Halmstad University Post-Print

Joint design of relay and packet combining schemes for wireless industrial 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. Joint design of relay and packet combining schemes for wireless industrial networks. In: 2008 IEEE 67th Vehicular Technology Conference-spring: Marina Bay, Singapore, 11-14 May 2008. Piscataway, N.J.: IEEE; 2008. p. 2441-2445.

DOI: <http://dx.doi.org/10.1109/VETECS.2008.539>

Copyright: IEEE

Post-Print available at: Halmstad University DiVA
<http://urn.kb.se/resolve?urn=urn:nbn:se:hh:diva-1910>

Joint Design of Relay and Packet Combining Schemes for Wireless Industrial Networks

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

[†]Center for Research on Embedded Systems, Halmstad University, Halmstad, Sweden, email: bettan@ide.hh.se

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

[§]Telecommunication Networks Group, Technical University of Berlin, Berlin, Germany, email: awillig@ieee.org

Abstract—Wireless industrial networks differ in many respects from other types of wireless networks. In particular, since many applications impose tight real-time and reliability requirements at the same time, and packet sizes tend to be small. In this paper we design a simple and practically implementable protocol in which relaying and packet combining work together to improve the probability that packets are delivered within a prescribed deadline over fading channels. The results indicate that such a combination can be fruitfully employed in wireless industrial networks.

I. INTRODUCTION

Industrial networks [1] differ in several respects from other types of networks. Firstly, since the data traffic mostly entails sensor observations or actuator commands, packet sizes tend to be small, on average in the order of a few bytes. Secondly, to allow controllers to act on a recent image of the underlying physical process, packet transmissions are often subject to tight constraints *both on real-time and reliability*. Finally, industrial networks usually have star topologies, in which all sensors and actuators exchange data with or through a central controller.

Wireless technologies are much coveted for industrial applications, since they allow easy attachment of mobile subsystems to a network, they reduce the danger of cable breakage and they can help shortening the machine downtimes required for re-configuration [2]. However, providing reliable and timely data transmissions over error-prone, time-varying wireless channels constitutes a significant challenge. *Spatial diversity techniques* [3] have been recognized as a key approach to take on the challenge. Among these techniques, cooperative diversity [4], [5], such as relay schemes, are especially interesting, since they can be implemented also in networks where all nodes, due to cost or form factor, only have one single antenna. Many information-theoretic results are available for relay channels (e.g., [6], [7]), the work on practical integration of relaying approaches into existing MAC and link layer protocols is ongoing, mostly with the goal of improving performance over existing WLAN technologies like IEEE 802.11 (e.g., [8], [9]).

In this paper we consider the design of relaying protocols especially suited for wireless industrial networks. Further, we

assimilate relaying with packet combining carried out by the relay nodes and by the destination node, and this scheme is in turn integrated into an ARQ protocol. We have paid special attention to make sure that our design is "practical", i.e. that it can be implemented on top of the baseband level of commercially available wireless transceivers, where typically no soft-information but only the hard decisions made about each received bit is available. Hence, our schemes belong to the class of *decode-and-forward* relaying schemes [10]. We evaluate our proposed protocols by simulation over frequency-flat block fading channels, and show that significant improvements of the probability to successfully deliver a packet within a prescribed deadline are possible.

The paper is structured as follows: in Section II the system model is described, detailing the network setup and the channel model. In Section III we describe our approaches to packet combining and our relaying protocol. In Section IV numerical results are presented for a random node deployment. Finally, our conclusions are given in Section V.

II. SYSTEM MODEL

A. Network setup

The two major performance measures adopted in this paper are the *success probability* and the *confirmation delay*. Both refer to packets that are to be transmitted from the source to the destination and do not take MAC delays or queuing delays into account. We consider the uplink direction from a source node towards a central controller. The *success probability* is the probability of delivering a packet successfully (i.e. such that the source receives an acknowledgement) within a prescribed deadline of 10 ms from a source S to the destination D . The *confirmation delay* measures the duration between when the source starts and stops working on a packet – stopping can happen either due to reception of an acknowledgement or deadline violation.

We have chosen a two-dimensional deployment with one centralized controller, to which all packets are directed. A total of 30 source nodes are randomly placed in a square of 35×35 meters, centered at the origin, where also the central controller is placed, Figure 1. It is assumed that each node in the network knows all its immediate neighbors, i.e. all nodes that potentially can be reached with a direct transmission.

E. Uhlemann is partly funded 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.

Fig. 1: Investigated node deployment

All nodes use simple half-duplex wireless transceivers which work on the same frequency band and always use the same transmission data rate. When two or more members having the same distance to a receiving member, transmit different messages in parallel, a collision results, rendering all packets undecodable. We assume that the wireless transceiver can run in two different reception modes. In one mode, the *regular transceiver mode*, it performs a CRC check on incoming packets and drops all erroneous packets. This is a standard feature available with most modern transceivers. In the second mode the transceiver does not drop erroneous packets if the CRC fails. Instead, it hands over the packet with the information that the checksum is wrong. We term this mode the *CRC ignorant transceiver mode*.

