Halmstad University Post-Print

Hard decision packet combining methods for industrial wireless relay networks

Elisabeth Uhlemann and Andreas Willig

N.B.: When citing this work, cite the original article.

©2008 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.

Uhlemann E, Willig A. Hard decision packet combining methods for industrial wireless relay networks. In: Second International Conference on Communications and Electronics, 2008. ICCE 2008.. Piscataway, N.J.: IEEE; 2008. p. 104-108.

DOI: <http://dx.doi.org/10.1109/CCE.2008.4578941>

Copyright: IEEE

Post-Print available at: Halmstad University DiVA
<http://urn.kb.se/resolve?urn=urn:nbn:se:hh:diva-1913>

Hard Decision Packet Combining Methods for Industrial Wireless Relay Networks

Elisabeth Uhlemann^{†,‡} and Andreas Willig[§]

[†]Centre for Research on Embedded Systems, Halmstad University, Halmstad, Sweden

[‡]Volvo Technology Corporation, Transport, Information and Communications, Göteborg, Sweden

[§]Telecommunication Networks Group, Technische Universität Berlin, Berlin, Germany
elisabeth.uhlemann@ieee.org, awillig@ieee.org

Abstract—We consider wireless relay networks for use in industrial applications with strict requirements on both reliable and timely communications. In most cases commercially available transceivers must be used, which implies that only hard decision detector outputs are available. Since relay networks typically use retransmissions of erroneous packets, packet combining methods have the potential to increase the information reliability without excessive delay. We evaluate three different hard decision packet combining methods for different placement of the source, the relay node and the destination. Packet combining can in general be improved with knowledge of the current channel state information, which, however, is often not available. In this paper we find a packet combining method which does not use channel state information but which delivers similar performance as the scheme that has knowledge of the channel state.

I. INTRODUCTION

The topic of providing reliable communication in *ad hoc* or wireless sensor network has been given considerable attention over the past years. Much is expected from spatial diversity techniques such as relaying networks [1]. From an information-theoretical perspective the main focus has been on finding and achieving the capacity limits regarding information flow through the network [2], on the achievable diversity gain [3] or on the tradeoffs among capacity and diversity gains [4]. From a computer networking perspective the main focus has been on finding practical protocols and tools for handling the unreliable wireless channel, while taking advantage of its inherent broadcast ability. In industrial wireless networking [5] the problem of jointly guaranteeing timeliness and reliability of packet transmissions is still one of the key challenges. There are also a number of additional constraints that should be taken into account when designing wireless networks for industrial applications. One important consideration is the almost mandatory use of commercially available wireless transceivers designed for office and home environments, which are not optimized for use in harsh industrial environments with lots of metallic clutter.

In this paper we consider a wireless, but static industrial network and address the problem of providing reliable information delivery from a source to a destination by means of a relay node. The goal is to transfer a message from the source to the destination as reliably as possible within a certain deadline, while subject to constraints imposed by the use of

practical protocols. This problem has been addressed by the authors in [6] and the solution is further improved here by considering different packet combining methods that can be applied when only hard decision detector outputs are available. More specifically, we evaluate the performance of three different packet combining methods as functions of the location of the source, the relay node and the destination. The results show that a combining scheme based on knowledge of channel state information (CSI) clearly outperforms a simple majority-voting (MV) scheme. However, we propose a third scheme which does *not* use explicit channel state information but merely selects the two copies where the minimum number of bits differ for combining, and which achieves performance similar to the CSI-based scheme. This scheme has the benefit that it can be implemented on top of commercial transceivers even if they do not offer any channel quality or signal strength information for the received packets.

The paper is structured as follows: in Section II we describe the considered system model. Following this, in Section III we describe the packet combining schemes used in this paper and in Section IV we provide simulation results comparing these schemes for different placements of the source, the relay node and the destination. We conclude the paper in Section V.

