Wireless LAN in Paired Radio Spectrum with Downlink-Uplink Separation

Stefan Schmid
Disney Research & ETH Zurich
Zurich, Switzerland

Stefan Mangold
Disney Research
Zurich, Switzerland

Thomas R. Gross
Dept. of Computer Science
ETH Zurich, Switzerland

***Abstract*—Wireless Local Area Networks (WLANs) based on the IEEE 802.11 standard apply a simple contention-based radio access protocol. Downlink communication from access points to mobile stations shares the radio channel with uplink communication from mobile stations to the access points. This protocol is due to the contention-based design that targets the operation in unlicensed spectrum. In the future, because of the growing demand for wireless communication services, WLANs might not only operate in unlicensed but also in licensed spectrum. However, licensed spectrum favors the use of separate (paired) radio channels for downlink and uplink communication – a setup that requires frequency-division-duplex communication. This paper describes and evaluates the feasibility of a WLAN system operating in paired spectrum with a proof of concept implementation. Our testbed employs off-the-shelf WLAN chips (two per device) and driver modifications that enable the system to operate with downlink-uplink separation while still maintaining the ability to function in unlicensed (single-channel) spectrum. We provide insights based on our testbed and evaluate the performance of our solution.**

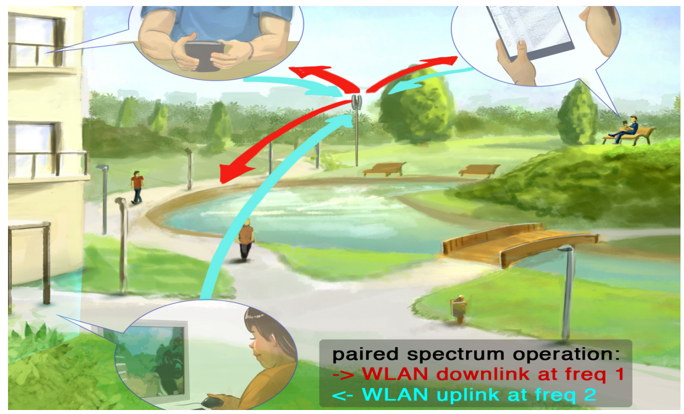

Fig. 1. Target scenario (© Disney): Stations operate with downlink-uplink separation with two WLAN modules per station (each operating on another frequency channel). Our driver modifications ensure that transmissions and receptions occur on different frequencies.

I. Introduction

The Wireless Local Area Network (WLAN) system was originally developed for data networks with coverage of less than 100 meters. WLAN applies a simple single-channel contention-based medium access protocol (Carrier Sense Multiple Access, CSMA) [1]–[3]. Communicating devices transmit and receive on the same shared radio channel. Channels for WLANs are available in the unlicensed 2.4 GHz ISM (Industry, Science, Medical) and the 5 GHz U-NII (Unlicensed National Information Infrastructure) radio spectrum, with additional channels available at 60 GHz and below 1 GHz. Over time, the WLAN standard has been enhanced towards support of new services including broadband access in rural areas or cellular data offloading. All these services need more channel capacity than available. Therefore it is beneficial to enable WLANs to operate in the licensed cellular spectrum, in addition to the unlicensed spectrum. However, a direct extension of the 802.11 WLAN standard to operate in cellular spectrum is difficult, due to regulatory requirements: Radio regulation for cellular networks favors paired spectrum (frequency division duplex with downlink-uplink separation) instead of a single radio channel for downlink and uplink (CSMA or time division duplex). For paired spectrum, downlink and uplink transmissions occur on different and strictly separated frequencies. The original WLAN protocol can therefore not directly be applied in paired spectrum: The CSMA protocol of WLANs must be modified so that it supports the separation of downlink and uplink data traffic. The following sections describe an 802.11-like protocol (based on [4]) that operates in paired spectrum and introduces new opportunities based on the fact that full-duplex communication is now possible. We describe the system architecture of a WLAN that can operate in (licensed) paired spectrum while maintaining the capability to operate in unlicensed (single-band) spectrum. All changes are limited to software only, so that off-the-shelf radios can be used. A testbed allows us to assess the necessary hardware and software modifications of standardized 802.11 radio devices. These modifications consist of (1) using two instead of one 802.11 chip per device for full duplex communication, and (2) driver modifications enabling contention-based paired spectrum operation. The evaluation results indicate that even with simple driver modifications, we can already implement a solid system that allows analyzing the new approach in great detail. This work is driven by our desire to improve radio spectrum efficiency. We want to show that WLAN can be used in paired spectrum as a proof of concept. The intention is not to promote or endorse the use of licensed and/or paired spectrum as different use cases or scenarios (e.g., nation-wide cellular networks, TV White Space spectrum [17], personal and local area networks) might require different trade-offs or be subject to a variety of constraints that cannot be discussed here due to lack of space.

