# Analysis of Injection Capabilities and Media Access of IEEE 802.11 Hardware in Monitor Mode

Stephan M. Günther*, Maurice Leclaire*, Julius Michaelis†, Georg Carle†
*Associate Institute for Signal Processing, Department of Electrical Engineering
†Institute for Network Architectures and Services, Department of Computer Science
Technische Universität München
Email: {guenther, j.michaelis, carle}@tum.de, leclaire@in.tum.de

***Abstract*—Support for monitor mode and frame injection is key to setup wireless testbeds based on IEEE 802.11 hardware that allow implementation and evaluation of custom link-layer protocols, e.g. network coding, opportunistic routing, and software defined networking. While monitor mode is a widely supported feature, frame injection seems to be limited to legacy data rates in the 2.4 GHz band if supported at all. In addition we found that many devices do not adhere to basic media access procedures when operating in monitor mode, which has severe effects in contended environments.**

**In this paper we investigate the injection capabilities and MAC procedures of different chipsets. To enable IEEE 802.11n rates and 5 GHz, we developed a series of small patches, which mostly apply to the generic part of the Linux drivers. In addition we present a command line tool for automated evaluation of injection capabilities of different devices. The patches, tools, and the underlying injection library used in this paper are publicly available [1].**

## I. Introduction

Monitor mode refers to an operational mode of wireless hardware that makes any type of valid IEEE 802.11 frames user-accessible. In contrast, a device operating in promiscuous mode accepts frames not destined for the local node as indicated by the receiver address but does not make available management and control frames. Frame injection, i.e., transmission of cooked frames including link layer header, is allowed only in monitor mode. Both features must be supported by the device drivers and firmware.

There are several examples of testbeds and protocols requiring monitor mode operation: In [2] a mesh testbed based on IEEE 802.11n hardware is presented that relies on monitor mode operation and raw frame injection. *MORE* [3] and COPE [4] are different network coding implementations that require a wireless interface operating in monitor mode capable of frame injection. *CloudMAC* [5], [6] is an OpenFlow-based [7] architecture that allows processing of IEEE 802.11 MAC frames on an OpenFlow controller. The implementation of access points in CloudMAC relies on monitor mode operation to forward link-layer frames. Investigating security issues of wireless networks also requires low-level access to the hardware. For instance, insecurities resulting from the virtual carrier sense mechanisms in IEEE 802.11 are investigated and practically evaluated in [8]–[10] which requires injection of control frames. The *Click modular router* [11] is a framework to create flexible software-based routers. It also offers the possibility to use monitor interfaces for frame injection, which was used for instance by the *MIT roofnet* project [12]. The variety of applications for native frame injection shows that there is reasonable scientific interest in hardware and drivers offering robust monitor mode operation.

978-1-4799-0913-1/14/$31.00 © 2014 IEEE

Identifying suitable chipsets and drivers for a testbed is difficult. Choosing devices with stable drivers and high throughput is a starting point but insufficient in general. The devices might still show significant MAC layer misbehavior, e.g. not adhering to basic media access rules or deliberately choosing non-standard backoff intervals. As a result, performance in a multi-node scenario is severely degraded although bulk injection rates of individual devices indicate good performance. Many researchers therefore rely on the popular Atheros/Qualcomm PCIe-based chipsets, most of them are known to support injection and offer solid and well-maintained drivers.

Media access procedures in wireless networks have been intensively studied in the past. The efficiency of collision avoidance mechanisms is analyzed in [13]. The basic access procedure, the *distributed coordination function (DCF)*, its backoff algorithm, as well as RTS/CTS protection are analytically modeled and analyzed in [14]. Theoretic throughput under heavy traffic conditions, i.e., many concurrent transmitters, is considered in [15]. An overview of various subsequent studies can be found in [16], and a comparative, measurement-based study of IEEE 802.11n compared to its predecessors is given in [17].

However, these analyses do not take the implication of monitor mode operation into account. One of the few publications dealing with performance of frame injection is [2], which presents a low-cost MIMO testbed based on IEEE 802.11n-capable Atheros/Qualcomm devices. Features such as per-packet rate selection and 5 GHz support also require driver patches that are not publicly available to the best of our knowledge.

This paper offers a comprehensive experimental analysis of IEEE 802.11 hardware. We investigate their injection capabilities and MAC procedures. This reveals significant differences between chipsets, which are partly due to MAC implementations not adhering to the standard. While this may give individual devices an advantage when contending for transmission opportunities, it may have serious side effects

in testbeds. In addition, we introduce `moep80211eval`, an automated test environment to evaluate the monitoring and injection capabilities of IEEE 802.11 hardware, which is based on our injection library `moep80211`. It allows to test the injection capabilities for a pair of devices with minimal effort, which is not only useful in identifying suitable chipsets but also to detect changes in behavior between different kernel and driver versions. The driver patches used in this paper, the injection library, and `moep80211eval` are publicly available at [1].

We emphasize that some of our experimental results – in particular those regarding violations of the DCF – may depend on driver versions and the operating system in use. For instance, the devices may exhibit different behavior when using drivers provided directly from the manufacturer instead of the Linux kernel drivers and their backports. The same holds for other operating systems. Table I found at the end of the paper lists the kernel and driver versions used for our measurements.

The remainder of this paper is organized as follows: Section II discusses the limits of frame injection and necessary modifications on the generic Linux drivers to enable full injection support on all PHYs. Section III gives a brief overview of the IEEE 802.11 MAC layer, derives upper bounds on achievable injection rates, and estimates the influence of nodes not adhering to basic MAC procedures in a two node network. Real-world performance and MAC behavior is analyzed by measurement for different chipsets in Section IV, proving MAC layer misbehavior on three out of six chipsets. Section V gives an overview of `moep80211eval`, an application for automated evaluation of monitor mode and frame injection capabilities of IEEE 802.11 hardware. Finally, Section VI summarizes our results and concludes the paper.

## II. Driver modifications

Without modifications, all devices we have tested so far are limited in three ways: First, injection is possible only at 802.11b/g rates in the $2.4\,\mathrm{GHz}$ band while monitoring is also possible in the $5\,\mathrm{GHz}$ band. Second, it is not possible to select the transmit rate on a per-packet basis as it should using radiotap headers. Instead, packets are sent at the native speed of the physical device that is used for injection, or at a fixed rate of $1\,\mathrm{Mbit/s}$ when a virtual monitor interface is being used. Third, injection at 802.11n rates is not possible by default.

