Experimental Evaluation of a Robust MAC Protocol for Network Coded Two-Way Relaying

Sebastian Bittl, Christoph Hausl, Onurcan Işcan

Experimental Evaluation of a Robust MAC Protocol for Network Coded Two-Way Relaying

Sebastian Bittl, Christoph Hausl and Onurcan İşcan

Institute for Communications Engineering
Technische Universität München
Theresienstraße 90, 80333 Munich, Germany
sebastian.bittl@mytum.de,christoph.hausl@tum.de,onurcan.iscan@tum.de

Abstract. Wireless half-duplex relay communication between two nodes is considered. A two-way decode-and-forward relaying strategy that uses network coding at the relay should be able to increase the data throughput. A specific medium access (MAC) protocol based on a TDD/TDMA scheme is proposed that establishes robust synchronization between the terminals. An experimental evaluation of the proposed MAC protocol is performed using a software-defined radio system consisting of a terminal for each node in the network. It is shown that the proposed protocol realizes the promised throughput-gain of network coding for large burst-lengths. Moreover, the additional amount of processing time, memory and signalling required due to network coding is described.

Keywords: two-way relaying, network coding, software-defined radio

1 Introduction

The usage of relays bears good prospects for improving current wireless communication systems like sensor, ad-hoc or cellular networks. In this work a wireless bidirectional communication between two nodes A and B, which is supported by a half-duplex relay R, is considered. Such a relay is not able to receive and transmit simultaneously in the same frequency band. A decode-and-forward scheme is used at the relay, which only forwards data that was received correctly. To increase the achievable data rate network coding [1] is performed at the relay.

Conventional two-way relaying (without network coding) uses a communication scheme with four phases within one cycle to transmit a data packet a, from node A to B, as well as a data packet b from node B to A (Figure 1(a)). A scheme exploiting network coding needs only three phases within one cycle to achieve the same result [2–4]. Thereby, network coding at the relay R is applied. This is done by performing a bitwise binary XOR (exclusive-or) operation between the contents of packets a and b. This is illustrated in Figure 1(b).

In this work we propose a medium access (MAC) protocol that allows to realize network-coded two-way relaying with a robust synchronization between the terminals. In our protocol the relay acts as master whereas the terminals act

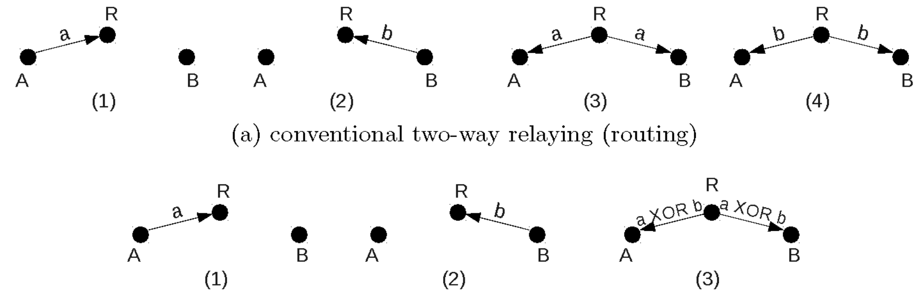


Fig. 1. Two-way relaying with and without network coding at the relay.

as slaves that are allowed to transmit only upon a pull command from the relay. We show that the proposed protocol realizes the promised throughput gain 4/3 of network coding, if the bit-length of the data burst is much larger than the one of the pull command. For short data bursts the gain of network coding decreases because network coding does not decrease the number of pull commands within one cycle. We also describe the additional amount of memory required due to network coding at each terminal to store its own packets. We evaluate the performance of the proposed protocol in an experiment with the software defined radio platform USRP (universal software radio peripheral) [5]. Although the use of pull commands is less bandwidth-efficient than a synchronization on the preamble of the relay transmission, it is a well-suited approach synchronizing software defined radio systems because it provides robustness against variations in the signal processing delay at the terminals [6].