978-1-4799-3083-8/14/$31.00 © 2014 IEEE

II. Protocol

All 802.11 transceivers are tuned to the same frequency channel to exchange messages. The 802.11 standard medium

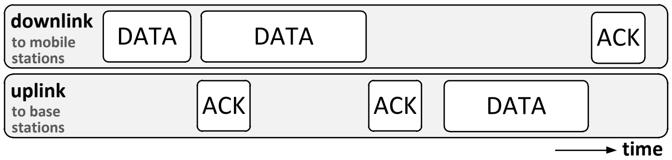

Fig. 2. Downlink / uplink communication using the modified protocol.

access is based on listen-before-talk: Before transmitting data, a station checks the channel by assessing the power levels and preambles. The data transmission can then continue only if the channel is determined to be unused (idle channel). Multiple stations might detect the idle channel at the same time and therefore start transmitting simultaneously, causing a collision. To mitigate the collision probability, 802.11 defines the Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) protocol [3], [5] that uses a contention window to randomize the time of transmission. An intended receiving station replies with an acknowledgment (ACK) to the source transmitter for every received packet. The ACK is immediately transmitted back after the packet has been correctly received. The source transmitter station waits for a time called ACK timeout and, in case of a missing ACK, retransmits the previous packet. The collision of an ACK and a data packet can be prevented by prioritising the ACK. To make sure that this is always the case, two parameters are introduced: the short inter-frame space (SIFS) and the distributed inter-frame space (DIFS). The SIFS varies from 10 to 20 microseconds for different standard versions, and the DIFS is computed as the sum of two slot times and SIFS. After a station receives a packet, it waits for SIFS and then sends the ACK. If a station wants to send a data packet and detects an idle channel, it waits for DIFS and a random number of slots in its contention window. This setup ensures that after receiving a data packet, the acknowledging station can always use the channel first.

A. Protocol with downlink-uplink separation

The new protocol design extends the 802.11 MAC protocol with an additional channel. A general use case is a base station communicating to multiple mobile stations. Each station uses two radio devices to send and receive on the two different channels at the same time. The base station uses the downlink channel to send data and control packets to the mobile stations. Mobile stations transmit data and control packets via the uplink channel back to the fixed station. Figure 2 shows an illustration of the protocol. We assume a cellular approach in which neighboring base stations operate on different radio channels. Since there is only limited interference between base stations operating on the same channel and therefore no contention or packet collision in downlink, it is possible

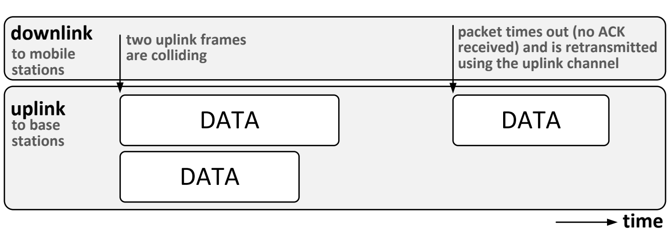

Fig. 3. Packet retransmission after collision in the uplink channel.

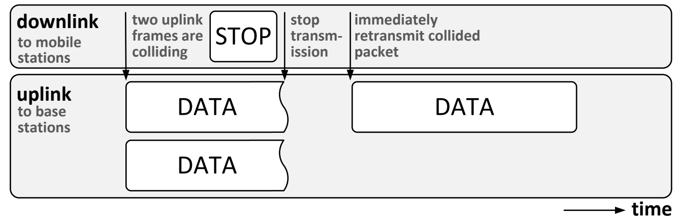

Fig. 4. The base station sends a stop packet to cancel transmissions.

to continuously send data to all the mobile stations without using the contention based protocol defined in 802.11 (i.e., no collision avoidance needed, contention window of size zero). There are no packet collisions possible but packets can still be lost due to an unfavorable signal-to-noise ratio at the receiver's end. Therefore the ACK control messages are still needed if one of the packets is lost in the downlink channel. These control packets are sent on the uplink channel and not on the downlink channel where the packet to be acknowledged has been received. The mobile stations use the same approach in the uplink channel.