The fact that these limitations apply to all chipsets indicates that the generic part of the Linux drivers, i.e., the mac80211 layer, does not support these features. Once we removed these limitations by a series of small patches [1] to the mac80211 layer, we were able to leverage the hardware's full injection capabilities. Supporting these features offers more flexibility in testing wireless protocols, in particular for wireless mesh networks. Furthermore, injection on the $5\,\mathrm{GHz}$ band allows to escape from the crowded $2.4\,\mathrm{GHz}$ band.

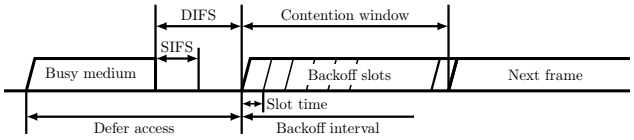



Figure 1. Media access procedure using DCF [18, p. 826].

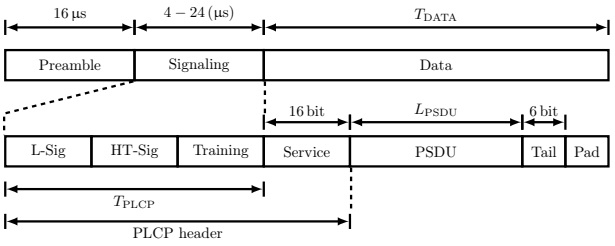



Figure 2. Generalized PLCP PDU: L-Sig ($4\,\mu\mathrm{s}$) is present only for legacy and HT mixed modes. HT-Sig ($8\,\mu\mathrm{s}$) is only present for HT mixed and HT GF modes. The Training subfield is also present for HT modes only. Its length depends on the number of spatial streams ($4/8/16/16\,\mu\mathrm{s}$ for 1/2/3/4 streams, respectively).

## III. Theoretic bounds

In this section we establish an upper bound on the achievable injection rate when operating in monitor mode. With the term *injection rate* we refer to the bulk transmit rate of a station without requiring that data transmitted is received or acknowledged. To calculate injection rates we consider the complete MPDU[1], i.e., a frame including its header and FCS. Throughput as seen by user space applications is thus slightly lower even in the absence of transmission errors due to protocol overhead. The bounds essentially apply to any IEEE 802.11 network using OFDM and operating in monitor mode.

Towards these bounds, we briefly discuss the media access procedure. Afterwards we discuss the implications of monitor mode operation and derive the expected media access time. Based on these results we derive the total transmit time of an MPDU which yields the injection rates. Finally, we derive an estimator for the injection rate when two nodes contend for the medium in dependency of the size of the backoff window for each node.

### A. Distributed coordination function

The basic method for media access is CSMA/CA. Since collision detection is in general not possible in wireless networks, collisions are avoided by requiring a minimum idle time plus a random backoff between consecutive transmissions. The idle time is determined by physically sensing the medium *and* virtual carrier sensing mechanisms, i.e., the frame duration field of the MAC header may set the network allocation vector at receiving stations indicating how long the medium will be busy. The latter offers interesting opportunities to mount DoS attacks against wireless networks as demonstrated in [8]–[10].

For IEEE 802.11 based networks the required idle time between frames is ensured by the *distributed coordination*

[1] MAC protocol data unit

*function (DCF)*, which must be supported by any station (see Figure 1). After preceding transmissions a constant minimum *DCF inter-frame space (DIFS)* plus a random number of slot times drawn uniformly i. i. d. from a backoff interval must be maintained. The DIFS denoted by $T_{\mathrm{DIFS}}$ is defined as

$$T_{\mathrm{DIFS}} = 2T_{\mathrm{SLOT}} + T_{\mathrm{SIFS}} \tag{1}$$

where $T_{\mathrm{SLOT}}$ and $T_{\mathrm{SIFS}}$ denote the slot time and *short inter-frame space (SIFS)*, respectively. The actual timing values of these variables depend on the PHY and can be found in [18, Sections 14 – 20]. The number of backoff slots to wait is defined by a random integer drawn uniformly i. i. d. from the set

$$\begin{aligned} \mathcal{C}_w &:= \{0, \ldots, C(n)\}, \\ C(n) &= \min\left\{2^{n+k} - 1, 255\right\}. \end{aligned} \tag{2}$$

Here, $n$ denotes the $n$-th retransmit of an MPDU and is reset to zero after successful transmission. An MPDU is considered to be transmitted successfully if it is either acknowledged by the receiver or completely transmitted in case of a multicast or broadcast. The initial (minimum) value given by $C(0)$ is determined by a PHY-specific parameter $k$.

As mentioned above, the DCF is the basic MAC procedure. There exist different refinements and a number of other coordination functions built on top of the DCF (see [18, Section 9]).

### B. MAC and PHY overhead in monitor mode

Since the DCF is the basic MAC procedure, it is likely that – if any MAC procedure is used – the DCF is employed while operating in monitor mode. However, it might not work as expected: As we aim at gaining as much control over the link layer as possible, we do not want to leave retransmits and acknowledgements to the driver. Nevertheless, the driver might expect an acknowledgement for unicast frames. If it is not received, the driver may make several attempts to retransmit an MPDU before giving up. This leads to superfluous and uncontrolled overhead and severe degradation of performance.

The common solution is to tell the driver not to expect an acknowledgement using the IEEE 802.11 radiotap[2] header [19]. However, this causes the contention window to remain at a maximum of $C(0)$ instead of growing since transmission errors can no longer be detected by the sender. The expected backoff time is thus easily calculated as $\overline{C} = \frac{1}{2}C(0)$. With the result of Equation (1) we obtain the expected time for media access

$$T_{\mathrm{MAC}} = 2T_{\mathrm{SLOT}} + T_{\mathrm{SIFS}} + \overline{C}. \tag{3}$$

This access time holds for all PHYs when sending data or management frames while operating under DCF. It does not apply to control frames, e.g. acknowledgements.

Aside from the MAC procedure, additional overhead is induced by the PLCP[3]. This differs significantly between specific PHYs. Assuming pure OFDM, the general PPDU[4] frame format is depicted in Figure 2. It consists of a preamble, signalling field, and the actual data. The format of the signalling field depends on whether or not frames are transmitted at legacy (802.11a/g), HT mixed, or HT greenfield rates. For HT rates, the training subfield additionally depends on the number of spatial streams used. The signalling subfield is sent at the most robust rate and coding scheme available. The transmit time of preamble and PLCP header is denoted by $T_{\mathrm{PLCP}}$. Timing values used in our calculations are included in Figure 2.