In the following paragraph we describe related work. A system using network coding for two-way relaying with an amplify-and-forward scheme is described in [7], leading to so-called analog network coding. The relay in [7] also uses a pull command to synchronize the nodes before sending, whereas the two terminals are pulled one after the other in our work and not simultaneously as in [7]. Reference [8] proposes a TDMA scheme for implementing two-way decode-and-forward relaying with network coding applied to Bluetooth and suggests an implementation using a software defined radio platform, but does not provide any deeper analysis or experimental evaluation. Software-defined radio experiments are applied for the evaluation of relaying systems without network coding in [10–15]. For example, [15] evaluates a decode-and-forward relaying system which includes three terminals. This system does not use network coding and overcomes the issue of synchronisation between the different terminals by controlling all of them by a single computer. In contrast, our system uses a separate computer for controlling each terminal, imposing full physical separation, and therefore makes use of pull commands as mentioned before.

This paper is organised as follows. Section 2 describes the used medium access (MAC) protocol. Its evaluation is given in section 3. Conclusions and possible directions for future work are provided in section 4.

2 Medium Access Protocol

The proposed medium access (MAC) protocol is based on a time division duplex (TDD) / time division multiple access (TDMA) scheme. This means only one frequency band is used for all the transmissions in the network.

2.1 Relay Transmission

The relay transmission contains a burst of $N > 0$ packets whereas each packet contains a header of $H = 6$ bytes and a payload of D bytes with $0 \leq D \leq 255$ and $L = H + D$. The relay, which acts as master, initializes the communication. Each header contains two pull-bits p_{A} and p_{B} whereas $p_i = 1$ signalizes that terminal i is supposed to transmit immediately after the current relay transmission. The combination $p_{\mathrm{A}} = 0, p_{\mathrm{B}} = 0$ is used to stop the communication. The combination $p_{\mathrm{A}} = 1, p_{\mathrm{B}} = 1$ is never used. Additionally, each header includes two bits s_{A} and s_{B} that indicate the source of the payload data whereas the combination $s_{\mathrm{A}} = 1, s_{\mathrm{B}} = 1$ indicates the use of network coding. In case no network coding is used, only one of the bits s_{A} and s_{B} ist set to '1'. Network coding is not used in case one of the two buffers at the relay corresponding to A and B is empty. This occures in case of asymmetric packet loss on the links between relay and terminals. Then, the relay just sends the remaining data from the nonempty buffer. Moreover, it is possible to switch off network coding to evaluate a reference system without network coding. Another bit specifies whether the packet was sent by the relay. This enables the nodes A and B to distinguish whether a received packet includes data destined for them. Moreover, the header includes sequence counters (two bytes for each terminal) in order to detect packet losses by discontinuities in the received sequence counters. The header includes also one byte that signalizes the length D of the payload. In case network coding is used, the length-information in the header consists of the xor-ed length-information of the combined packets a and b. The other parts of the header are not affected by network coding. The relay releases the wireless channel by setting a *last in burst* bit in the header of the last packet sent in a burst. The remaining two bits in the header are currently not used.

We denote bursts with $N = 1$ and $D = 0$ as pull transmissions whereas other bursts are denoted as data transmissions. For example, the initial relay transmission is a pull transmission, because no data is available at the relay at this time. Later the relay includes xor-ed packets from its buffer in its transmission. We use four cyclic redundancy check (CRC) bytes in a packet for the header and the payload to allow error detection.

The communication over a noisy channel requires the ability to recover from connection losses due to errors, which lead to a loss of synchronization between the stations in the network. In order to cope with this issue, the relay uses a timeout to supervise whether the connection is still alive. That means the relay tries to reinitialize the communication in case the terminals do not respond.

2.2 Terminal Transmission

The terminals which act as slaves initially listen to the channel. As soon as a terminal successfully receives a packet from the relay whose header indicates that network coding was used, it network-decodes the data whereas the packet ID in the header is used to identify the required own packet in the buffer. The transmission of the terminals have the same structure as the relay-transmission. A terminal transmits a new packet if the relevant pull-bit is set to '1' in the received transmission. A transmitted packet is stored at the terminal until it or a subsequent packet is identified in the header of the relay transmission.