B. Channel model

For each pair of members there exists a wireless channel with error behavior that is stochastically independent of all other channels. Channel errors are the result of thermal noise, path loss and multipath fading. We do not take additional interference from other, co-located systems into account. We assume that all channels are symmetric and all nodes are stationary. The thermal noise created in the receiver circuitry is a parameter in our system and it is assumed to be additive white Gaussian noise (AWGN) with power spectral density $N_0/2$. The path loss model chosen is the standard log-distance model [11]. We further assume block fading on each pathloss AWGN channel, i.e. the fading remains constant for a time that (on average) is significantly longer than the packet size. This assumption is justified by the dominance of small packets in industrial applications. As one particular model for block fading we adopt a Gilbert-Elliot model, i.e. a two-state Markov chain. During either of the two states the fading attenuation is constant: 0 dB in the "good" state and -30 dB in the "bad" state, and the state holding times are exponentially distributed. Finally, we assume binary phase shift keying (BPSK) modulation.

III. PROTOCOL DESIGN

We describe the packet combining schemes, followed by a description of the relaying framework. The latter has also been presented in [12], without considering packet combining.

A. Packet combining

Packet combining schemes, termed type-III hybrid ARQ schemes [13], have the property that 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. Packet combining algorithms can make use of hard or soft information about the received bits. We have only hard information available here since one design goal is that our schemes should be usable on top of commercially available transceivers. We do, however, assume that we have access to the *CRC ignorant transceiver mode* so that no erroneous packets are discarded. Packet combining can be applied the same way both in the relay nodes and in the destination node. Each receiver maintains a *packet cache* in which the different copies pertaining to the same source packet, but received over different channels (from the source or from a relay node at different time instances) are stored. The packet combining algorithm works on the contents of this packet cache. An important consideration is then when the packet cache should be cleared. Each receiver must detect when a newly arrived packet belongs to a source packet different from the one in the packet cache. We solve this by using the values of three MAC header fields: the source MAC address, the destination MAC address and the value of a packet sequence number generated by the source. When any of these values has changed for an incoming packet, the packet cache is cleared and the new packet is placed into it, otherwise the new packet is appended to the packet cache. We further assume that the MAC header is protected by its own checksum, and an incoming packet is dropped when this checksum is wrong.

A simple example of a combining scheme using hard information is bitwise majority voting (MV), which can be uniquely applied for any odd numbers of packets in the packet cache. For any even number of packets MV cannot be directly applied. In this paper we therefore use an algorithm, which combines MV with *combinatorial testing*, and which runs each time a new packet is added to the cache. It works as follows: whenever the number of packets stored in the cache is odd, the receiver performs bitwise MV on all stored packets. If the resulting packet checksum is correct, the packet is accepted. Whenever the number of stored packets is even, the receiver performs bit-wise MV over all packets, but memorizes those bit positions that have the same number of zero and one votes (termed *testable positions*). For those bits a combinatorial testing procedure is adopted: all possible assignments of zeros and ones to the testable positions are evaluated to see if one of them satisfies the checksum test. This process stops when the checksum is correct or when a certain maximum number of possible assignments has been tested without success. Of course, this algorithm suffers from exponential complexity in the number of testable positions. Even with a hardware implementation (which is highly advisable) there is a limit on the number of assignments that can be tested. In the remainder of the paper we assume that a maximum of eight testable positions are allowed, resulting in a maximum of 2^8 CRC

Fig. 2: The effect of one relay node, using two different relay strategies and packet combining

tests.

Next, we consider how a relay node determines if it should retransmit a packet or whether it is better to let the source retransmit. We evaluate two different scenarios. In the first scenario, termed *Always*, the relay node does not apply packet combining but simply retransmits whatever it has previously received, irrespectively of if it was erroneous or not. Packet combining thus only takes place in the destination node. If the relay node is located somewhere in-between the source and the destination, it has a lower pathloss and as such may be the best candidate to retransmit. In the second scenario, termed *Only Correct*, the relay node only takes care of the retransmission if the received packet has passed the checksum test. Packet combining can in this scenario be used both in the relay and in the destination node and it can help a potential relay node to become a relay.

Figure 2 shows the bit error rate (BER) versus the ratio between the bit energy, E_b and the noise N_0 for several ARQ schemes allowing a maximum of three transmissions (from source or relay). The results are obtained for a fading channel without LOS. For the schemes having a relay node, it is located half way between the source and the destination. Packet combining with MV and combinatorial testing is used. The benefit of relaying and packet combining is clearly shown. When packet combining is introduced in the relay node as well, it results only in minor improvements for low E_b/N_0 . Further, it clearly shows that the scheme *Only Correct* has better performance than *Always*, and thus a potential relay node should only become a relay once it has correctly received a packet.