The service field is shared between both the PLCP header and the data field. The reason is that it semantically belongs to the PLCP header but is sent at the same rate and modulation as the remaining frame. The service field, PSDU, and the tail field are converted into a sequence of OFDM symbols which may involve padding to a multiple of data bits per symbol denoted by $N_{\mathrm{DBPS}}$. Given the symbol period $T_{\mathrm{SYM}}$ and the guard interval $T_{\mathrm{GI}}$, the transmit time of the data field is given by

$$T_{\mathrm{DATA}} = \left\lceil \frac{22 + L_{\mathrm{PSDU}}}{N_{\mathrm{DBPS}}} \right\rceil (T_{\mathrm{SYM}} + T_{\mathrm{GI}}). \tag{4}$$

The total transmission time is then given by the sum

$$T = T_{\mathrm{MAC}} + T_{\mathrm{PLCP}} + T_{\mathrm{DATA}}. \tag{5}$$

### C. Achievable injection rate

Based on Equation (5) we calculate upper bounds on achievable transmit rates for an IEEE 802.11n based node operating in the 5 GHz band in HT mixed mode. The results are shown in Figure 3. The surface shows the bounds according to Equation (5) that take the DCF, MPDU size, and MCS index into account. The efficiency ranges from 7 % to 58 % compared to nominal[5] transmit rates at MPDU sizes varying between 200 B and 3500 B. This underlines the well-known need for large MPDUs in high-speed wireless networks, e.g. [20], [21].

Furthermore, these results give a hint why – to the best of our knowledge – no wireless adapters with four spatial streams are available: the theoretic throughput gain at large MPDUs is only 14 % at HT 40. In addition, analyses of the open source Linux drivers revealed that most adapters seem to be restricted to a maximum MPDU size of about 4000 B or even the legacy value of 2346 B, which effectively nullifies the gain of a fourth spatial stream.

### D. MAC fairness

Assume two backlogged nodes 1 and 2 within range of each other that contend for the wireless broadcast medium. Assuming ideal sensing, a collision occurs if both nodes start transmitting at the same time slot, i. e., both nodes choose the same number of backoff slots. Given the current maximum

[2] Although commonly referred to as *IEEE 802.11* radiotap, it is an open source extension for Linux and BSD variants which is not officially supported or maintained by the IEEE.

[3] physical layer convergence procedure

[4] physical protocol data unit

[5] Nominal data rates are given by the MCS index, e.g. 150 Mbit/s at MCS 7 on a 40 MHz channel with 400 ns GI. According to Figure 3, the time average transmit rate ranges from roughly 11 Mbit/s at 200 B to 86 Mbit/s at 3500 B per MPDU.

size of the contention window $C_i$ at either node $i \in \{1, 2\}$, let the random variable $X_i \in \{0, 1, \ldots, C_i\}$ denote the number of backoff slots drawn for the given contention phase. According to the DCF, $X_1$ and $X_2$ are drawn independently and uniformly distributed. The probability that node 1 wins the contention phase is therefore given by

$$\begin{aligned}\Pr[X_1 < X_2] &= \sum_{k=0}^{C_2} \Pr[X_1 < X_2 \,|\, X_2 = k] \\ &= \sum_{k=0}^{C_2} \Pr[X_2 = k] \sum_{n=0}^{k-1} \Pr[X_1 = n] \\ &= \frac{2C_2 - C_1}{2(C_2 + 1)}. \end{aligned} \tag{6}$$

For sufficiently large contention windows, Equation (6) converges to $1 - C_i/(2C_j)$ while the probability of collision tends to zero. Consequently, the medium is equally shared between nodes 1 and 2 if the same window sizes are used. In contrast, when $C_1 = C_2/4$ is chosen, node 1 occupies the medium for approximately $88\,\%$ of the time. It has been demonstrated in [22] by simulations that nodes in a BSS consisting of 9 stations (one acting as AP) still suffer a $50\,\%$ performance loss if one station uses $C_{\min}/4$ compared to the others.

Next, we consider the expected length of the contention phase when two nodes contend for media access. Given random variables $X_i \in \{0, 1, \ldots, C_i\}$ for $i \in \{1, 2\}$ independently and uniformly distributed, each representing the number of slot times a transmitter waits before starting to transmit in a given contention phase. Without loss of generality let $C_1 \leq C_2$. Then the expected length of the contention phase is given by

$$\begin{aligned}&\mathrm{E}[\min\{X_1, X_2\}] = \\ &= \sum_{k=0}^{C_1 - 1} \left(1 - \Pr[X_1 > k]\right) \left(1 - \Pr[X_2 > k]\right) \\ &= \frac{C_1(3C_2 - C_1 + 1)}{6(C_2 + 1)} \leq \mathrm{E}[X_1] \leq \mathrm{E}[X_2]. \end{aligned} \tag{7}$$

Consequently, the average time the contention phase lasts is reduced, resulting in less idle time of the medium and thus increased combined injection rates. For the special case $C_1 = C_2 = C$, Equation (7) simplifies to

$$\mathrm{E}[\min\{X_1, X_2\}] = \frac{C(2C + 1)}{6(C + 1)}. \tag{8}$$

As an example consider a node operating at MCS 7 on a $40\,\mathrm{MHz}$ channel with $400\,\mathrm{ns}$ GI. According to Figure 3, the theoretic maximum injection rate for a single node is roughly $88\,\mathrm{Mbit/s}$. The average length of the contention phase is $7.5$ slot times since backoff slots are drawn uniformly i. i. d from the set $\mathcal{C}_w = \{0, 1, \ldots, 15\}$. Now assume that two nodes are transmitting concurrently. According to Equation (8), the expected length of the contention phase reduces to $4.84$ slot times. This increases the theoretic injection rate to $95\,\mathrm{Mbit/s}$. In addition both nodes start transmitting at the same time with probability $1/16$, which further increases the sum of injection rates to $101\,\mathrm{Mbit/s}$. Of course, this additional traffic is subject to collisions and therefore does not result in higher goodput.