B. Collision handling

Collisions are still possible in the uplink channel as shown in Figure 3. With an increasing number of stations, the probability of such a collision also increases. If the base station is able to detect this collision on the uplink channel (by simple noise detection), it can broadcast a STOP control packet. Since the mobile stations are listening on a channel separate from the channel used for sending, the mobile stations can receive this STOP packet and immediately interrupt the colliding transmission as illustrated in Figure 4.

C. Discussion

The new protocol was analyzed analytically and verified using simulation for scenarios consisting out of one base station and a varying number of mobile stations [4]. Using the proposed collision detection and STOP messages, the model shows that throughput loss with a high number of mobile stations can be decreased without affecting the downlink channel in a noticeable way. To assess the impact of the operating system of the stations, and to explore the timing dependencies between operating system actions and channel utilization, we built a testbed. As we want to maintain the ability to operate in unlicensed spectrum (i.e., the single channel setting used by today's WLAN installations), we focus on a software-only approach. A special-purpose hardware unit for paired channel WLAN may be attractive in future deployments – after we understand the interaction(s) between the operating system and the communication system.

III. Experimental Evaluation

The testbed consists of five mobile stations (the platform supports an arbitrary number of mobile stations), one monitor station (scanner in Fig. 5) and one base station. The mobile stations are equipped with two radios each connected to small omnidirectional antennas with 5 dB gain. The same hardware is used for the base station, but instead of the small omnidirectional antennas, two directional sector antennas are

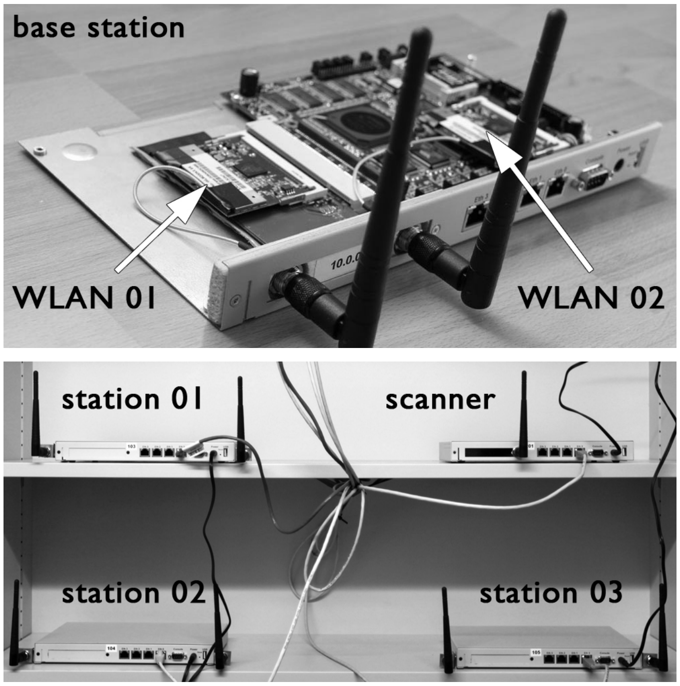

Fig. 5. Testbed: One base station and up to five stations are equipped with two WLAN modules each. The two modules operate on different frequency channels for every station. The driver software handles transmission and reception at the different modules to enable downlink-uplink separation.

connected to the station, each with 20 dB gain (not shown in Fig. 5). The WLAN chips employed typically operate with an on-board 44 MHz clock. With every tick of the clock, the cycle count register is incremented by one. Another register (called the busy register) is only incremented by one if the channel has been busy during the last tick. Busy means that the power level reached a certain threshold. In normal operation, this threshold is used to determine if the radio can receive a signal. Incrementing the busy register can also mean that during the last tick the radio was receiving a part of a packet. By reading out these registers continuously, a busy-idle pattern can be reconstructed. This pattern then allows to visualize the activity on the wireless channels. The testbed therefore also includes a scanning station running a sampling driver, which is used to visualize the protocol. The maximum sampling rate is around 0.3 MHz, which makes it possible to take a sample approximately every three microseconds, providing qualitative visualisation of the channel activity, as shown in Figs. 7, 8, 10 and 16. These figures show real measurements collected by the sampling driver. The black ticks indicate the samples. The two radio channels used for the measurements are in the upper 5 GHz band to reduce interference with other WLANs: Channel 149 (5745 MHz) and channel 165 (5825 MHz) are not occupied at the testbed's location in our lab and provide enough separation from the channels used by other networks. Fig. 6 shows the spectrum measured between the two antennas during data exchange. The interference created by the antennas' nearfield for the used channels (149 and 165) is clearly visible. To remove this interfering effect from the measurements, we placed the two antennas as far as possible, about 30 cm from each other. This self-interference can be reduced without physical separation as shown in [6], [7] by implementing a hardware solution. For our testbed, separat-