2.3 MAC Protocol without Network Coding

In this section, we consider the system when network coding is disabled. The access scheme is depicted in Figure 2. The the relay (R) lets the nodes A and B access the wireless channel alternately. The shaded box represents a pull transmission whereas the white boxes represent data transmissions. This scenario corresponds to the one shown in Figure 1(a). Besides the initialization, four

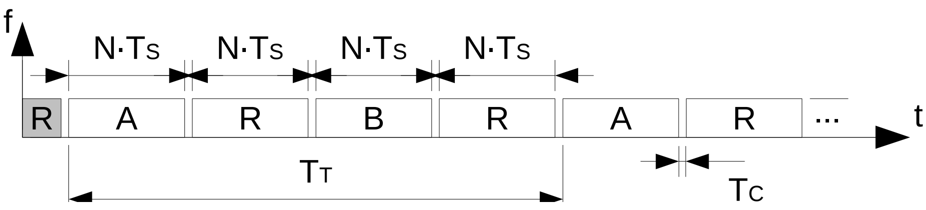


Fig. 2. Time allocation for two-way relaying with routing.

data transmissions are used within one cycle, and thus the overall cycle time T_T is given by

$$T_T = N \cdot 4 \cdot T_S + 4 \cdot T_C(N) + 4 \cdot T_P \ , \tag{1}$$

where T_S is the time it takes to send a data packet, $T_C(N)$ denotes the necessary processor calculation time at a station and T_P is the propagation time it takes the electromagnetic wave to arrive at the receiver after its transmission by the sender. Sending a pull transmission takes the time T_R.

No general formula can be provided for $T_C(N)$ as this time depends on the used system and on how this value changes when N is modified. It can be assumed that for increasing values of N the values of $T_C(N)$ also increase because more data has to be processed at the stations during each time slot.

Using Equation 1 together with the amount of data (in byte) L included in a packet, the number of header fields (in byte) H required by the communication protocols and the number of packets transmitted in each burst N leads to

$$R_{NoNC} = \frac{N \cdot (L - H)}{T_T} \ , \tag{2}$$

which determines the achievable data rate R_{NoNC} at each node.

2.4 MAC Protocol with Network Coding

Figure 3 displays the time allocation in case network coding is performed at the relay by using an XOR operation between the data received from nodes A and B in the first two time steps (see also Figure 1(b)). Besides the initialization,

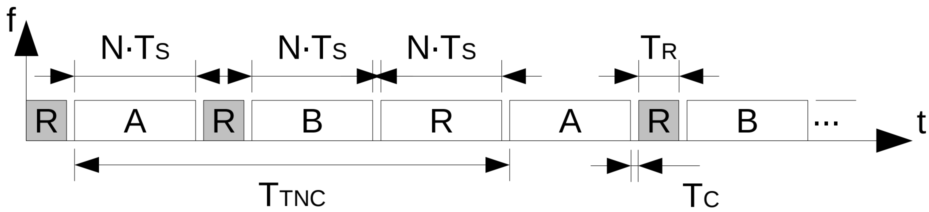


Fig. 3. Time allocation for two-way relaying with network coding.

three data transmissions (white boxes) and one pull transmission (shaded box) are used within one cycle and thus, the overall cycle time T_{TNC} is given by

$$T_{TNC} = N \cdot 3 \cdot T_S + 4 \cdot T_C(N) + 4 \cdot T_P + T_R \; . \tag{3}$$

The difference between the cycle times of the cases with and without network coding can be calculated as

$$\Delta T = T_T - T_{TNC} = N \cdot T_S - T_R \; . \tag{4}$$

In analogy to the case without network coding the achievable data rate R_{NC} for relaying with network coding is given by

$$R_{NC} = \frac{N \cdot (L - H)}{T_{TNC}} \; . \tag{5}$$

The ratio between the achievable data rates for two-way relaying with and without network coding at the relay is given as

$$G_T = \frac{R_{NC}}{R_{NoNC}} = \frac{T_T}{T_{TNC}} = \frac{T_T}{T_T - \Delta T} = \frac{1}{1 - \frac{N \cdot T_S - T_R}{N \cdot 4 \cdot T_S + 4 \cdot T_C(N) + 4 \cdot T_P}} \; . \tag{6}$$

Under the assumption that processing time $T_C(N)$ is negligible compared to the transmission time $N \cdot T_S$ Equation 6 can be simplified to

$$\lim_{N \to \infty} G_T = \frac{1}{1 - \frac{1}{4}} = \frac{4}{3} \; , \tag{7}$$

providing an upper bound on the ratio G_T between R_{NC} and R_{NoNC}. How well the measured ratio can approach this bound in a real system depends mostly on

$T_C(N)$. It also scales with varying N as $N \cdot T_S$ does, while the other components of T_T are constant.