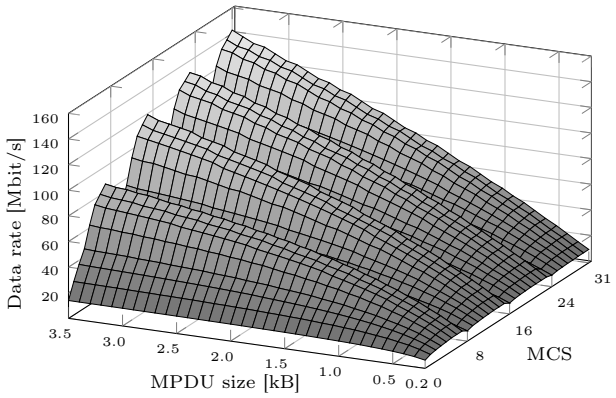



Figure 3. HT 40 mixed, 400 ns, 5 GHz. The surface plot shows the calculated upper bound with respect to DCF in dependency of MCS index and MPDU size.

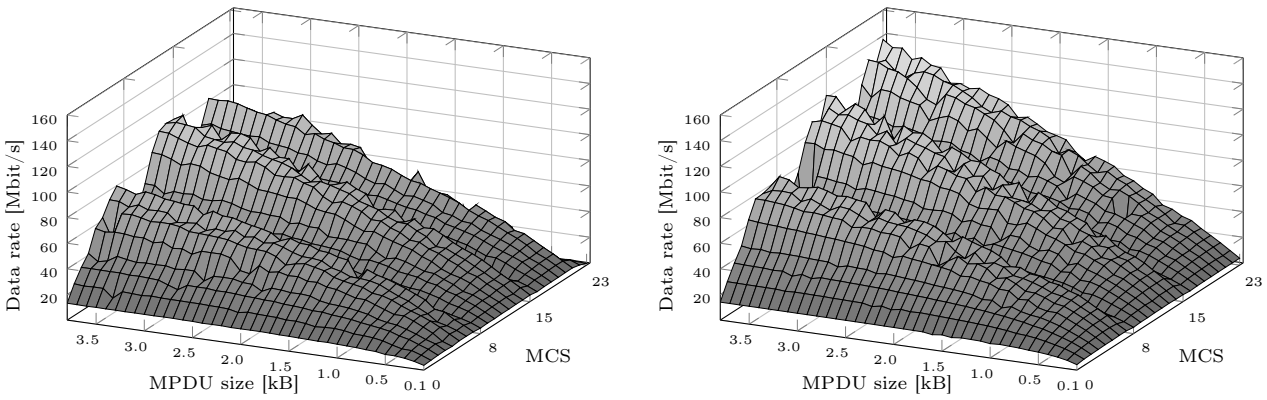
## IV. Experimental analyses

We analyze injection rate, MAC behavior, and fairness for a variety of different chipsets. The first test series in Section IV-A compares the injection rate (bulk transfer rate) and goodput (fraction of successfully received traffic) for the AR9380 chipset at various settings in both the 2.4 and 5 GHz bands. The results validate the theoretic bounds derived in Section III. Furthermore, this gives an overview of achievable injection rates, underlines the advantages of using the $5\,\mathrm{GHz}$ band, demonstrates that large MPDUs are critical for high throughput, and unveils detection issues at sophisticated modulation and coding schemes in spite of excellent link conditions. Section IV-B gives an overview of injection rates and goodputs for a variety of different chipsets. This gives an idea of different hardware capabilities and unveils that some of the chipsets transmit at rates higher than predicted in Section III-C, which hints at violations of MAC procedures. To analyze whether or not the chipsets adhere to the DCF when operating in monitor mode, we measure interarrival times of frames using hardware timestamps and derive the empirical CDF of backoff slots used in Section IV-C. Finally, Section IV-D investigates the influence of devices not adhering to the DCF on standard-conform devices. All tests are performed on our mesh nodes with identical antenna setups[6] and patched drivers. A list of hardware and driver versions is given in Table I at the end of this paper.

### A. 2.4 vs. 5 GHz band

We compare the goodput of unidirectional traffic between two AR9380 chipsets. We tested MCS 0–23 at MPDUs ranging from $200\,\mathrm{B}$ to $3800\,\mathrm{B}$. The AR9380 chipset was chosen for this test as it supports up to three spatial streams and delivers stable and reproducible results. Note that the theoretic

[6]Internal/standard antennas are used for the RT2870-based USB adapter.




(a) 2432 MHz

(b) 5825 MHz

Figure 4. Goodput between two Atheros AR9380 chipsets at HT 40+, 400 ns GI.

bounds according Section III are approximately the same for both frequencies. In fact, the 2.4 GHz band offers slightly larger transmit rates: at MCS 23 and an MPDU of 3500 B the achievable rate is 2.9 % larger than in the 5 GHz band. The reason consists in different SIFS intervals (10 µs compared to 16 µs) and thus shorter DCF interframe spacings.

The results are depicted in Figure 4. The throughput gain when using the 5 GHz band is approx. 20 % but results are not always more stable as we might expect due to less interfering networks. It should be noted that both measurements have been conducted during periods of low activity, which reduces the interference of other networks for the 2.4 GHz band significantly: At a channel width of 40 MHz with the main channel centered at 2432 MHz (channel 5) and the secondary channel at 2412 MHz (channel 1), any networks operating on channels 1–8 were interfering with our setup.

Although transmitter and receiver were positioned in direct line of sight at a distance of 2 m, goodput significantly fluctuates at large MPDUs and dense coding schemes. We found that MCS 22–23 are virtually unusable since lower MCS 20–21 (less dense MCS at three spatial streams) or even MCS 14–15 (dense MCS at two spatial streams) yield considerably lower packet loss.

The injection rates at the sending node (not shown in Figure 4) are slightly lower than the bounds derived in Section III, e.g. 96 % – 99 % of the theoretic values for the tests at 5 GHz. In the 2.4 GHz band, the injection rate reaches only 78 % – 83 % of the theoretic values, which is due to contention with other networks.

### B. Chipsets

We compare the injection rate (TX) and the goodput (RX) between six different IEEE 802.11n-capable chipsets. To this end we equipped our stations with two identical chipsets each time. The tests are performed at channel 5 in the 2.4 GHz band as not all of the chipsets support 5 GHz. We test MCS 0–7 at an MPDU size of 1500 B using both 20 MHz channels with 800 ns GI and 40 MHz channels with 400 ns GI. The results for the different setups are shown in Figure 5. The horizontal lines indicate the upper TX bound for the respective settings as derived in Section III-C.