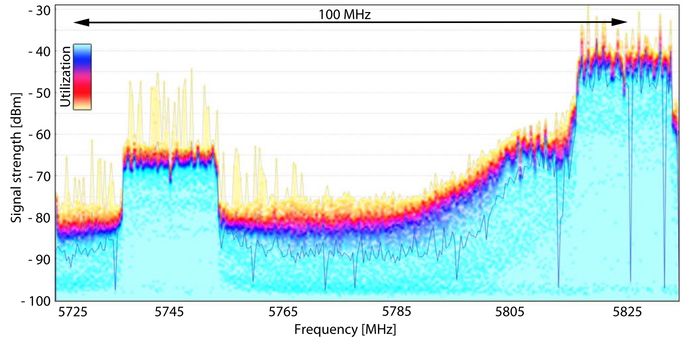

Fig. 6. Frequency spectrum for the uplink and downlink (channel 149 and 165) measured between the antennas of a station during data transmission.

ing the antennas provided us with stable measurements. The testbed's performance is measured using data throughput in byte per second (B/s). The data throughput is computed as the sum of the successfully received data packets' payload sizes. Only the usable payload is measured, without the protocol headers. The packets are generated at the application layer of the sending station (source) and collected at the receiving station's application layer (destination). Since the evaluation is done at the application layer, possible retransmissions by lower layers (MAC, TCP) will not perturb the (end-to-end) measurements. Only successfully received packets are counted.

A. Single channel performance and optimal timeout for the software ACK

Fig. 7 shows a snapshot of the channel while transmitting an UDP data stream. The data is sent from the base station to the mobile station using the existing hardware ACK implementation. The figure shows that the ACK arrives approximately after a SIFS, which is ten microseconds in the used settings. Fig. 8 shows a channel snapshot of an UDP data stream using the software ACK implementation. The transmission of the software ACK is started after 100 microseconds. On receiving a DATA packet, the driver issues an interrupt and starts a handler routine to create an ACK and to finally send it back to the packet's originator. These steps take significantly more time than the hardware implementation and the ACK timeout needs to be adjusted. The generation and recognition of an ACK in software takes about 150 microseconds. To find an optimal setting for the ACK timeout, the timeout value is varied from 50 microseconds to 350 microseconds

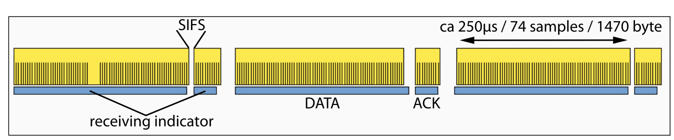

Fig. 7. Captured single channel UDP data stream (packet payload 1470 byte, no contention window).

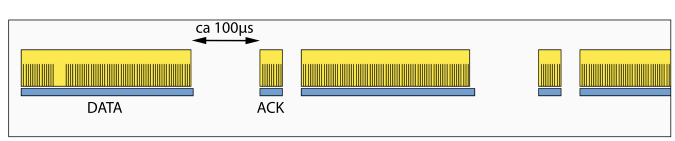

Fig. 8. Captured single channel UDP data stream with software generated ACK (packet payload 1470 byte, no contention window).

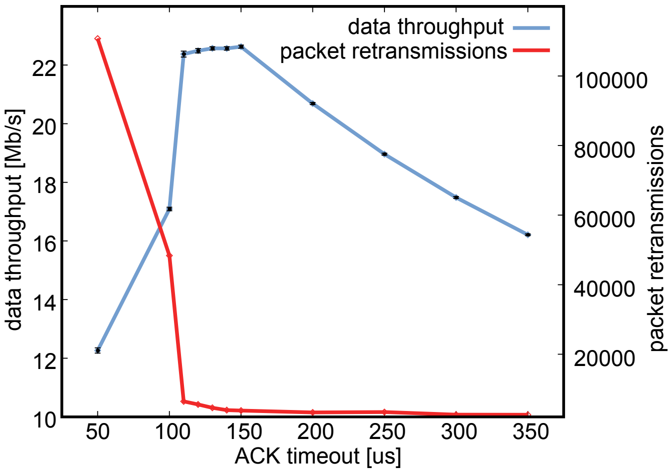

Fig. 9. Throughput and retransmission vs. software ACK timeout.

as shown in Fig. 9. The plot shows that for a delay smaller than 110 microseconds, the ACK does not arrive in time, and as result, packets are retransmitted. The figure also shows that the packet retransmission mechanism implemented in software works. From 110 to 150 microseconds, the throughput stays approximately the same but the retransmissions can still slightly be reduced. For higher ACK timeout values, the throughput decreases again because fewer packets are transmitted. The measurements show that an optimal setting for the ACK timeout is at 150 microseconds.