The required memory m in bytes at each terminal to store its own packets is $m = N \cdot L$. Contrary to other possible protocols, it is not necessary to store several bursts, because the relay and terminals stop after their own transmission and wait until the other's transmission is finished. This stop-and-wait behavior makes the throughput R_{NC} decrease with growing propagation time T_P. The system delay $T_{TNC} - N \cdot T_S$ increases with growing burst size N.

In the following section an experimental evaluation of the proposed MAC protocol with software-defined radio is described.

3 Experimental Evaluation using Software-Defined Radio

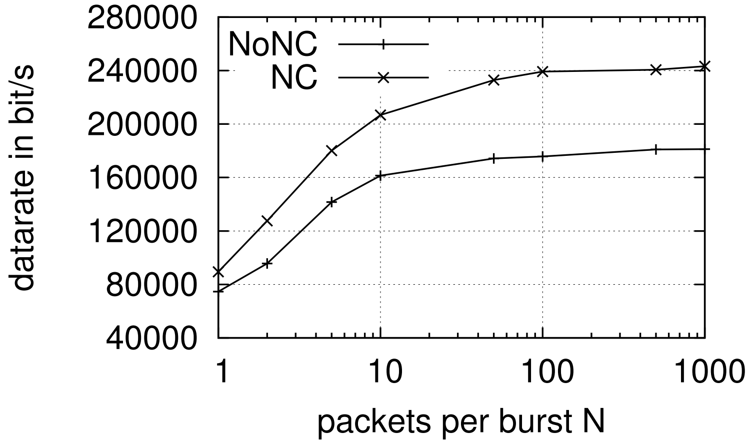
The above described MAC protocol was evaluated using a network consisting of three SDR-terminals. Thereby, a combination of the software GNU Radio in version 3.2.2 [16] and an USRP of version 1 [5] equipped with a XCVR2450 daughterboard and a VERT2450 antenna was used. The used software is available online at *The Comprehensive GNU Radio Network (CGRAN)*[1]. During experiments the distance between two stations was $d = 3$ meters. They were placed at the corners of a equilateral triangular. The physical layer achieves a gross data rate of $R_g = 1024$ kbit/s and uses Reed-Solomon Codes [17] to protect the pull and data transmissions with code rates of $\frac{2}{3}$ and 0.87, respectively.

Figure 4(a) depicts the measured data rate R_{NC} of two-way relaying with network coding (NC) at a node and the measured data rate R_{NoNC} of two-way relaying without network coding (NoNC). They clearly show the throughput gain of network coding. Fig. 4(b) depicts the relative gain $G_T = \frac{R_{NC}}{R_{TW}}$ due to network coding. The gain increases with growing N and achieves asymptotically the theoretical gain of 4/3. The gain of network coding decreases for small N, because network coding does not allow to decrease the number of pull commands within one cycle.

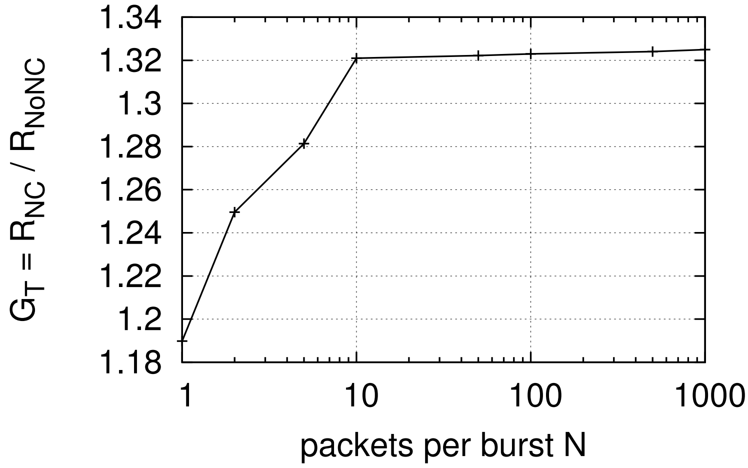
Besides the data rate, we also determined the relation between the processing time T_C and the number of packets sent per burst N. The data rate was measured during the experiment for different values of N and the corresponding T_C was calculated from Equations 2 and 5. This is possible because the values of $L = 255$ and $H = 6$ bytes as well as the gross data rate R_g are adjusted by the user and thereby known a priori. Therefore, T_R and T_S can be easily calculated.The delay T_P is known from the distance of the stations d and determined by $T_P = \frac{d}{c}$ whereby c is the speed of light).

