{max} limit defined by 802.11.

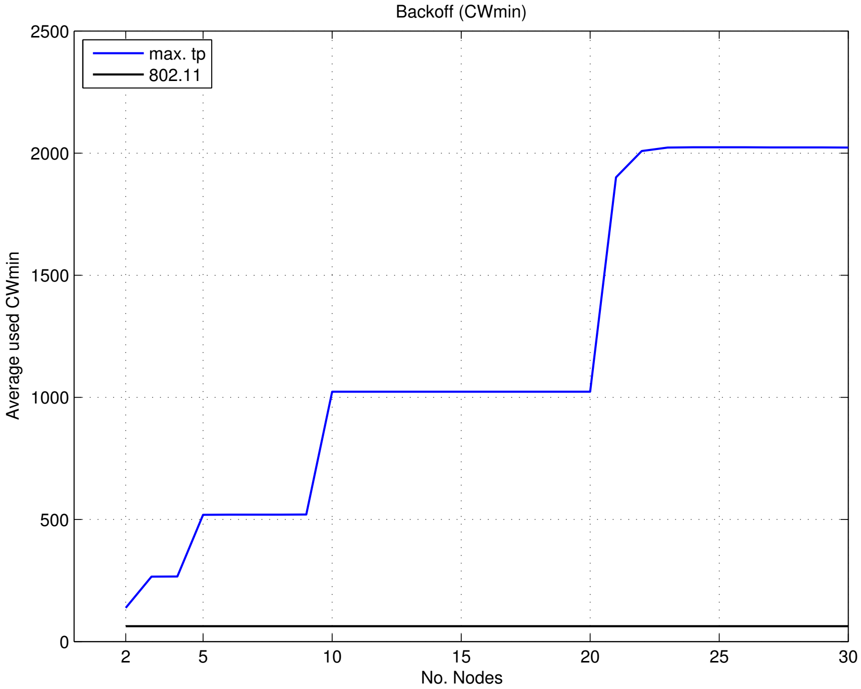

Figure 8: Average CW_{min} picked by max. TP

Fig.9 shows the fairness index results of *max. TP*. When comparing these results with the *Nbs* scheme, one can see, that *max.TP* surpasses *Nbs* in every regard because for even 30 contenting nodes it reaches 80 kbit/s more overall summarized throughput while still scoring higher in the used fairness index. This means that it can utilize eight percent more of the medium capacity while still achieving a more equal arrangement of medium access for the contenting nodes. This can be an indicator that using more parameters, e.g. number of neighbours, used MCS and MSDU size, than just the number of neighbours generally yields better results when done appropriately.

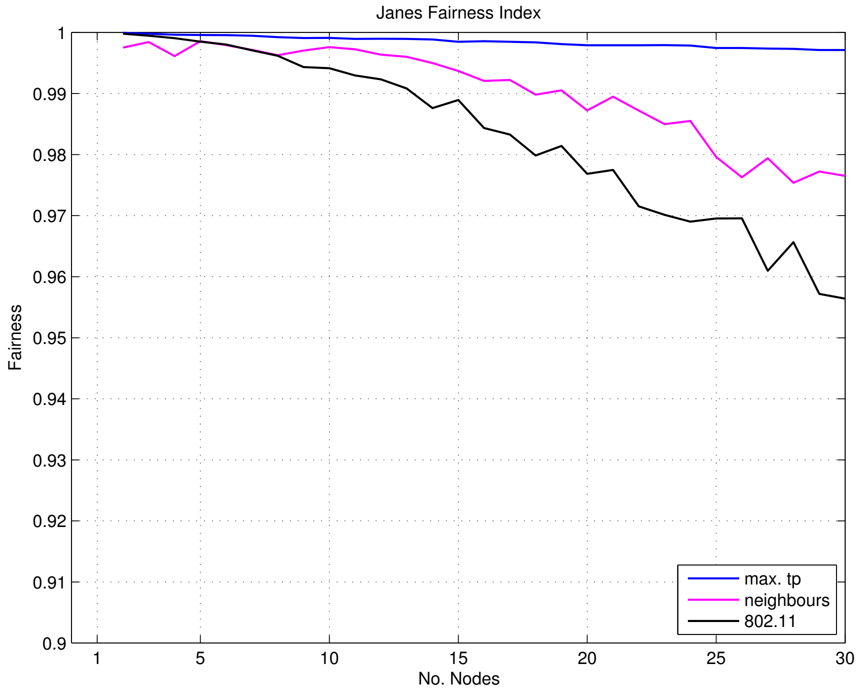

Figure 9: Jain's Fairness Index for the *max.TP* scheme

5.3 Ad hoc schemes

Sticking with the order of section 4 the next schemes to be evaluated are the *Learning*, *Busy Aware* and *Tx Aware* schemes. These are termed "ad hoc schemes" since they don't use pre-applied simulation but rather rely only and data that can be gathered during the live scenario.

5.3.1 Learning

The *Learning* scheme tries to evolve the value of CW_{min} by on the one hand remembering the CW_{min} used for the last transmission and on the other hand by only halving the contention window size in case of a successful transmission instead of resetting it to CW_{min}. To do so my implementation of the proposed scheme as a *Click* element used so called *Tx Feedback Packets*. After a packet was transmitted a feedback packet gets send up the network stack signaling whether or not the transmission was successful. This packet contains the number of retries used for the transmitted packet. For every retry the last packet needed, CW_{min} gets doubled, because, as proposed in [17], CW_{min} shall be doubled for every transmit failure. If the retry count for the last packet was 0, then CW_{min} gets halved, again, exactly as proposed in [17] and explained in section 4.

Fig.10 shows that this approach seems to optimize the overall throughput since it surpasses 802.11 for more than ten transmitters in a single collision domain. However, compared to the *max.TP* scheme, one can see that there are massive throughput gains left out by the *Learning* scheme. For as much as 30 contenting nodes the summarized throughout difference between the *Learning* and *max.TP* scheme is 80 kbit/s which amounts to 8% of the available medium capacity since all in all simulations a fixed rate of 1 Mbit/s was being used.