B. Dual channel performance of the software ACK

Fig. 10 shows a snapshot of dual channel data traffic. We observe that using two channels decreased the ACK delay (compared to Fig. 8) but it takes significantly more time to deliver the next packet. In single channel mode, it is possible to still use the hardware retransmissions since a software generated ACK looks like a hardware ACK. Now different modules handle receiving and sending, so incoming ACKs cannot stop the hardware retransmissions anymore. Hence, the hardware retransmission must be disabled and packets are resent using the driver. This implementation disables and enables the software packet queue. To send the next DATA packet, the queue is re-enabled after receiving an ACK, and these actions lead to the 200 microseconds delay.

C. Throughput and packet loss in paired spectrum

Fig. 11 illustrates the measured paired spectrum throughput with one transmitter and one receiver and constant packet sizes. The graphs confirm the expected throughput results that are typical for contention-based protocols: With increasing offer,

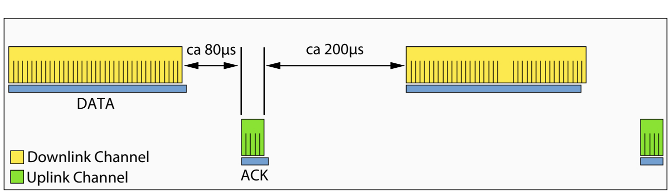

Fig. 10. Captured paired channel UDP data stream. (packet payload 1470 byte, no contention window).

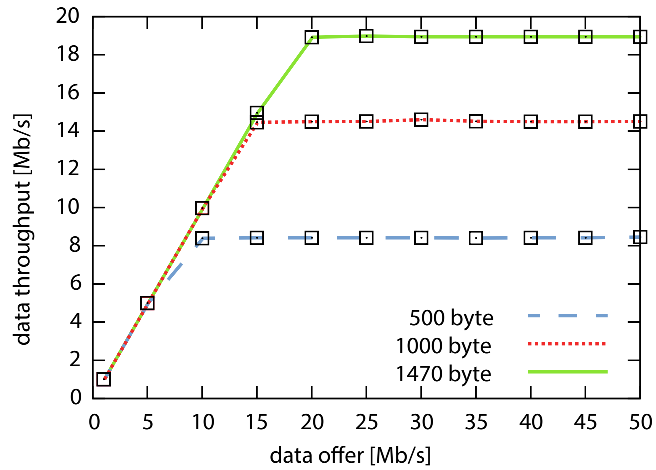

Fig. 11. Achievable data throughput for different packet sizes.

the overall throughput increases until it reaches the saturation point. This maximum throughput depends on the packet size: Similar to 802.11, the contention-based paired spectrum protocol observes higher throughput with larger packet sizes.

Fig. 12 shows UPD data throughput for one to five stations for different packet sizes. The mobile stations are all sending to one base station, which acknowledges the incoming packets. Standard contention window settings are used. The graph shows that larger packets lead to higher throughput. The ACK and queue handling, which is done in software for the paired-spectrum driver, introduces a lot of channel idle time. For larger packets, more data gets through per packet and as a result there is higher throughput. Multiple stations sending in parallel also result in higher throughput because a station can transmit while another station is trapped in the driver. This feature fills up some holes where the channel is idle. The plot shows also the typical behavior of multiple station saturating a channel.

Fig. 13 compares the throughput for multiple stations with standard contention window settings to a scenario with a disabled contention window. For one to three stations the throughput of the zero contention window is higher because

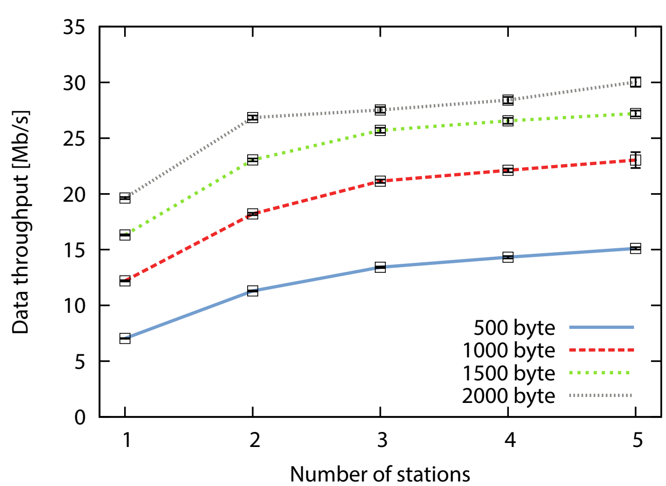

Fig. 12. UDP data throughput.