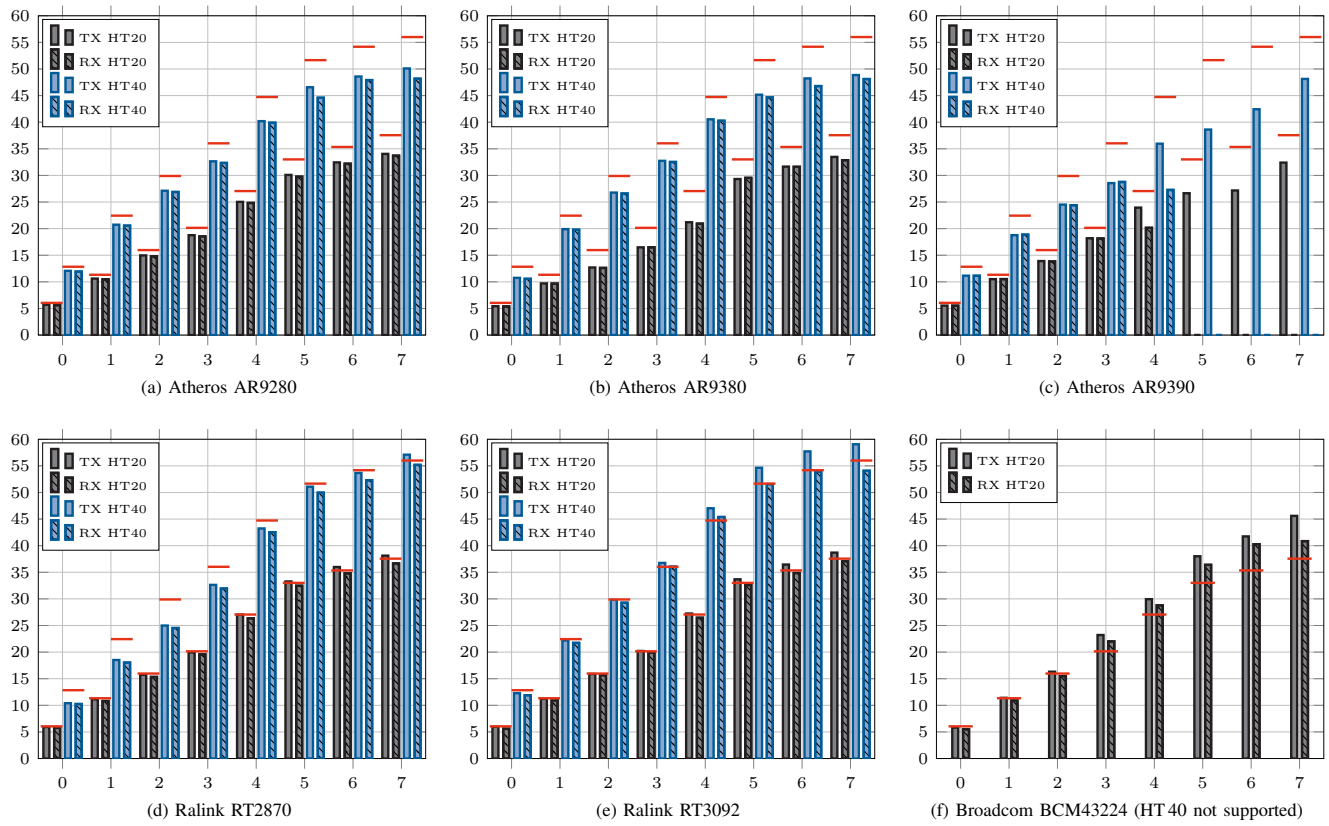
Figures 5a–5c show the results for three different Atheros chipsets. Most notably the AR9390 seems to malfunction beginning at MCS 5. As MCS 5–7 are the only demanding 64-QAM, this behavior may indicate a problem with detecting densely coded QAM signals at the receiver. However, an AR9280 running in the background was also unable to detect the transmission of AR9390 beginning at MCS 5, which definitely hints at a problem at the transmitter side. The AR9280 and AR9380 chipsets shown in Figure 5a and 5b deliver virtually the same injection rates and goodputs.

Figures 5d–5e show the results for a Ralink RT2870-based USB adapter and an RT3092-based mini-PCIe card. Despite excellent performance and low packet loss it is very interesting that both devices slightly exceed our upper bounds on the achievable injection rate. As proven in Section IV-C and IV-D, this phenomenon is not an error in our bound but due to backoffs chosen from the wrong interval, which gives these devices an advantage in media access.

Finally, Figure 5f shows the results for a Broadcom BCM43224-based device. The `brcmsmac` driver does not support 40 MHz channels at the moment. The results at 20 MHz considerably exceed our bounds. Again, this is due to misbehavior in MAC procedures: the BCM43224 paired with the `brcmsmac` does not use any contention phase, which results in excellent injection rates and goodput in the unidirectional case. However, it is expected that standard-conform devices almost fail completely while a BCM43224 is transmitting. Further analyses of the BCM43224 chipset are provided in Sections IV-C and IV-D.

### C. Analysis of MAC procedures

Noting that the abnormal high transmit rates of three chipsets become admissible according to our bounds when the expected length of the contention window in Equation (3) is reduced, we analyzed the inter-arrival times of packets at the receiver. For this purpose we equipped the receiver with an AR9380 chipset and read the TSFT timestamps provided by the driver. These represent the time when the first bit of a




(a) Atheros AR9280 (b) Atheros AR9380 (c) Atheros AR9390

(d) Ralink RT2870 (e) Ralink RT3092 (f) Broadcom BCM43224 (HT 40 not supported)

Figure 5. Injection rate (TX) and goodput (RX) measured in Mbit/s at MCS 0–7 for different pairs of chipsets. All tests were conducted with MPDUs of 1500 B at 2432 MHz (channel 5) at HT20 / 800 ns GI and HT 40- / 400 ns GI, respectively. The only exception is the BCM34224 since its driver `brcmsmac` does not yet support HT 40. The horizontal (red) bars indicate the theoretic injection rates as derived in Section III-C.

frame arrives at the PHY measured in µs relative to a fixed point in time. If the transmitter adheres to the DCF, then

- the minimum time between two consecutive frames must not be smaller than $2T_{\text{slot}} + T_{\text{SIFS}}$, and
- the average time between two consecutive frames should equal $T_{\text{MAC}}$.