B. Relaying protocol

When constructing a relaying protocol with several potential relayers, a number of issues have to be resolved. One important task is to control the *activation* of relayers. An option would be to enable relaying whenever the destination of a packet does not send a direct MAC-layer acknowledgement. However, due

Fig. 3: Round structure of the relaying framework ($n = 2$)

to the properties of the wireless medium and depending on the precise kind of destination feedback, it might happen that the source node receives an acknowledgement while the relayers do not. In such a case the relayers become active while the source node has started to work on the next packet. Source and relayers should not be able to make contradictory decisions, we refer to this as keeping *consistency*. A second important task is the *selection* of relayers. When the source transmits a packet, the broadcast property of the wireless medium might lead to a situation where multiple relayers $R_1, R_2, \dots, R_n$ receive a correct copy of the packet. This set of relayers is of random size and varies over time. Without any coordination, the relayed packets issued by $R_1, \dots, R_n$ would collide at the destination, rendering the whole relaying procedure useless. Three fundamental approaches can be conceived for relayer selection: source-controlled relayer selection, relayer-controlled relayer selection and destination-controlled relayer selection. The scheme described in this paper belongs to the class of source-controlled schemes, where the selection of a relayer is controlled by the source node. To achieve this, the source S includes into its packet additional MAC header fields specifying the (set of) relayers – at the expense of increased packet lengths. Clearly, this approach avoids consistency issues upfront.

We present a simple relaying framework that is tailored to the case of small data packets and therefore useful for wireless industrial applications. The source can use one or more relayers, or it can avoid relaying at all.

1) *Description of relaying framework:* The framework is round-based (compare Figure 3). A round starts with a transmission from the source S to the destination D . This initial packet is followed by a number of n relay slots, one acknowledgement slot for the destination and finally n acknowledgement slots for the relayers (arranged in reverse order as compared to the relay slots) in which the relayers forward the acknowledgement whenever they have picked it up. When the source node has not received an acknowledgement at the end of the round but has still sufficient time before deadline expiration, a new round is started.

The source transmits the initial data packet. The extended MAC header of this packet contains a flag indicating the desire to enable relaying, the number n of relaying slots following the source packet, a list of n relayer MAC addresses, and a field denoting the current relaying slot (initialized with zero, denoting a transmission coming directly from the source). When at the beginning of relay slot the relayer R_i listed at position $i \in \{1, \dots, n\}$ possesses a correct copy of the packet (compare the advantage of “Only correct” over “Always” in Section III-A), it simply transmits the packet in this slot. In

addition, R_i writes his slot number i into the packet header of the relayed packet and re-calculates the packet checksum. This has two purposes:

- It allows the destination at any time to calculate the point in time where it can send its acknowledgement.
- It gives “downstream” relayers a chance to operate even when they have not received the packet from the source. When a downstream relayer R successfully decodes the packet, it can check whether there is a slot allocated for him and, using the slot number, whether this slot has already passed or whether there is still a chance for him to transmit.

The n data slots and the n acknowledgement slots occur unconditionally – there is no additional carrier sensing by the source or relayers to check for the presence of an acknowledgement. This simplifies the design and eliminates consistency issues upfront. As a downside, however, when no relayer picks up the source packet, the time for all the n data slots and the n acknowledgement slots is unexploited.

2) *Relayer selection*: The source has freedom to decide how many, n , and which relayers it wants to use. The choice of n will in practice be determined by the packet deadline and by the network topology – when for a source node there is a relayer placed at a “good position” (for example in the middle between source and destination) a value of $n = 1$ might suffice, but higher values for n gives the protocol a chance to find good *chains* of relayers when no single good relayer exists.

We assume that the network is static. We adopt two different relayer selection strategies. The first strategy, called *random*, serves as a baseline strategy. Here, the source node simply picks randomly n out of its neighbors as relayers. In the second strategy, called *genetic*, a learning approach based on a genetic algorithm [14] is adopted to find good chains. The algorithm works on a *population* of chains (each of length n), having a fixed *population size*. Initially, the population is selected at random. Each chain in the current population is tested for T times. An individual test is either successful (i.e. the source receives an acknowledgement when using this specific chain) or fails. Testing all the members of a population is referred to as a *testing round*. The algorithm performs a limited number of testing rounds. At the end of a testing round a new population is created using the results for the current population. The $\alpha \cdot 100\%$ best (having the highest number of successes) members of the current population are carried over into the new population. These are called *survivors*. The next $\beta \cdot 100\%$ members of the new population are created from mutations of randomly chosen members of the $\alpha \cdot 100\%$ survivors (please refer to [12] for details of this). An additional $\gamma \cdot 100\%$ members of the new population are created from crossovers of the survivors. The remaining $(1 - \alpha - \beta - \gamma) \cdot 100\%$ of the members are randomly chosen from the set of all possible chains.