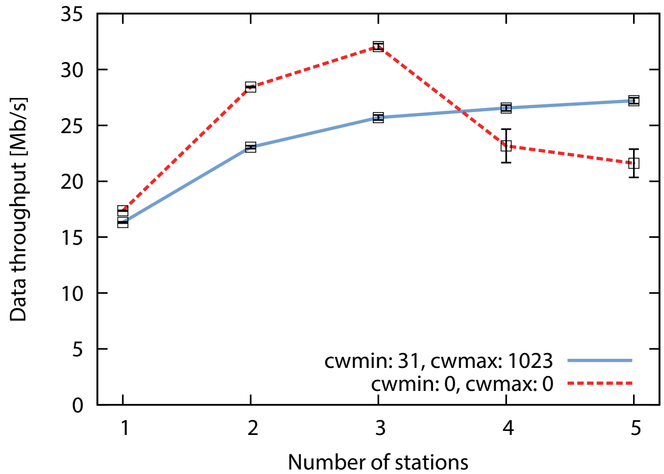

Fig. 13. UDP data throughput with disabled contention window.

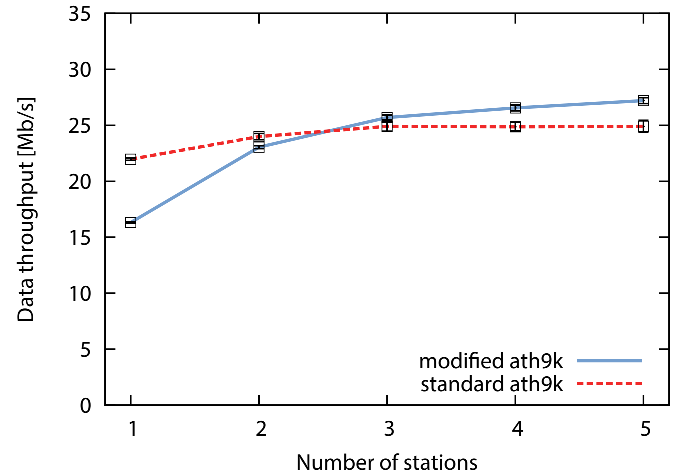

Fig. 15. Comparison between an unmodified driver and the paired-spectrum version for a packet size of 1500 byte.

there is less channel idle time, and the probability of a collision (normally prevented by a contention window) is small. For a higher number of stations, however, the probability of a collision increases and therefore more packets are lost. The throughput performance decreases compared to the setting with enabled contention window.

Fig. 14 shows the packet loss for the throughput measurements described in the last two paragraphs. The packet retransmission on the MAC layer is disabled for these measurements. The plots highlight that the modified contention based protocol is still working. Disabling the contention windows introduces high packet loss (four times higher compared to standard settings) due to collisions. The colored segments of the bars and their height show the packet lost distribution per station. The almost equal sizes of the shares point out the fairness between the stations for successful packet delivery.

Fig. 15 compares the throughput of the unmodified driver (standard 802.11) to the paired-spectrum driver. For a small number of stations, the standard driver performs better than

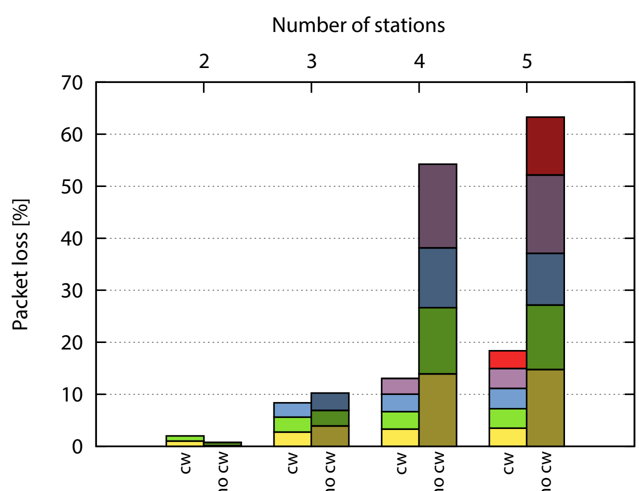

Fig. 14. Packet loss for different number of stations and packet size 1500 byte for enabled (cw) and disabled (no cw) contention window. The height of the different colored segments indicates the packet loss distribution per station.