II. SYSTEM MODEL

We consider a scenario with three network stations or nodes. For each pair of nodes there exists a wireless channel with error behavior that is stochastically independent of all other channels. We assume that all channels are symmetric and all nodes are stationary. We do not take additional interference from other, co-located systems into account and thus channel errors are the result of thermal noise, pathloss and multipath fading. The thermal noise, $n(t)$, created in the receiver circuitry is assumed to be additive white Gaussian noise (AWGN) with power spectral density $N_0/2$. The chosen path loss model, $l(d)$, is the standard log-distance model [7], where d is the distance between the transmitter and the receiver. Hence, we assume that the received signal energy is inversely proportional to d^γ , where the pathloss component, γ , is chosen as $\gamma = 2$. We further assume frequency-flat block Rayleigh fading, $h(t)$, on each pathloss AWGN channel, i.e. the fading remains constant for a time that (on average) is longer than the packet size. The block-fading assumption is justified by the dominance of small packets in industrial

E. Uhlemann is partly sponsored by the Knowledge Foundation, www.kks.se. Part of this work was conducted while E. Uhlemann was visiting A. Willig at TU Berlin on a travel grant from the Swedish Research Council.

applications and we have chosen a packet size of 160 bits accordingly. The assumption of a frequency flat fading channel is justified by the fact that the bit rates used in industrial networks are only small to moderate and channel bandwidths are thus typically below the coherence bandwidth. We draw the value of $h(t)$ periodically from a Rayleigh distribution and keep it constant for the duration of the packet. The received signal is then $r(t) = l(d)h(t)s(t) + n(t)$, where $s(t)$ is the transmitted signal. We assume binary phase shift keying (BPSK) modulation such that a binary 0 is mapped to $-\sqrt{E_c}$ and a binary 1 to $+\sqrt{E_c}$, where E_c is the energy of a transmitted bit.

We consider a source, a destination and a relay node. The distance between source and destination is varied which impacts $l(d)$ and vice versa, i.e., by varying $l(d)$ we vary the distance. When the relay node is present, it is placed in between the source and the destination at different distances; $d/4$, $d/2$ and $3d/4$, where d is the distance between the source and the destination. The value $d/4$ refers to a situation where the distance between source and relay is 25% of d and thus the distance between relay and destination is 75% of d .

The following protocol aspects are adopted. We assume that all packets carry a CRC checksum at the end of the packet. The checksum covers the whole packet and is assumed to have a negligible rate of undetected errors. Thus, this CRC check can be used to determine two things: if a retransmission is necessary and if the quality of a received packet is good enough to relay. The former check is made by the destination and the latter by each potential relay node individually. Many commercial wireless transceivers possess the ability to automatically generate the CRC on the fly when transmitting a packet, and at the receiver side the CRC value can be automatically checked. While the standard behavior of a receiver is to simply drop packets having an incorrect CRC, we assume that it is possible to modify this behavior such that the receiver actually delivers an erroneous packet but with a flag indicating this situation. Such a feature is available for example with the IEEE-802.15-4-compliant ChipCon CC2420 transceiver [8]. Without this feature, packet combining cannot be implemented. Each node that is capable of performing packet combining (i.e. the destination alone or both the relay node and the destination) maintains a *packet cache*. When an arriving packet is correct, the packet cache is cleared and the correct packet is delivered. When an arriving packet is incorrect, it is stored in the packet cache and when enough packets are stored, the receiver combines them in an attempt to restore the original packet (which can be recognized from the CRC check). Whenever an attempt is made to combine packets from different retransmissions, it must be checked that all packets do indeed pertain to the same original packet, received either directly from the source or from the relay node. To achieve this, header fields like source address, destination address and sequence numbers are checked to determine whether they are the same as for the packets already present in the cache. We assume that no errors occur in the header.

An important protocol aspect concerns the question when a relay node actually becomes active. In [6] it was shown that a relay node should only become active when it possesses a

correct copy of the packet (either received directly or obtained by combining erroneous copies received so far). In this case the relay node handles *all* remaining retransmissions. When the relay node does not become active and the available retransmissions are not yet exhausted, the source node performs another trial. The source and the relay node cease transmissions when they receive positive acknowledgement from the destination. The feedback channel is assumed to be error free and thus no acknowledgement messages are ever lost or corrupted. Further, the additional energy used to transmit acknowledgement packets is ignored.