Furthermore, it was investigated whether the cycle time is stable or suffers from fluctuations. An experimental measurement for $N = 1$ of a point-to-point communication where the relay always pulls the same node and does not forward the received data was done. For this setup the cycle time T_T was measured 1000 times. The results of this measurement and the calculated values of the known parts of T_T for $N = 1$ are given in Table 1. The standard deviation of the cycle

[1] https://www.cgran.org/wiki/RelayingSchemesImplementation



(a) Measured data rates at a node



(b) Ratio of data rates at a node

Fig. 4. Dependency of throughput (gain) on packets per burst N

Table 1. Cycle time T_T, its parts T_R, T_S, T_P, T_C and its variation in seconds.

T_R	T_S	T_P	T_C	T_T	StD(T_T)
$1.6 \cdot 10^{-4}$	$2.0 \cdot 10^{-3}$	$10 \cdot 10^{-8}$	$5.5 \cdot 10^{-3}$	$7.7 \cdot 10^{-3}$	$9.1 \cdot 10^{-5}$

time is approximately 1.2% of the average cycle time. Additionally, the value of T_P is small compared to the other values contributing to T_T such that its influence is negligible. Furthermore, T_R as part of the overhead consisting of T_R, T_P and T_C (see also Equation 3) is more than 10 times smaller than the time used to transmit data T_S. The influence of T_R on T_T will further decrease when N increases, as it stays constant while $N \cdot T_S$ and $T_C(N)$ increase (see Equation 3).

The measurement results of T_C as a function of N are given in Figure 5. The

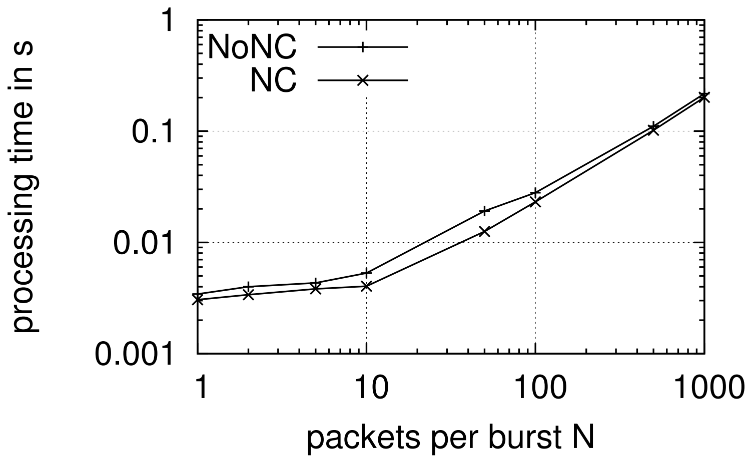


Fig. 5. Dependency of processing time T_C on packets per burst N

measured data rates corresponding to the values used for Figure 5 are provided in Figure 4(a). From Figure 5 one can see that T_C is almost constant in the area of $1 \leq N \leq 10$ and above a value of $N = 10$ increases linearly with increasing values of N. The gradient of the graph is thereby slightly below one.

This result shows that in the range from $1 \leq N \leq 10$ it is clearly better to go for higher values of N, as the transmission time $N \cdot T_S$ increases linearly with rising N while $T_C(N)$ stays constant and therefore the ratio between useful time $N \cdot T_S$ and overhead $T_C(N)$ drops significantly. For $N = 10$, $T_C(10)$ is just about $\frac{1}{5}$ of $10 \cdot T_S$, while for $N = 1$, $T_C(1)$ is more than twice T_S.

From $N = 10$ on, the value of $T_C(N)$ increases almost linearly with rising values of N. As the gradient is less than one, the ratio between $N \cdot T_S$ and $T_C(N)$ decreases further but much slower than in the range of $1 \leq N \leq 10$.