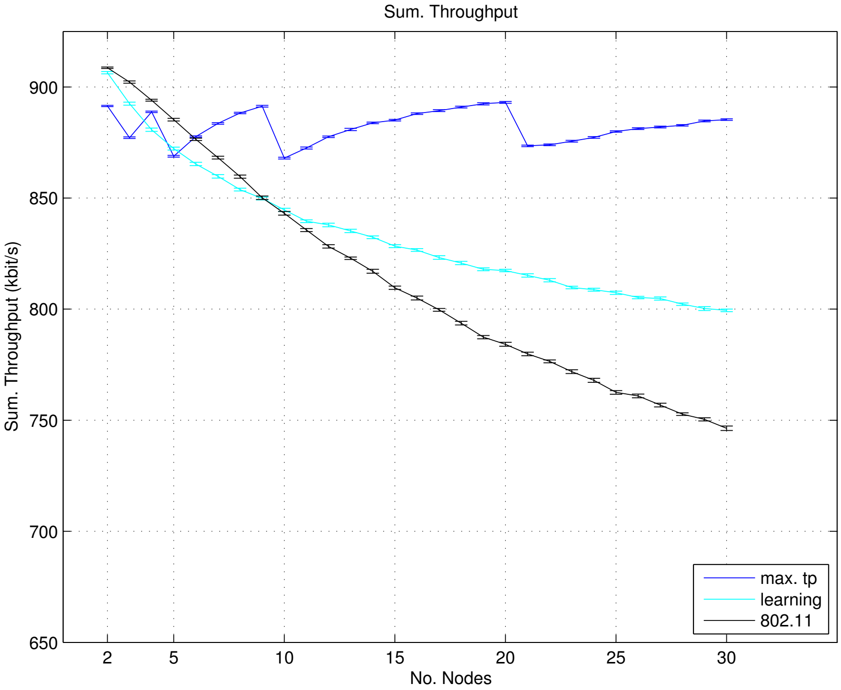

Figure 10: Mean of the summarized throughput using the *Learning* scheme. Standard error plotted as errorbars for each set of simulations.

When plotting the results of Jain's Fairness Index one can see, in Fig.11, that this approach seems to fail the fairness criterion because it is surpassed by plain 802.11 for even less than five contenting nodes and continues to deteriorate even further for an increasing number of contenting nodes. Again compared to the *max.TP* scheme this shows how far off this concrete learning approach implementation seems to be from what is actually possible.

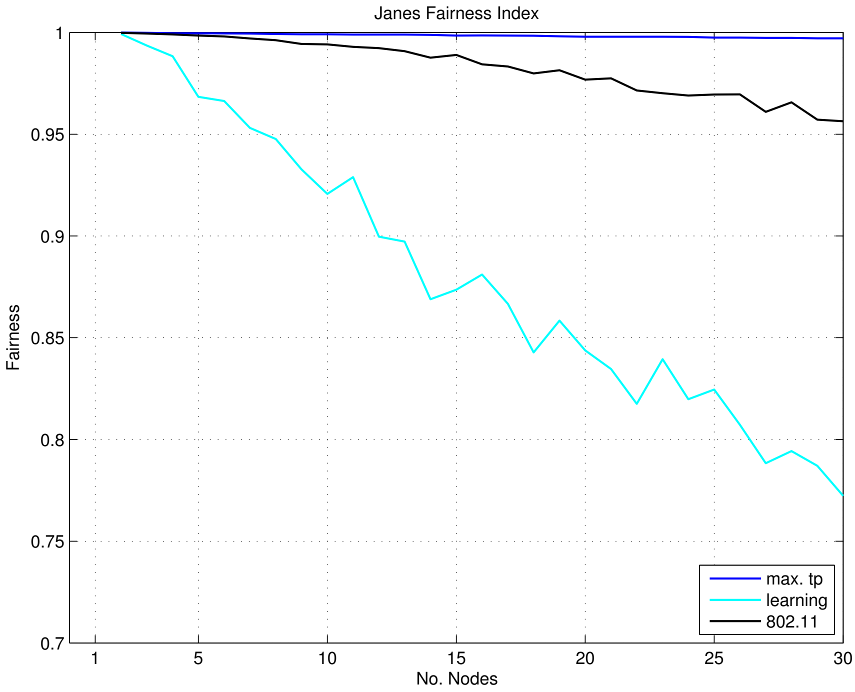

Figure 11: Jain's Fairness Index for the *Learning* scheme

This deficit can be explained by the overall approach of this scheme. Imagine two nodes, A and B, currently using Bo_A and Bo_B as CW_{min} with

$$Bo_A << Bo_B$$

(one CW_{min} being considerably smaller than the other). This is an unfair situation because while node B is waiting for his backoff counter to reach 0, node A can send untroubled and therefore deliver more packets. Since node B gets to send less packets he also gets less chances of decreasing his backoff, because he can only decreases his backoff for every *transmitted* packet. Therefore, if the difference between these two CW sizes does not get resolved somehow, then node A will permantly be able to transmit more packets than node B and B will have less chances of securing his rightful amount of medium share by decreasing his backoff.

If however node A never (or only occasionally) needs a retry then he

a) will not change his CW_{min} considerably, since from his perspective, there is nothing that tells him to so

b) has no chance to ever infer, that there might be some other node, that struggles to access the medium because of him

This results in sustained unfair behavior that does not get resolved by the *Learning* scheme and b) could be an indicator that some cooperative component is needed to radically improve this approach. When all nodes start with the standard CW size of 32 and some nodes need one, two or perhaps even three retries for their packet, they get catapulted into backoffs of 128 or 256 whilst all the other nodes whose packets did not collide continue to use 32 as a CW size. This means that those "outcast" have to wait longer before transmitting any new packet and therefore need more time until they have reduced their backoff back to where they started because they can only reduce their backoff for a transmitted packet. In that period of time all the other nodes with lower backoffs get so send more packets because of their lower backoffs but also because there where temporarily fewer contenting nodes, so the chance of collision was also slightly decreased. Since all the nodes were sending backlogged with a fixed MCS of 1 Mbit/s and a fixed MSDU size of 1500 byte, *fairness* is expressed only in the number of packets these nodes were able to transmit and therefore, even if a node who was sidelined earlier is able to reduce his backoff over time, he cannot catch up in terms of number of packets he transmitted overall which results in the observed drop in fairness. To solve this problem a global mechanism would be needed which organizes the sidelining of nodes so that the sidelining of nodes is equal among all contenting nodes.