Two major performance measures for industrial communications are the *success probability* and the *confirmation delay*. The success probability is the probability of delivering a packet successfully such that the source receives an acknowledgement within a prescribed deadline. The confirmation delay measures the duration between when the source starts and stops working on a packet – note that stopping can happen either due to reception of an acknowledgement or deadline violation. We assume that the relay systems considered here have deadlines allowing a maximum of two retransmissions, i.e., three transmissions in total. Consequently, a deadline is missed only if the packet is not successfully delivered after the third attempt. We are therefore interested in how many bits are in error after the second retransmission. The performance measure adopted in this paper is thus the bit error rate (BER) as a function of the signal-to-noise ratio, E_b/N_0 , in dB, where E_b denotes the information bit energy and N_0 the noise power spectral density. It should be noted that whenever we have an uncoded system without retransmissions, E_b is the same as the energy of each bit transmitted over the channel, E_c . However, if we have a coded system, either using an error control code or using the repetition code implicit for retransmissions, E_b is no longer equal to E_c . In the case of a rate 1/2 code, i.e. one retransmission of an entire uncoded packet, we have $E_b = 2E_c$. This code rate is sometimes referred to as throughput for ARQ schemes. A system allowing retransmissions only when needed will have different code rates for different attempts and thus the overall code rate is the average of all these attempts. The simulation is thus carried out for a specific E_c/N_0 and the BER obtained is then plotted at the resulting E_b/N_0 after compensating for the increased energy due to the retransmissions. The success probability and the confirmation delay are related to the BER and the E_b/N_0 respectively. The range of E_b/N_0 can be seen as a function of the distance between transmitter and receiver.

III. PACKET COMBINING

In packet combining schemes, also referred to as type-III hybrid ARQ schemes [9, 10], the receiver does not throw away erroneous copies of a packet but instead uses the information contained therein by combining all copies pertaining to the same source packet in an attempt to restore the original packet. Packet combining algorithms can use hard or soft information about the received bits. Since one important design criterion is that our schemes should be possible to use on top of commercially available transceivers we assume that only hard

information is available. We do, however, utilize the transceivers ability to not throw away packets with an incorrect CRC value. Instead, these packets are stored in the packet cache. The packet combining algorithm works on the contents of this packet cache. Packet combining can be applied the same way both in a relay node and in the destination node.

A simple example of a combining scheme using hard information is bitwise MV, which can be uniquely applied for any odd numbers of packets in the packet cache. For an even number of packets MV can, however, not be directly applied. In this paper we therefore use *combinatorial testing* whenever two erroneous packets are contained within the packet cache. Combinatorial testing implies that the receiver tests all possible assignments of zeros and ones to the bit positions where the two packets differ. For each assignment of zeros and ones a CRC check is performed and if successful the packet is accepted. Of course, this algorithm suffers from exponential complexity and thus the number of bit allocations that can be tested is bounded. This is best achieved with hardware support which, however, does not need to be placed in the wireless transceiver and can thus be provided independently. If none of the tested assignments passes the CRC check a second retransmission is requested.

As soon as a second retransmission occurs, a CRC check is made on the newly received packet. If it does not satisfy the CRC check there will be three erroneous copies in the packet cache. We examine three different packet combining methods in this case. The first one is traditional bitwise MV. The second

one will check the number of differing positions between each pair of packets. The two most similar packets, i.e., with the least number of differing positions are thereafter selected and combinatorial testing is made on these two packets. We term this method similar pair (SP). In the third method we assume that we have knowledge about the CSI for the fading coefficient $h(t)$ for each of the three erroneous packets. The two packets with the best fading coefficients are then selected for combinatorial testing. This method is denoted CSI.

IV. SIMULATION RESULTS

The following results are compiled using Monte-Carlo simulations in Matlab with settings according to Section 2. At least 500 000 trials is made for each value of E_b/N_0 . The BER is the number of bits still in error after all allowed retransmissions has been exhausted.

In Figure 1 three different relay schemes are compared. Each scheme allows a maximum of two retransmissions. The first scheme uses no packet combining. The second applies packet combining at the destination such that combinatorial testing is made after two failed transmissions and MV after the third failed transmission. The third scheme applies the same packet combining strategy but now both at the relay node as well as at the destination. For each of these three schemes, the relay node is placed at $d/4$, $d/2$ and $3d/4$, where d is the distance between the source and the destination. Three schemes without relay nodes are also plotted for reference: the first one does not allow retransmissions, the second allows a maximum

Figure 1. BER for schemes with relay nodes at different locations, with and without combining.

of two retransmissions but no packet combining, and the third one allows a maximum of two retransmissions as well as packet combining at the destination using combinatorial testing and MV.