In particular, $T_{\text{MAC}}$ contains the randomized contention window $\mathcal{C}_w$ measured in slot times, which should be chosen uniformly distributed and independently drawn for each frame.

We tested all chipsets at MCS 7[7] using a 20 MHz channel at 2432 MHz. At these settings, backoff slots should be chosen from the window $\mathcal{C}_w = \{0, 1, \ldots, 15\}$ giving an expected number of $\overline{C} = 7.5$ backoff slots [18]. The empirical CDF of backoff slots chosen are shown in Figure 6. The ECDF of a device adhering to the DCF should represent a stepwise linear function reaching 1 at $15 \leq N < 16$ since backoff slots are drawn uniformly and independently distributed from $\mathcal{C}_w$. Note that stations are not synchronized with each other, which is the reason why the steps are not perfectly aligned at integer values.

The first thing to note is that none of the plots reaches 1, i.e., a fraction of $5 - 10\,\%$ of the frames are deferred for more than 15 slot times. This is not unusual even in monitor mode since interfering stations of wireless networks not belonging to our testbed may win the contention phase, which in turn causes a frame to be deferred. Furthermore, the monitoring station may have missed frames which results in gaps between consecutive TSFT values. A possible way to detect such missed frames would be to assign sequence numbers to our test traffic, which has not been done for our measurements.

Regarding our chipsets, the Atheros-based devices perform close to expectations. A close look shows that the fraction of frames deferred by more than 15 slot times is a bit larger for the AR9390 chipset. This is probably a result of a higher packet loss compared to the other Atheros devices (compare to Figure 5c, MCS 4 at 40 MHz).

The test further reveals that the Ralink chipsets obviously choose their contention windows from the set $\{0, 1, \ldots, 7\}$, essentially cutting the expectation of slot times in half. At MCS 7 and a 40 MHz channel with 400 ns GI, this means an increase from 56 Mbit/s to 67 Mbit/s according to our bounds, which is an increase of roughly 20 %. Note that this makes the achieved transmit rates in Figures 5d–5e admissible.

Finally, the BCM43224 does obviously not use any contention window for at least 90 % of its traffic. The remaining 10 % are either frames missed by the monitor station or deferred frames due to cross traffic by other networks. Note that choosing zero backoff slots does not guarantee to win the contention phase as stations are not synchronized. As a consequence, slot times overlap such that another station that

[7] MCS 4 for AR9390 due to the issues with QAM.

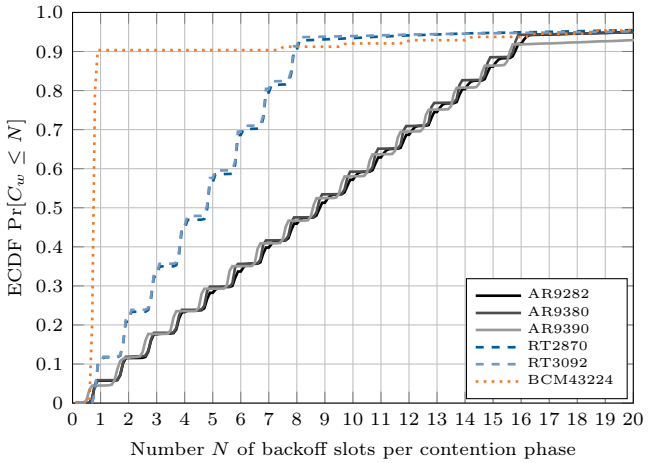



Figure 6. Empirical cumulative distribution function (ECDF) indicating the probability that the number of backoff slots per contention window is smaller or equal to $N$.

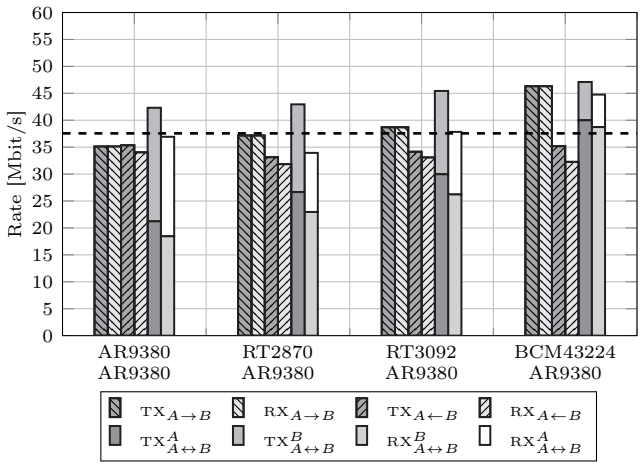



Figure 7. Measurements were conducted at 2432 MHz, HT 20, 800 ns GI with MCS 7. The bars indicate traffic $\mathrm{TX}_{A\to B}$ injected by $A$ and goodput $\mathrm{RX}_{A\to B}$ received by $B$. $\mathrm{TX}_{A\leftarrow B}$ and $\mathrm{RX}_{A\leftarrow B}$ denote the opposite direction. The cumulative bars depict total injection and goodput rates, i.e., the sum of both nodes. The shading represents which node is injecting or receiving, e.g. $\mathrm{TX}^{A}_{A\leftrightarrow B}$ means the fraction of traffic injected by $A$ when both nodes are transmitting concurrently. The dashed line indicates the maximum injection rate for a single node at these settings when the DCF is obeyed.

has chosen zero backoff slots may start transmitting before the Broadcom device does, resulting in either a collision (and thus a frame missed by the monitor station) or the BCM43224 remaining silent. Due to the missing contention phase, the theoretic injection rate increases from 38 Mbit/s to 48 Mbit/s, making the injection rates shown in Figure 5f admissible again.

### D. MAC fairness

We confirm the results of Section IV-C by analyzing the injection rates and goodput between two nodes when both sides contend for media access. First, we determine average injection and goodput rates when only one station transmits. Afterwards, we determine the sum of the rates at which both station are transmitting and receiving.

The results are depicted in Figure 7. For each chipset the first two bars indicate the unidirectional injection rate $\mathrm{TX}_{A\to B}$ and the goodput $\mathrm{RX}_{A\to B}$. The next two bars indicate the same values for the reverse direction. The fifth bar indicates the injection rates $\mathrm{TX}^{A}_{A\leftrightarrow B}$ and $\mathrm{TX}^{B}_{A\leftrightarrow B}$ of nodes $A$ and $B$, respectively, when both stations are transmitting concurrently. The last bar indicates the goodputs $\mathrm{RX}^{A}_{A\leftrightarrow B}$ and $\mathrm{RX}^{B}_{A\leftrightarrow B}$ in this scenario. The dashed line indicates the upper bound on the injection rate of a single node according to the standard.