I have tried to "relax" the scheme by not doubling the CW_{min} for every retry but rather only double once if there were *any* retries for a given packet transmission but that only improved the fairness performance slightly and did not solve the general problems of this approach.

5.3.2 Busy Aware

Next are the two backoff schemes *Busy Aware* and *Tx Aware* I consider my own approach which were introduced in section 4. The *Busy Aware* scheme only uses the *busy* register which represents the actual channel load a node detects. This channel load includes the node's own receive and transmit activity as well as whatever he receives of others. The target channel load that each node tried to achieve was set to 95% which represents a tradeoff between saturating the medium to the maximum bot also leaving *some* room for others. Additionally this allows nodes to overstep this boundary, which would not have been possible when set to 100%. Furthermore there is a slight wiggle room for channel loads *just* under the target channel load. This was introduced to relax the scheme and favour fairness rather than throughput because it prevents the scheme from decreasing its CW_{min} when the measured channel load is just underachieved by a few percent. Nodes are supposed to reduce their contention window size only when they really have to.

Fig.12 shows the throughput results which already look suspicious since the throupught seems to be just one steady value, no matter how many transmitters are competing for the medium. The drops however, when using less than twelve transmitters, seem to indicate some of the power of two rounding behavior we have already seen in the other backoff schemes. This can be explained by the wiggle room *Busy Aware* uses, to determine whether to change the current CW_{min} or not. The decision process can be described as follows:

$$new\ CW_{min} = \begin{cases} curr.\ CW_{min} * 0.5 & \text{if } chan.\ load < (target\ load - 5\%) \\ curr.\ CW_{min} * 2 & \text{if } chan.\ load > target\ load \\ curr.\ CW_{min} & otherwise \end{cases}$$

So that, for example, the channel load generated by five transmitters lies within that wiggle room that prevents decreasing the CW_{min} and therefore five transmitters waste medium idle time but for six or seven transmitter, even though their channel load still lies within that wiggle room, the same CW_{min} is closer to the optimum and therefore yields better throughput. Again the size of the wiggle room was chosen based on early tests.

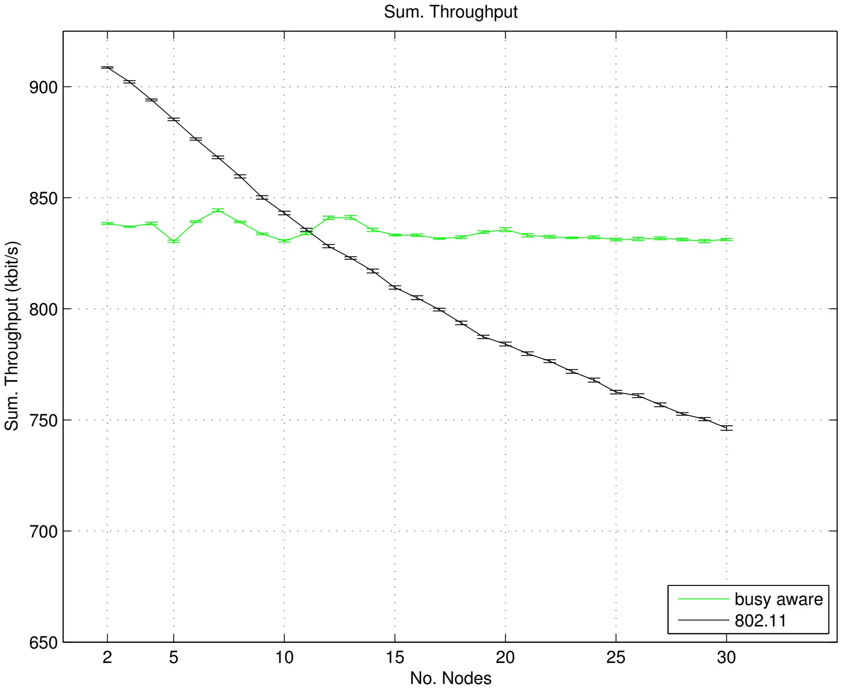

Figure 12: Mean of the summarized throughput using the *Busy Aware* scheme. Standard error plotted as errorbars for each set of simulations.

When plotting the fairness as shown in Fig.13 the suspicion gets confirmed and one can clearly see, that this scheme has some serious problems which unfortunately render it somehow useless, which was not immediately apparent when constructing it in theory.

As one can see, the fairness drops drastically when using more than ten transmitters which is exactly the number of transmitters where it naturally occurs that some nodes do not get access to the medium for even a second or two because nodes can only decrease their backoff when the medium is idle and with ten or more contenting nodes there is a high chance that the medium is so tightly saturated that a node does not reach 0 with his backoff during one second. This is exactly where *Busy Aware* starts to fail because a node only gets the register content statistic periodically. The interval that has been used for these so called *channel stats* was 1 s, so after a second a node knows whether or not his current CW_{min} is working or not (depending on the target load). If it is too high or too low, he will change his CW_{min} the next time *he transmits*. What happens now, if he does not get access to medium like everyone else? After an idle second he will get the new channel load of that second he spent idling, but which includes the activity of everyone else. If this still tells him to change his CW_{min} he is one step behind everyone else since everyone else already changed their CW_{min} when accessing the medium and they will now change it *again* when accessing the medium again. Even if the node does get access now, everybody sees the same channel load and therefore behaves the same. This results in *static shifts* where everyone changes their CW_{min} in the same way, but since there are offsets between some of the CW_{min} these offsets only get shifted but never get resolved.