It can be seen from Figure 1 that having and using a relay node is always beneficial. Further, packet combining improves performance at the expense of increased receiver complexity. Depending of the location of the relay node and the distance between the source and the destination, i.e., by their pathloss $l(d)$, a scheme with a relay node but without packet combining can outperform a scheme without relay node but with packet combining. It can further be seen that combining both at the relay node and at the destination gives the best performance. In addition, placing the relay node at $d/4$ always results in the worst performance. Finally, whether it is best with the relay node located at $d/2$ or $3d/4$ depends on the signal-to-noise ratio, i.e., the distance between the source and the destination. If the source and the destination are located far apart such that the signal-to-noise ratio is low, it is better to place the relay node midway between them, so that it can work as a multihop node. However, if the source and the destination is well within communication range it is better to place the relay node closer to the destination so that packets that almost makes it to the destination, will make it to the relay node which can then retransmit virtually error free.

In Figure 2 three different packet combining methods are compared. A scheme without relay node but with

combinatorial combining and MV combining is still used as a reference. All schemes apply the respective packet combining method both at the relay node and at the destination. The scheme using MV both at the relay node and at the destination from Figure 1 is plotted again for comparison in Figure 2. In addition a scheme using combinatorial testing after two failed transmissions and SP after the third failed transmission is plotted for different locations of the relay node. Finally, a scheme using combinatorial testing after two failed transmissions and CSI after the third failed transmission is plotted for different locations of the relay node.

It can be seen from Figure 2 that both SP and CSI outperforms MV and that the relay location has a higher influence on the performance than the specific type of packet combining method used. More importantly, SP and CSI have very similar performance which indicates that good performance can be obtained even without knowledge about the current channel state. In many cases SP is slightly better than CSI with may imply that the fading coefficient is less important to the overall quality of a packet than e.g., the effects of pathloss. Therefore, knowledge about whether the packet is obtained from the relay node or from the source node may further improve performance when selecting which packets to combine.

V. CONCLUSIONS

Three different packet combining methods suitable for wireless industrial relay networks have been evaluated. It was

Figure 2. BER for schemes with relay nodes at different locations, using different packet combining methods.

concluded that packet combining significantly improves the performance such that higher information reliability can be obtained with fewer retransmission attempts and thus lower delay. Packet combining using majority voting provides good performance but even better for the fading channel is to exclude the “worst” of three erroneous copies and do combinatorial combining on the two remaining packets. In one combining scheme the worst packet was selected using knowledge of the channel state information and in the other scheme the two packets having the fewest differing bit positions were selected. These two strategies resulted in very similar performance. Since the latter method does not require any channel state information, it does not require additional support from the wireless transceiver and consequently it can be implemented even when using the simplest transceivers.

REFERENCES

- [1] S. N. Diggavi, N. Al-Dhahir, A. Stamoulis, and A. R. Calderbank, "Great expectations: the value of spatial diversity in wireless networks," *Proceedings of the IEEE*, vol. 92, no. 2, pp. 219-270, February 2004.
- [2] G. Kramer, M. Gastpar, and P. Gupta, "Cooperative strategies and capacity theorems for relay networks," *IEEE Transactions on Information Theory*, vol. 51, no. 9, pp. 3037-3063, September 2005.
- [3] J. N. Laneman, D. N. C. Tse, and G. W. Wornell, "Cooperative diversity in wireless networks: Efficient protocols and outage behaviour," *IEEE Transactions on Communications*, vol. 50, no. 12, pp. 3062-3080, December 2004.
- [4] M. Yuksel and E. Erkip, "Multiple-antenna cooperative wireless systems: A diversity-multiplexing tradeoff perspective," *IEEE Transactions on Information Theory*, vol. 53, no. 10, pp. 3677-3696, October 2007.
- [5] A. Willig, K. Matheus, and A. Wolisz, "Wireless technology in industrial networks," *Proceedings of the IEEE*, vol. 93, no. 6, pp. 1130-1151, June 2005.
- [6] E. Uhlemann and A. Willig, "Joint design of relay and packet combining schemes for wireless industrial networks," to appear in *Proc. IEEE Vehicular Technology Conference*, Singapore, 2008, 5 pages.
- [7] T. S. Rappaport, *Wireless Communications - Principles and Practice*: Prentice-Hall, 1996.
- [8] Chipcon, *2.4 GHz IEEE 802.15.4 / ZigBee-ready RF Transceiver*, Chipcon Products from Texas Instruments, 2004.
- [9] P. Sindhu, "Retransmission error control with memory," *IEEE Transactions on Communications*, vol. 25, no. 5, pp. 423-429, May 1977.
- [10] S. Kallel, "Complementary punctured convolutional (CPC) codes and their applications," *IEEE Transactions on Communications*, vol. 43, no. 6, pp. 2005-2009, June 1995.