The AR9380 chipset shows a symmetric connection, little packet loss, and in particular an approximately even sharing of bandwidth in case of bidirectional communication. Note that the combined transmit rate of two AR9280 chipsets exceeds the maximum for a single transmitter. As derived in Equation (8), the average time the contention phase lasts is reduced, resulting in less idle time of the medium and thus increased combined injection rates. In addition both stations may start transmitting concurrently with probability $\frac{1}{C+1}$, which causes a collision but is counted in the injection rate. Given $C = 15$ for this PHY, the the theoretic maximum for the combined rate increases from about 37 Mbit/s to about 42 Mbit/s. On the other hand, on average $1/16$ of the transmitted frames collide, which decreases the theoretic goodput to 39 Mbit/s in case of no further losses. Both estimates correlate with the results in Figure 7.

Next we consider mixed setups, i.e., we test the AR9380 with other chipsets. When combined with an RT2870 we see that only $1/3$ of the injected traffic originates at the AR9380, which is expected since the RT2870 uses smaller contention windows. Inserting $C_1 = 7$, $C_2 = 15$ into Equation (6), 71.8 % of the contention phases are won by the RT2870 chipset. Considering the goodputs according to Figure 3 for this combination, wie find that 69.3 % of the total goodput originates at the RT2870. Very similar results are obtained when the AR9380 is paired with the RT3092, which is also expected since both Ralink chipsets choose backoff slots from the same set.

In combination with the BCM43224 we see that only 15 % of the traffic originates at the AR9380 – which is more than expected provided that the BCM43224 does not use a contention window at all. The explanation for this phenomenon is that IEEE 802.11 is slotted but not synchronized, i.e., when the AR9380 chooses zero slot times for the current contention phase it has a chance to start transmitting before the BCM43224 does.

These results demonstrate the consequences of not adhering to the DCF. In the worst case, vast amounts of collisions occur. When combining selfish devices with standard-conform chipsets, one direction significantly suffers. Thus, it is essential to be aware of how devices implement media access when building testbeds or evaluating protocols. This is of particular importance for mesh networks as forwarding nodes must not be disadvantaged when it comes to media access. Evaluation results may otherwise be biased or even inconclusive.

## V. MOEP80211EVAL

As demonstrated in this paper it is important to quickly and automatically test wireless hardware for their injection capabilities and interoperability with other chipsets. This is particularly true when building testbeds for wireless mesh networks that rely on frame injection. To this end we developed `moep80211eval` that is based on our injection library `moep80211`, which are both available at [1]. The evaluation tool requires two wireless devices attached to the same physical node. For a complete test, it only requires the interface names of the network devices to be tested. Please see [1] and the `README` for details on how to compile `moep80211` as shared library. Afterwards, `moep80211eval` can be started from the command line. First, the tool tries to activate monitor mode for both WLAN devices. Afterwards, It determines the maximum working MTU for both devices. Note, that the MTU varies between driver versions and depends on whether or not our patches are applied. Next, the available channels are checked by setting the corresponding frequency. Channels are implicitly numbered in ascending order starting at channel 1. Supported channel numbers are indicated by a "+" on the command line while a fail to set the channel is indicated by a "-". Finally, legacy and MCS data rates are checked for each channel. Success and fail are indicated in the same way as the supported channels. An example for a Ralink RT2870-based device combined with an Atheros AR9380 chipset is given in Listing 1:

Listing 1. moep80211eval sample. Output is abbreviated to fit column width an remove redundant information.

```
root@m7 ~/moep80211/moepeval# ./moepeval wlan0 wlan1
Computing maximum MTU...
max MTU wlan0: 3925
max MTU wlan1: 2325

Testing available channels...
available channels wlan0: ++++++++++++++++++++++++++++++(...)
available channels wlan1: ++++++++++++++----------------(...)

Testing working rates...
wlan0 -> wlan1, 2412 MHz: ++++++++++++|++++++++--------(...)
wlan0 -> wlan1, 2417 MHz: ++++++++++++|++++++++--------(...)
wlan0 -> wlan1, 2422 MHz: ++++++++++++|++++++++--------(...)
(...)
wlan0 -> wlan1, 2472 MHz: ++++++++++++|++++++++--------(...)
wlan0 -> wlan1, 2484 MHz: ------------|----------------(...)
wlan1 -> wlan0, 2412 MHz: ++++++++++++|++++++++--------(...)
wlan1 -> wlan0, 2417 MHz: ++++++++++++|++++++++--------(...)
wlan1 -> wlan0, 2422 MHz: ++++++++++++|++++++++--------(...)
(...)
wlan1 -> wlan0, 2472 MHz: ++++++++++++|++++++++--------(...)
wlan1 -> wlan0, 2484 MHz: ------------|----------------(...)
```

Interface `wlan0` is obviously the Atheros device as it supports channels in the 5 GHz band while the Ralink adapter is limited to channels 1–14. Afterwards the tool checks whether or not data is mutually being received. As can be seen, the Ralink adapter limits the setup to legacy rates on channel 1–13 (channel 14 at 2484 MHz is not working) and MCS 1–7 (corresponding to a single spatial stream). This way one can obtain a quick overview of the capabilities of new devices or changes introduced with new driver and kernel versions.

Table I
TESTED CHIPSETS AND DRIVERS
(COMPAT-DRIVERS-3.9-RC4-2, KERNEL 3.7-1-GRML-AMD64)

| Chipset | Driver | Comments |
|---|---|---|
| AR9282 | ath9k | adheres to DCF |
| AR9380 | ath9k | adheres to DCF, three spatial streams |
| AR9390 | ath9k | adheres to DCF, transmitter problems when QAM is being used |
| RT2870 | rt2800usb | violates DCF |
| RT3092 | rt2800pci | violates DCF |
| BCM43224 | brcmsmac | ignores DCF, no HT 40 |

## VI. CONCLUSION

In this paper we gave a comprehensive overview and experimental analysis of injection capabilities and MAC behavior of different IEEE 802.11 chipsets. To this end, we discussed the relevant details of media access procedures, namely the DCF, and derived a general upper bound for achievable injection rates under these conditions. In addition, we gave an estimate of how stations adhering to the DCF are affected when selfish stations deliberately choose smaller contention windows than allowed.