This results again in two or more nodes having a sustained offset in their CW_{min} which is considered unfair and which unfortunately does not get resolved by the *Busy Aware* scheme. One possible solution for this problem would be to combine the data of the *busy* and *tx* registers, so that a node does not change his CW_{min} when his *tx* register says he did not get *any* medium access but this is considered future work.

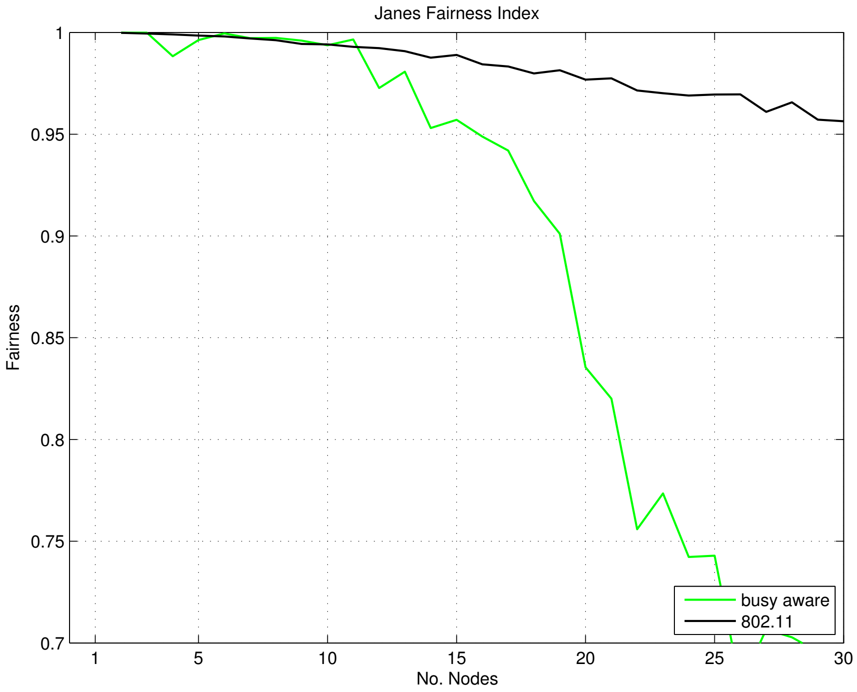

Figure 13: Jain's Fairness Index for the *Busy Aware* scheme

5.3.3 Tx Aware

The last scheme to be evaluated is the *Tx Aware* scheme which, as explained in section 4, uses the *tx* register to infer what percentage of the available medium a node did occupy by transmitting. To be able to react to any possible unfairness, a node needs the number of his neighbouring and therefore competing nodes, so that he can infer what percentage of the medium he is entitled to, by deviding it equally amongst all competing nodes.

As the statistics interval is set to 1 s a node knows what percentage of the last second he spend transmitting and therefore bases his backoff decision on what percentage he was supposed to occupy and what he actually did occupy.

Fig.14 shows the throughput results of the *Tx Aware* scheme which look promising as it surpasses 802.11 even when using only five transmitters.

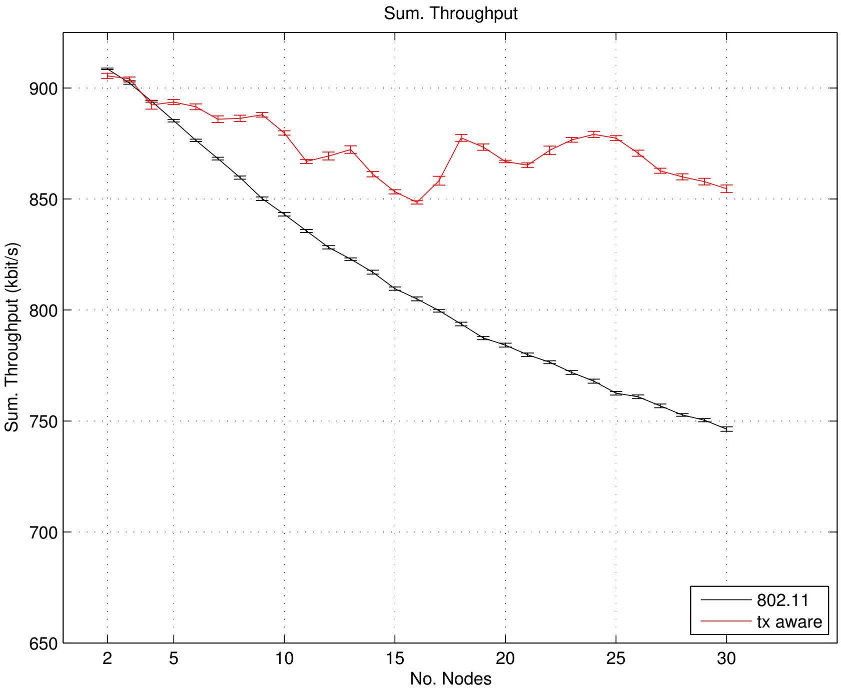

Figure 14: Mean of the summarized throughput using the *Tx Aware* scheme. Standard error plotted as errorbars for each set of simulations.

However, one must note that one disadvantage of this scheme is the unsteady performance which especially gets out of hand when the number of transmitting nodes surpasses ten. This can be explained again by the fact, that for ten and more transmitters, some nodes naturally do not get access to the medium within a second. Since the statistics interval is set to 1 s, if a node does not get access to the medium within that second, everyone else will not detect him as a neighbour and will therefore calculate his share of the medium considering seven or eight neighbours, rather than ten or eleven. Additionally the powers of two used as CW_{min} are too imprecise to achieve the optimal amount of medium share for certain numbers of nodes. For example when using 16 transmitters the summarized throughput of the *Tx Aware* scheme drops considerably as shown in Fig.14. This is because in this case, with 16 contending nodes, they are using a CW_{min} of 64 most of the time but are getting more medium share than they want to but when they increase their CW_{min} to 128 they get way less. They can't pick the *right* CW_{min} that gets them exactly the amount of medium share they are targeting. This results in unnecessary collisions and overall less throughput. For 18 nodes however this is totally different because in this case a CW_{min} of 128 turns out to be *just right* so that most of the time nodes do get their targeted medium share and only occasionally change their CW_{min} up or down. This results in less collisions and therefore a throughput gain.