our solution due to its hardware ACK generation and queue handling. For a higher number of stations, the throughput of the paired-spectrum driver is slightly higher. This result is partially due to the increased channel capacity (ACKs transmitted on different channel). Another reason for the better performance of the new driver is that it does not handle packet retransmission whereas the standard driver recovers from collisions. The performance of the testbed system should not be compared directly to the performance of optimized hardware systems. But these measurements show that WLAN in paired spectrum behaves similar to standard WLAN, and a hardware implementation of the protocol is feasible for future studies.

D. *How to detect collisions*

The used WLAN platform supports a counting register that only counts if the chip is in receiving mode. The register starts counting with the begin of the preamble. If the preamble is decoded successfully, the register continues counting, if the device fails to recognize the preamble correctly, the receiving register stops counting. The decoding failure can have several reasons, but most of the time it is due to packet collision. Hence, this register can be used as an indicator for packet collision. A base station can be equipped with this software and issue STOP packets according to the receiving register state. Fig. 16 shows a real packet stream with collisions visualized by the sampling driver. It is also obvious that if a packet was not received correctly (no receiving indicator) no ACK was generated. The STOP packet has not been integrated into the testbed. A software-based system is not fast enough to detect a collision, issue a STOP packet and recognize it on the receiving side within the transmission duration. If a hardware implementation of the described technology is used instead, the system could react within microseconds and successfully cancel ongoing transmissions.

IV. RELATED WORK

The original 802.11 standard is based on CSMA, which is an extension of the slotted Aloha protocol. The world's first packet radio network, AlohaNet in Hawaii, was based on slotted Aloha [8]. At that time, transceivers were simple and not able to switch between transmit and receive in a short time interval. The AlohaNet in fact used two radios per location, one for uplink and one for downlink, similar to our approach. IEEE 802.11 is an established technology for wireless communication that has been subject to intensive academic research throughout the last decades. Much work has been done in the area of medium access protocols and spectrum management [2], [5], [9]. A special focus has been around multi-radio (multi-channel) architectures, for example in [10]–[12]. Our work differs from the state-of-art by not only introducing two or more channels for parallel operation with multiple WLAN transceivers (two in our case), but by separating uplink and downlink in addition so that WLAN operates in the spectrum like a cellular system (e.g., transmitting DATA and ACK at different radio channels). The concept of FDD is as old as cellular networks but is still subject to research, mainly with focus on spectral coexistence, see for example [13], [14]. Contrary to our work, in these papers, FDD systems do not operate with the CSMA medium access protocols, instead with deterministic and centralized protocols. The basic idea of modifying WLAN towards FDD support was expressed in [9]. The work in [4] contains an early simulation study of this idea suggesting the feasibility of the paired-spectrum approach for CSMA. We used off-the-shelf WLAN hardware, which operates in the unlicensed ISM band (2.4/5 GHz), to create our testbed. From there it is possible to use a band converter [15] or change the radio front-end of the WLAN hardware to move the operating spectrum of the testbed to a licensed band. The testbed uses two WLAN modules per station. Recent work [6], [16] shows that it is possible to send and receive with the same radio front-end using different spectrum slices but these approaches employadditional hardware or extensive hardware changes whereas this protocol relies on software changes only. We also propose collision detection/notification by introducing a STOP packet, which has been studied in [7] and implemented by using software defined radios. We do not investigate such an implementation, but provide all the building blocks to construct such a system.

V. Conclusions

This paper presents a protocol for a WLAN system operating in paired spectrum employing the well-established contention-based 802.11 protocol. A working (Linux-based) system can be realized using off-the-shelf WLAN chips (two per device) and a modified version of the Linux wireless drivers. With the driver modification, a wireless network is realized that can conform with radio regulation of much of today's radio spectrum, including unlicensed spectrum and licensed cellular paired spectrum. The protocol described here has been implemented as a proof of concept for a testbed and delivers performance that is competitive with a (standard) WLAN system. With the presented architecture in place, further (hardware) optimization can be explored, and the system can easily be replicated and evaluated in large-scale scenarios. It is an affordable platform to analyze a wide range of new research problems related to reliability, quality-of-service support and scalability.