To enable features such as per-packet rate control, injection at MCS rates and in the 5 GHz band, a series of patches is necessary which we provide for download (see [1]). Injection and goodput rates were thoroughly analyzed and revealed considerable differences even between similar devices. These tests also revealed that three out of six devices violate or ignore the DCF, which results in unusual high injection rates but causes significant problems when stations are transmitting concurrently. This was confirmed by determining the distribution of contention windows per device by using hardware timestamps and by testing goodput with concurrent transmitters.

Low level access to IEEE 802.11 hardware such as monitor mode and frame injection has been around for a long time. However, without some minor patching of drivers injection is limited to legacy rates and might even not work as expected, e.g. drivers tend to ignore the important no-ACK radiotap option which avoids superfluous link-layer retransmits. In addition, devices may work differently than expected when operating in monitor mode. Knowing about these details may help both in planning testbeds and improvement of drivers.

To allow for quick and automatic testing of devices and their drivers we presented `moep80211eval`, an evaluation utility for IEEE 802.11 hardware in monitor mode based on our frame injection library `moep80211`.

Given the results presented in this paper, the next step is to evaluate hardware operating in IBSS and infrastructure mode as well as on different operating systems. First tests in infrastructure mode with the Ralink and Broadcom chipsets indicate similar MAC-layer misbehavior. It is interesting to note that these devices exhibit a different behavior compared to monitor mode and even show a dependency on which mode a device had been set to since the last reset cycle.

## REFERENCES


[1] S. Günther and M. Leclaire, "Supplemental material: driver patches for advanced frame injection and numerical data," http://moep80211.net/plink/noms2014.

[2] A. Zubow and R. Sombrutzki, "A Low-cost MIMO Mesh Testbed based on 802.11n," in *Wireless Communications and Networking Conference (WCNC), IEEE*, april 2012, pp. 3171–3176.

[3] S. Chachulski, M. Jennings, S. Katti, and D. Katabi, "Trading Structure for Randomness in Wireless Opportunistic Routing," 2007, pp. 169–180.

[4] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Médard, and J. Crowcroft, "XORs in the Air: Practical Wireless Network Coding," *IEEE/ACM Trans. Netw.*, vol. 16, no. 3, pp. 497–510, Jun. 2008.

[5] J. Vestin, P. Dely, A. Kassler, N. Bayer, H. Einsiedler, and C. Peylo, "CloudMAC: Towards Software Defined WLANs," vol. 16, no. 4, Instanbul, TR, 2012, pp. 42–45.

[6] P. Dely, J. Vestin, A. Kassler, N. Bayer, H. Einsiedler, and C. Peylo, "CloudMAC – An OpenFlow based Architecture for 802.11 MAC Layer Processing in the Cloud," in *Broadband Wireless Access Workshop (Globecom 2012 Workshops)*, Anaheim, USA, 2012, pp. 186–191.

[7] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "OpenFlow: Enabling Innovation in Campus Networks," *SIGCOMM Computation Communication Review*, vol. 38, no. 2, pp. 69–74, Mar. 2008.

[8] B. Chen and V. Muthukkumarasamy, "Denial of Service Attacks against 802.11 DCF," in *IADIS International Conference: Applied Computing 2006*, 2007.

[9] M. Malekzadeh, A. Azim, A. Ghani, J. Desa, and S. Subramaniam, "Empirical Analysis of Virtual Carrier Sense Flooding Attacks over Wireless Local Area Network," *Journal of Computer Science*, vol. 5, pp. 214–220, 2009.

[10] J. Soryal and T. Saadawi, "IEEE 802.11 Denial of Service Attack Detection in MANET," in *Wireless Telecommunications Symposium (WTS)*, april 2012, pp. 1 – 8.

[11] E. Kohler, R. Morris, B. Chen, J. Jannotti, and M. Kaashoek, "The Click Modular Router," *ACM Transactions on Computer Systems*, vol. 18, no. 3, pp. 263–297, Aug. 2000.

[12] B. Chambers, "The Grid Roofnet: A Rooftop Ad Hoc Wireless Network," M.Sc. Thesis, Massachusetts Institute of Technology, 2002.

[13] T. Ho and K. Chen, "Performance Analysis of IEEE 802.11 CSMA/CA Medium Access Control Protocol," in *Personal, Indoor and Mobile Radio Communications, 7th IEEE International Symposium on*, vol. 2, Oct. 1996, pp. 407–411.

[14] G. Bianchi, "Performance Analysis of the IEEE 802.11 Distributed Coordination Function," *Selected Areas in Communications, IEEE Journal on*, vol. 18, no. 3, pp. 535–547, 2000.

[15] E. Ziouva and T. Antonakopoulos, "CSMA/CA Performance under High Traffic Conditions: Throughput and Delay Analysis," *Computer Communications*, vol. 25, no. 3, pp. 313–321, Feb. 2002.

[16] L. Dai and X. Sun, "A Unified Analysis of IEEE 802.11 DCF Networks: Stability, Throughput and Delay," *Mobile Computing, IEEE Transactions on*, no. 99, p. 1, 2012.

[17] V. Shrivastava, S. Rayanchu, J. Yoonj, and S. Banerjee, "802.11n under the Microscope," in *Proceedings of the 8th ACM SIGCOMM Conference on Internet Measurement*, ser. IMC. New York, NY, USA: ACM, 2008, pp. 105–110.

[18] *Standard for Information Technology – Telecommunications and Information Exchange between Systems – Local and Metropolitan Area Networks – Specific Requirements – Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, IEEE, 2012.

[19] D. Young and M. Kershaw, "IEEE 802.11 radiotap," http://www.radiotap.org.

[20] J. Jun, P. Peddabachagari, and M. Sichitiu, "Theoretical Maximum Throughput of IEEE 802.11 and its Applications," in *International IEEE Symposium on Network Computing and Applications (NCA 2003)*, 2003, pp. 249–256.

[21] K. Youngsoo, C. Sunghyun, J. Kyunghun, and H. Hyosun, "Throughput enhancement of ieee 802.11 wlan via frame aggregation," in *Vehicular Technology Conference (VTC2004-Fall)*, vol. 4, 2004, pp. 3030–3034.

[22] P. Kyasanur and N. Vaidya, "Detection and Handling of MAC Layer Misbehavior in Wireless Networks," in *International Conference on Dependable Systems and Networks*, 2003, pp. 173–182.