However the *Tx Aware* scheme has another problem because it only works under the assumption, that every node wants to access the medium *all the time*. If however, there are neighbours who currently don't want to transmit but still occasionally send beacon packets, they will, depending on their beacon interval, get detected as neighbours and so any node again calculates his share of the medium considering perhaps more active neighbours than there actually are.

Looking at the fairness of the *Tx Aware* scheme, as shown in Fig.15, one sees satisfactory results because not only did the throughput surpass 802.11 but so does the fairness as well.

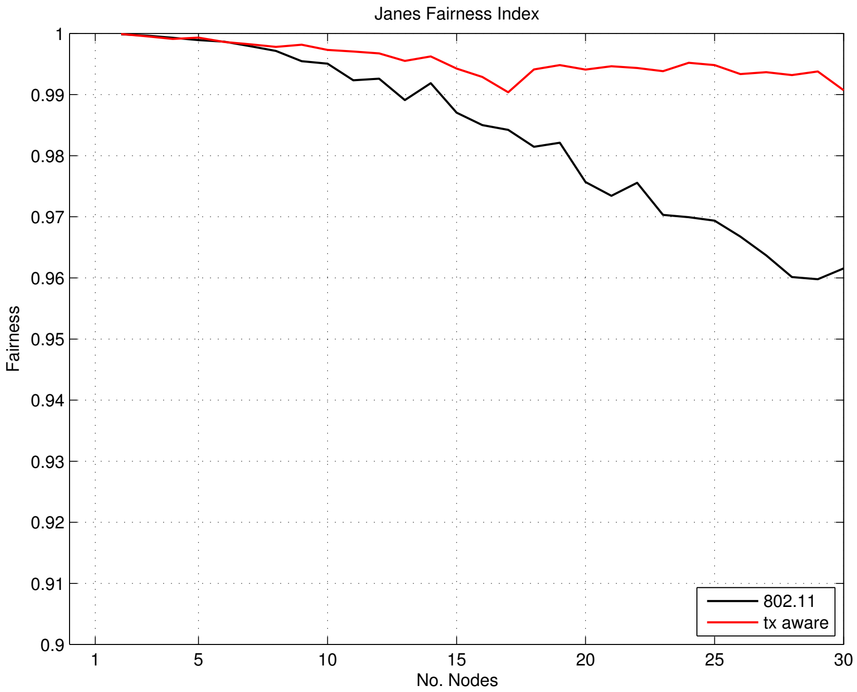

Figure 15: Jain's Fairness Index for the *Busy Aware* scheme

Unfortunately there is another problem with this approach, because for 30 transmitters the scheme already operates close to what can still be considered reasonable. A node detecting 29 neighbours, will reach the conclusion that he is entitled to roughly 3% of the medium. If the number of transmitters inside a single collision domain reaches 50, 100 or even more transmitters, how will a node choose his backoff to only access 1.5% or 0.5% of the medium? Since he can only double or halve his CW_{min} he reaches a point, where he cannot tune his backoff as delicately as his calculation of medium share demands.

This is actually how the scheme is able to cope with incorrect numbers of neighbouring nodes. As stated above, when using as much as 25 or more transmitters, the chance of every transmitter accessing the medium at least once during a second are very slim, at least when the nodes can not rely on information or backoffs from pre-applied simulations that sort of guarantee access to the medium at least once a second. So that when a node only detects 25 neighbours but actually has 29, the difference in percentage is very small considering that he can only double or halve his CW_{min} which will either result in way too much medium access or way less.

This shows a genuine tradeoff for this approach because the register contents or any live statistics are gathered periodically. This means that for a certain time interval in which new information are gathered, nodes have to operate on the old data from the last time interval. If this statistics interval is chosen to be very small then information is fresh and probably very close to the current situation but it is also possible that because of that not all important information could be gathered. If for example the statistics interval for the register contents in the shown experiments was dropped from 1 s to 500 ms it would be even more unlikely that all neighbouring nodes would have transmitted a packet in that time frame and therefore the neighbour detection would diverge even more from the current situation. If one chooses the interval to be larger the chance increases of capturing all important information but nodes now have to operate longer on the same old data before new data comes in.

To address the granularity issue one could imagine a lower bound for medium share. So that for 20 neighbours or more, a node assumes a medium share of 5% but however small that lower bound would be, assuming the same backlogged traffic demands for each node, one can always find a number of contenting nodes so that this lower bound yields a high amount of collisions and probably unfairness because nodes asume a fix medium share which is too optimistic.

5.4 Evaluation summary

To summarize the results of the individual evaluations Fig.16 and Fig. 17 show the results of the *Nbs*, *max.TP* and *TX Aware* scheme. These three backoff schemes are the top performers of the simulations shown here because not only do they surpass 802.11 by maximizing the overall network throughput but by doing so, they still provide fairness which, as a combination, is way harder to achieve than just optimizing for throughput alone.