The approach presented here demonstrates the potential benefits of a solution that is based on software-only changes. WLAN devices are widely used consumer products – software modification can enable these WLAN devices to meet the regulatory requirements of new, so far not foreseen frequencies, and would allow use of such devices in such spectrum with just a minimal effort. As there are evolving scenarios that favor paired spectrum (in some countries the spectrum of wide area and rural area networks is regulated as paired spectrum, TV White Space spectrum that is regulated differently for downlink and uplink), we expect increased interest in a communication platform that can operate in both single-band and dual-band scenarios.

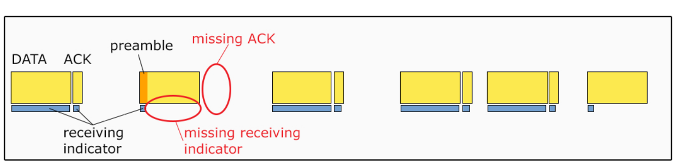

Fig. 16. Collision detection visualized with the sampling driver.

References

[1] IEEE, "P802.11-2007, IEEE Standard for Wireless LAN Media Access Control (MAC) and Physical Layer (PHY) Specifications," Nov. 2007.

[2] B. Walke, S. Mangold, and L. Berlemann, *IEEE 802 Wireless Systems: Protocols, Multi-Hop Mesh/Relaying, Performance and Spectrum Coexistence*. John Wiley & Sons, Nov. 2006.

[3] B. O'Hara and A. Petrick, *The IEEE 802.11 Handbook: A Designer's Companion*. Standards Information Network IEEE Press, 1999.

[4] S. Mangold and T. Gross, "A dual-radio contention-based protocol for paired spectrum access and TV white space," in *Wireless On-demand Network Systems and Services (WONS) 2012*, Jan. 2012, pp. 107–112.

[5] S. Mangold *et al.*, "Analysis of IEEE 802.11e for QoS support in wireless LANs," *Wireless Communications, IEEE*, vol. 10, no. 6, pp. 40–50, Dec. 2003.

[6] K. Chintalapudi *et al.*, "WiFi-NC: WiFi over narrow channels," in *Conference on Networked Systems Design and Implementation (NSDI) 2012*. Berkeley, CA, USA: USENIX Association, 2012.

[7] S. Sen, R. Roy Choudhury, and S. Nelakuditi, "CSMA/CN: carrier sense multiple access with collision notification," in *MobiCom 2010*. New York, NY, USA: ACM, 2010, pp. 25–36.

[8] M. Schwartz and N. Abramson, "The AlohaNet - surfing for wireless data [History of Communications]," *Communications Magazine, IEEE*, vol. 47, no. 12, pp. 21–25, Dec. 2009.

[9] L. Berlemann and S. Mangold, *Cognitive Radio and Dynamic Spectrum Access*. John Wiley and Sons, 2009.

[10] F. Dreier, V. Vukadinovic, and S. Mangold, "Performance of Dual Wi-Fi Radios in Infrastructure-Supported Multi-Hop Networks," in *Mobile Adhoc and Sensor Systems (MASS) 2011*, 2011, pp. 739–745.

[11] F. Chen, H. Zhai, and Y. Fang, "An opportunistic multiradio MAC protocol in multirate wireless ad hoc networks," *Wireless Communications, IEEE Transactions on*, vol. 8, no. 5, pp. 2642–2651, 2009.

[12] B. Ko and V. Misra, "Distributed channel assignment in multi-radio 802.11 mesh networks," in *WCNC 2007*, 2007, pp. 3978–3983.

[13] A. Otyakmaz *et al.*, "Parallel operation of half- and full-duplex FDD in future multi-hop mobile radio networks," in *Proceedings of European Wireless Conference 2008*, vol. 1, no. 1, June 2008, pp. 1–7.

[14] P. Chan *et al.*, "The evolution path of 4G networks: FDD or TDD?" *Communications Magazine, IEEE*, vol. 44, no. 12, pp. 42–50, 2006.

[15] P. Bahl *et al.*, "White space networking with Wi-Fi like connectivity," in *SIGCOMM 2009*, 2009, pp. 27–38.

[16] S. S. Hong, J. Mehlman, and S. Katti, "Picasso: flexible RF and spectrum slicing," in *SIGCOMM 2012*. New York, NY, USA: ACM, pp. 37–48.

[17] FCC, "FCC 08-260 (2008) Federal Communications Commission, Second Report and Order and Memorandum Opinion and Order." Nov. 2008.