4 Conclusion and Future Work

The proposed MAC protocol enables a software-defined radio implementation of a two-way decode-and-forward relaying scheme with usage of network coding at a half-duplex relay. It achieves the promised throughput gain of network coding for large burst-lengths. The protocol achieves frame-level synchronisation by using pull-commands, which are either transmitted in a separate pulling packet or are part of the control information included in a data packet. Measurements in a real system show that the computational effort caused by network coding only limits the actual data rate by a small amount.

Future work includes expansion of the created system to exploit the direct link between the two terminals in order to gain diversity [18]. This does not require any changes of the proposed MAC protocol.

5 Acknowledgments

We thank Prof. Gerhard Kramer for helpful comments. The work was partly supported by the Alexander von Humboldt-Foundation and the German Ministry of Education and Research.

References

1. Ahlswede, R., Cai, N., Li, S.Y.R., Yeung, R.W.: Network Information Flow. IEEE Trans. on Inform. Theory 46(4), 1204–1216 (July 2000)
2. Yeung, R.W., Li, S.-Y.R., Cai, N., Zhang, Z.: Network Coding Theory, Part I: Single Source. NOW
3. Wu, Y., Chou, P.A., Kung, S.-Y.: Information Exchange in Wireless Networks with Network Coding and Physical-Layer Broadcast. In: Conf. on Information Sciences and Systems (March 2005)
4. Larsson, P., Johansson, N., Sunell, K.-E.: Coded bi-directional relaying. In: Presentation at 5th Swedish Workshop on Wireless Ad-hoc Networks, Stockholm, Sweden (May 2005)
5. Universal Software Radio Peripheral, `http://www.ettus.com/downloads/ettus_ds_usrp_v7.pdf`, data sheet of Ettus Research LLC
6. Dhar, R., George, G., Malani, A., Steenkiste, P.: Supporting Integrated MAC and PHY Software Development for the USRP SDR. IEEE Workshop on Networking Technologies for Software Defined Radio Networks (Sept. 2006)
7. Katti, S., Gollakota, S., Katabi, D.: Embracing Wireless Interference: Analog Network Coding. In: Proceedings of SIGCOMM, pp. 397 – 408 (2007)
8. Freund-Hansen, B., Nielsen, J.S., Popovski, P., Larsen, T.: Experimental Evaluation of Network Coding in Bluetooth Scenarios, Department of Electronic Systems, Aalborg University (2009)
9. Murphy, P., Sabharwal, A., Aazhang, B.: Design of WARP: a Wireless Open-Access Research Platform. In: Proceedings of EUSIPCO (2006)
10. Amiri, K., Wu, M., Duarte, M., Cavallaro, J.R.: Physical Layer Algorithm and Hardware Verification of MIMO Relays Using Cooperative Partial Detection. In: IEEE ICASSP. pp. 5614–5617 (June 2010)
11. Sharma, A., vikas Gerlara, Singh, S.R., Korakis, T., Liu, P., Panwar, S.: Implementation of a Cooperative MAC protocol using a Software Defined Radio Platform. 16th IEEE LANMAN. pp. 96 – 101 (Sept. 2008)

12. Jia, J., Zhang, J., Zhang, Q.: Cooperative Relay for Cognitive Radio Networks. In: IEEE INFOCOM. pp. 2304–2312 (April 2009)
13. Zhang, Q., Jia, J., Zhang, J.: Cooperative relay to improve diversity in cognitive radio networks. IEEE Communications Magazine 47(2), pp. 111–117 (Feb. 2009)
14. Chang, Y.J., Ingram, M.A., Frazier, R.S.: Cluster Transmission Time Synchronization for Cooperative Transmission Using Software-Defined Radio. IEEE International Conference on Communications Workshops, pp. 1–5 (May 2010)
15. Bradford, G.J.: A Framework for Implementation and Evaluation of Cooperative Diversity in Software-defined Radio. Master's thesis, University of Notre Dame (Dec. 2008)
16. GNU Radio, `http://gnuradio.org/redmine/wiki/gnuradio`, (2009)
17. Reed, I.S., Solomon, G.: Polynomial Codes Over Certain Finite Fields. Journal of the Society for Industrial and Applied Mathematics 8(2), 300 – 304 (June 1960)
18. Bou Saleh, A., Hausl, C., Koetter, R.: Outage Behavior of Bidirectional Half-Duplex Relaying Schemes, In: Proc. IEEE Information Theory Workshop, Taormina, Italy, pp. 1 – 5 (Oct. 2009)