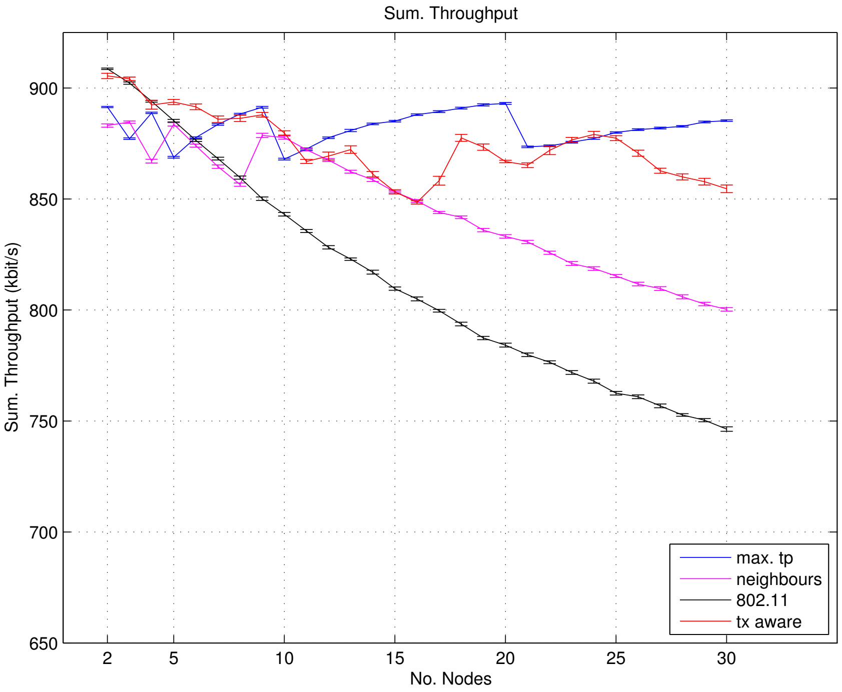

Figure 16: Mean of the summarized throughput of the best schemes. Standard error plotted as errorbars for each set of simulations.

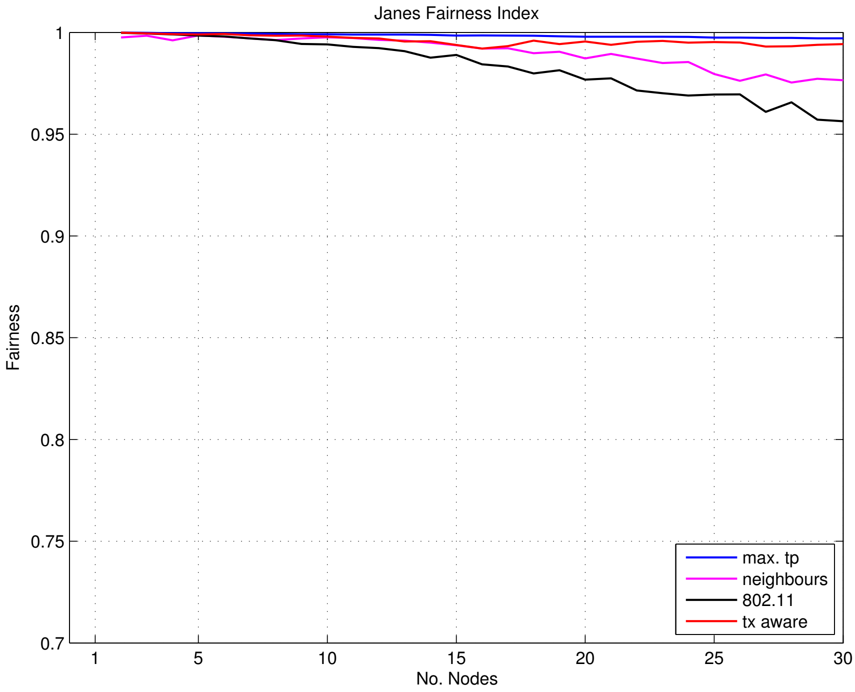

Figure 17: Jain's Fairness Index of the *Nbs*, *max.TP* and *Tx Aware* scheme

I left out the *Learning* and *Busy Aware* scheme because of their fundamental disadvantages and shortcomings as explained in section 5.3.1 and 5.3.2.

As can be seen, the *Tx Aware* scheme is able to compete with the two pre-applied simulation schemes *Nbs* and *max.TP* although it only uses live statistics in the form of the register contents of the wifi chip, as was explained in section 5.3.3. However, all three schemes rely on the accurate number of neighbours during the live scenario. In the presented simulations, the interval for gathering this information was 1 second which is bound to a tradeoff. If one uses a smaller interval, the information is more up to date since the duration of the interval determines when the nodes can actually react to the information gathered. So in this case, they have to wait 1 second until they can react to any new information. During the second they choose their backoff based on the information gathered during the last second. If this interval is decreased, they can react faster but the chance increases, that the information is not correct since the chance is higher that not every neighbouring node transmitted at least once during this smaller time frame.

This problem can be dealt with, when using pre-applied simulation. Predetermined by means of simulation, the *max.TP* always picks a backoff that provides a very high chance, that each node accesses the medium at least once every second and the formula used by *Nbs* scheme, which was also predetermine by means of simulation, exploits the rounding to powers of two to deal with an inaccurate number of neighbours. The *Tx Aware* scheme however really relies on the correct number of neighbouring nodes which explains the turbulent throughput performance.

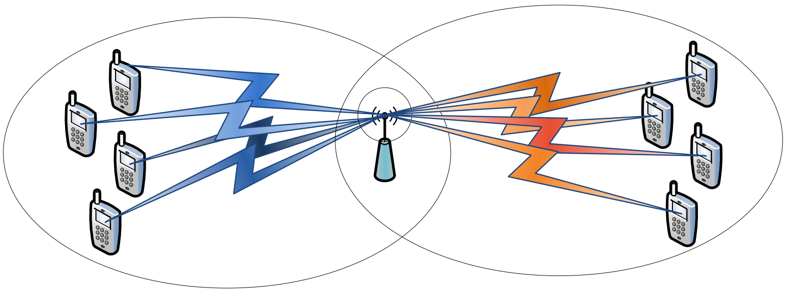

Figure 18: Multiple devices from two disjoint collision domains targetting a single AP

5.4.1 Hidden Nodes

Another problem that arrises since all three schemes need the number of neighbouring nodes in order to work, are *hidden nodes*. The hidden node problem is a fundamental problem of wifi networks [9, p.37-38] and describes a situation in which two transmitters target the same receiver without being able to sense each other. This results in each transmitter not knowing that there is another transmitter which means, he does not detect the other transmitter as a neighbouring node. Since all three backoff schemes rely on the number of neighbouring nodes this is a real problem because any hidden node is not detected as such although he is also competing for the medium. To demonstrate the effects of a hidden node scenario Fig.19 shows the throughput and fairness results of the same simulation parameter settings but instead of a single collision domain the transmitters were placed to form two disjoint groups that were out of carrier sensing range of each other, as shown in Fig.18.

If no hidden node detection or any hidden node interference counter mechanism like *Request To Send/Clear To Send* (RTS/CTS) [9, p.38] is being used, the performance of all three schemes as well as 802.11 is as near as makes no difference to non-existing. Partly because they all base their decisions on false information. If, for example, 20 transmitters are split up into two disjoint collision domains, both targeting a single receiver in the middle, as was the case in Fig.19, then every single transmitter in each of these groups detects only nine neighbours because the groups are out of carrier sensing range of each other. Using the *Tx Aware* scheme for example, this would mean dividing the medium equally considering ten nodes, when there really are 20.

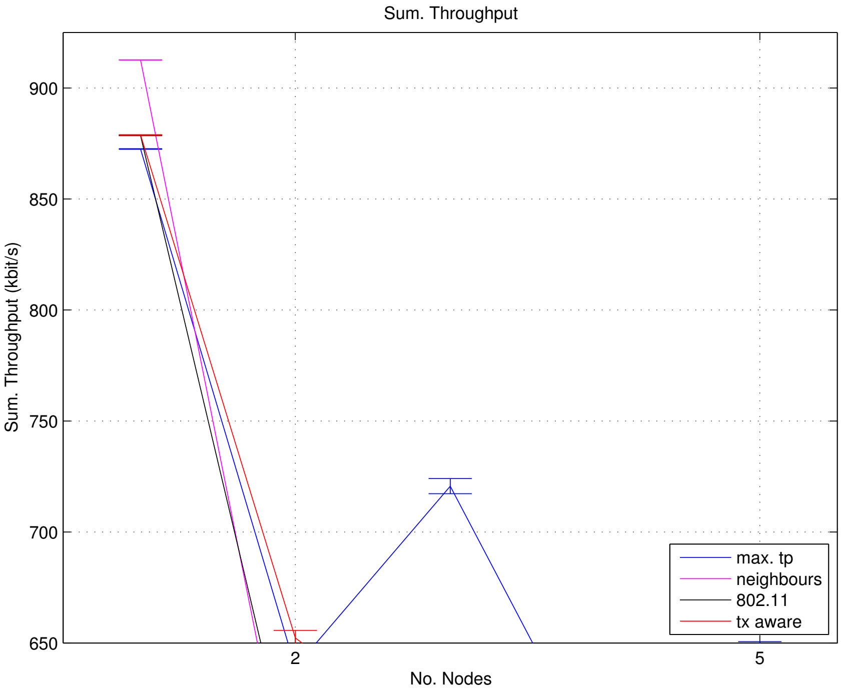

Figure 19: Mean of the summarized throughput in a multiple disjoint collision domain scenario *without* RTS/CTS

To account for hidden node scenarios the basic counter mechanism of plain 802.11, namely RTS/CTS would be needed. Additionally the pre-applied simulation schemes would need additional simulations where every possible case of a hidden node scenario would be simulated, to find the optimal backoff, which is then ready to use in the live scenario. The *Tx Aware* scheme however, would only need the information, how many hidden nodes there are, so that the calculation of medium fragmentation yields the correct results. Hidden node detection is an ongoing research topic. Even at the time of this writing, plans are discussed to incoorperate something like *cooperative channel stats* into the HWL simulation framework which could detect the number of hidden nodes in a given simulation but implementing this is considered out of the scope of this work and is rather considered future work.

So in summary this evaluation has shown, that it is possible to design a backoff scheme that uses only ad hoc live information to deduce which backoff to choose and still performs as well as pre-applied simulation schemes, at least for single collision domains.

6 Summary

In this work five alternative backoff schemes for IEEE 802.11 based wifi networks have been introduced, explained and evaluated by means of network simulation. Three of those are considered related work because they were proposed either as conference proceedings or, as in one case, a diploma thesis. The other two backoff schemes are my own proposed approach. Since a prominent way of realizing alternative backoff schemes is by using pre-applied simulation to test any possible situation for predetermining the optimal backoff, the proposed approach uses only live ad hoc information in the form of register contents of the hardware wifi chip. To realize this using simulation, the network simulator used as part of the HWL [5] simulation framework needed to be extended and as the simulation results presented in this work show, one of the proposed backoff schemes, namely *Tx Aware* is able to compete with the pre-applied simulation schemes *max.Throughput* and *Neighbours*. To see whether or not this was possible, was one of the main goals of this work.

The results show that the proposed *Tx Aware* scheme surpasses the 802.11 standard for more than eight contending nodes, delivering 850 kbit/s summarized throughput of the available 1 Mbit/s with 30 contending nodes instead of 750 kbit/s offered by 802.11. So it utilizes 10% more of the available medium capacity than the 802.11 standard for as much as 30 contenting nodes. It is only surpassed by the *max. TP* scheme which is able to deliver 880 kbit/s for 30 contending nodes. The *Tx Aware* scheme constantly surpasses the *Nbs* scheme for more than 16 nodes, delivering 55 kbit/s more for 16 contending nodes and still 51 kbit/s more for 30 contending nodes.

However, the scope of the evaluation results is limited because only a single collision domain was properly evaluated and since the hidden node problem is a fundamental problem of wifi networks in general and occurs heavily in real world wireless networks, these schemes should be evaluated using especially hidden node scenarios. Extending my own approach to deal with hidden nodes is future work.

Further still, the evaluation results are not only bound by the used single collision domain scenario and the backoff schemes alone but rather by using simulation in general, since it only allows a broad approximation of how these backoff schemes would perform, when being used by real hardware.

It would be interesting to see, how these backoff schemes would perform using the HWL hardware testbed, but again this is considered out of the scope of this work and is considered future work.

7 List of abbreviations

ACI *Adjacent Channel Interference*

ACK *Acknowledgement*

AP *Access Point*

BEB *Binary Exponential Backoff*

CSMA/CA *Carrier Sense Multiple Acces with Collision Avoidance*

CSMA/CD *Carrier Sense Multiple Acces with Collision Detection*

CW *Contention Window*

DCF *Distributed Coordination Function*

DLL *Data Link Layer*

DSSS *Direct Sequence Spread Spectrum*

FHSS *Frequency Hopping Spread Spectrum*

HWL *Humboldt Wireless Lab*

IEEE *Institute of Electrical and Electronics Engineers*

IFS *Inter Frame Spacing*

ISM *Industrial, Scientific and Medical*

MAC *Medium Access Control*

MANET *Mobile ad hoc Network*

MCS *Modulation Coding Scheme*

Mbit/s *Megabit per second*

MILD *Multiplicative Increase Linear Decrease*

MSDU *MAC Service Data Unit*

Nbs *Neighbours*

OFDM *Orthogonal Devision Frequency Multiplexing*

OSI *Open Systems Interconnection*

PCF *Point Coordination Function*

PHY *Physical Layer*

PLEB *Pessimistic Linear Exponential Backoff*

PRNG *Pseudo Random Number Generator*

RF *Radio Frequency*

RTS/CTS *Request To Send/Clear To Send*

SNIR *Signal to Nois and Interference Ratio*

WLAN *Wireless Local Area Network*

References

[1] Gartner: *Gartner Says Annual Smartphone Sales Surpassed Sales of Feature Phones for the First Time in 2013.* `http://www.gartner.com/newsroom/id/2665715`. Version: März 2014

[2] *Picture of the Mensa Nord of the Humboldt-Universität zu Berlin.* `http://www.haefner-jimenez.de/sites/default/files/Mensa_HU_Nord_Terrasse_01.jpg`. Version: February 2014

[3] *Picture of the central library of the Humboldt-Universität zu Berlin.* `http://upload.wikimedia.org/wikipedia/commons/5/53/Grimm-Zentrum_Leseterrassen.jpg`. Version: February 2014

[4] *IEEE 802.11 Working Group.* `http://grouper.ieee.org/groups/802/11/`. Version: 1999

[5] *The Humboldt Wireless Lab.* `http://hwl.hu-berlin.de/`. Version: February 2014

[6] *NS-2 Network Simulator.* `http://www.isi.edu/nsnam/ns/`. Version: February 2014

[7] Morris, Robert ; Kohler, Eddie ; Jannotti, John ; Kaashoek, M F.: The Click modular router. In: *ACM SIGOPS Operating Systems Review* Bd. 33 ACM, 1999, S. 217–231

[8] Tanenbaum, Andrew: *Computer Networks.* 4th. Prentice Hall Professional Technical Reference, 2002. – 41 – 43 S.

[9] Gast, Matthew S.: *802.11 Wireless Networks: The Definitive Guide, Second Edition.* O'Reilly Media, Inc., 2005

[10] *ITU-R Working Group.* `http://www.itu.int/en/publications/ITU-R/Pages/default.aspx`. Version: November 2013

[11] Schiller, J.H.: *Mobile Communications.* Addison-Wesley, 2003

[12] Bharghavan, Vaduvur ; Demers, Alan ; Shenker, Scott ; Zhang, Lixia: MACAW: a media access protocol for wireless LAN's. In: *ACM SIGCOMM Computer Communication Review* Bd. 24 ACM, 1994, S. 212–225

[13] Manaseer, S ; Masadeh, Muneer: Pessimistic backoff for mobile ad hoc networks. In: *Al-Zaytoonah University, the International Conference on Information Technology (ICIT'09), Jordan*, 2009

[14] Manaseer, S ; Ould-Khaoua, Mohamed ; Mackenzie, L: Fibonacci backoff algorithm for mobile ad hoc networks. In: *Liverpool John Moores University, the 7th Annual Postgraduate Symposium on the Convergence of Telecommunications, Networking and Broadcasting (PGNET 06), Liverpool* Citeseer, 2006

[15] Taifour, Mahmoud ; Naït-Abdesselam, Farid ; Simplot-Ryl, David: Neighbourhood backoff algorithm for optimizing bandwidth in single hop wireless ad-hoc networks. In: *Wireless Networks, Communications and Mobile Computing, 2005 International Conference on* Bd. 1 IEEE, 2005, S. 336–341

[16] Schröder, Boris: *Adaption von Übertragunsparametern in IEEE 802.11 Netzwerken*, Humboldt-Universität zu Berlin, diploma thesis, 2013

[17] Kuo, Wen-Kuang ; Liu, Fang ; Kuo, C-C Jay J.: Enhanced backoff scheme in CSMA/CA for IEEE 802.11. In: *AeroSense 2003* International Society for Optics and Photonics, 2003, S. 92–103

[18] Hühn, Thomas: *A measurement based joint power and rate controller for IEEE 802.11 networks*, Technische Universität Berlin, dissertation, 2013. – 49–53 S.

[19] *Madwifi Project - Atheros wifi chip registers.* `http://madwifi-project.org/wiki/DevDocs/AtherosRegisters`. Version: February 2014

[20] *The Click Modular Router Framework.* `http://read.cs.ucla.edu/click/click`. Version: February 2014

[21] Jain, Raj ; Chiu, Dah-Ming ; Hawe, William R.: *A quantitative measure of fairness and discrimination for resource allocation in shared computer system.* Eastern Research Laboratory, Digital Equipment Corporation, 1984. – 6–7 S.

Acknowledgement

I would like to thank everyone who helped me finish this work. Especially my advisor Robert Sombrutzki for enduring endless hours of me, asking stupid questions and helping me getting to grips with the simulation framework.

Statement of authorship

I declare that I completed this thesis on my own and that information which has been directly or indirectly taken from other sources has been noted as such. Neither this nor a similar work has been presented to an examination committee.

Berlin, June 12, 2014 . . . .