The genetic scheme is used in two different flavors. In the first flavor (called *genetic*) the single best survivor chain is always returned at the end of the learning phase. In the second

variant (called *genetic-switch*) the best survivor chain is used in the first round, the second-best survivor chain in the second round and so forth.

IV. RESULTS

In this section we present numerical results for the success probability and the confirmation delay for the network setup shown in Figure 1. The results have been obtained with the help of a discrete-event simulation tool. For each of the possible source nodes i we simulate until the relative precision for the success probability p_i at a confidence level of 1% is below 1% of the achieved success probability, however, always a minimum of 40,000 packets is simulated. Queueing effects and medium access control are not considered.

The major channel parameters are as follows. The path loss exponent is three. For the two-state Markov fading model both the good state holding time and the bad state holding time are fixed to 65 ms. In the good state, the channel attenuation on top of the log-distance attenuation is 0dB, in the bad state it is -30dB. The relevant physical layer parameters are based on existing IEEE 802.15.4 transceivers [15]. The data rate is 250 kbit/s, and the transceiver turnover time between transmit and receive modes corresponds to 40 bit times.

The major protocol parameters are as follows. The number n of relaying slots is fixed to $n = 2$, the number of test trials per chain is $T = 30$, the width of a relayer MAC address is 8 bits (each relayer’s MAC address must be added to the packet, but industrial networks tend to have small node counts and addresses can be kept short), the number of testing rounds for the genetic algorithm is 15 and the population size is 20. The operational parameters for the genetic algorithm have been chosen as $\alpha = 0.3$, $\beta = 0.2$ and $\gamma = 0.2$. The user data size is 80 bits, the MAC header and trailer size (without relaying-related fields) is 76 bits, and the acknowledgement is of size 56 bits.

Fig. 4: Comparison of pure ARQ policies with and without packet combining (mean bad state holding time: 65ms)

In Figure 4 we compare the success probabilities of a pure ARQ scheme (i.e. a scheme where the source makes immediate retransmissions according to the stop-and-wait protocol without using any relayer) with and without packet combining by means of density plots. In the figures, brighter areas indicate a higher success probability, darker areas a smaller one. It should be noted that for visualization purposes the plots are

Experiment	Avg./Min.Succ.Prob.	Avg./Max.Delay
Pure ARQ (no comb.)	0.283 / 0	7.31 / 9.36 ms
Pure ARQ (4 testable pos.)	0.349 / $3.5 \cdot 10^{-4}$	7.02 / 9.36 ms
Pure ARQ (8 testable pos.)	0.358 / $3.75 \cdot 10^{-4}$	6.97 / 9.36 ms
Genetic (8 testable pos.)	0.528 / 0.021	6.95 / 10.0 ms
Gen.-Switch (8 testable pos.)	0.558 / 0.008	6.87 / 10.0 ms
Random (8 testable pos.)	0.352 / 0.002	8.23 / 10.0 ms
Genetic (no comb.)	0.498 / $2.5 \cdot 10^{-4}$	7.04 / 10.0 ms
Genetic-switch (no comb.)	0.519 / $1.2 \cdot 10^{-4}$	7.05 / 10.0 ms

TABLE I: Average/minimum success prob. and avg./max. confirmation delay, taken over all source nodes

smoothed. It can be seen that packet combining provides advantages. This is also substantiated by the results for the average success probability (taken over all possible sources), see Table I.

Fig. 5: Comparison of relaying policies with/without packet combining (mean bad state holding time: 65ms)

In Figure 5 we present the success probabilities for the case of relaying (with and without combining). Again, the average success probabilities (taken over all source nodes) are displayed in Table I. Relaying plus combining is on average better than relaying without combining, and furthermore the genetic-switch policy is advantageous as compared to the pure genetic policy. All relaying policies improve significantly over the pure ARQ schemes with or without combining. The random relay selection scheme (with packet combining) gives virtually no advantages over a pure ARQ scheme (with packet combining).

We also briefly compare the average confirmation delays of the different schemes, shown in Table I. The average confirmation delay (again averaged over all source nodes) is a direct measure of the total bandwidth consumption of these schemes. The numbers show that the relaying approaches (except the random selection scheme) slightly outperform the pure ARQ

approaches, i.e. they have smaller average confirmation delays. This results from the presence of a sufficient number of “poor” nodes for which the pure ARQ schemes would exhaust the complete time budget. This outweighs those cases where pure ARQ and relaying would need only one round and where pure ARQ has an advantage due to its shorter round time.

V. CONCLUSIONS

The results presented in this paper demonstrate that schemes merging relaying and packet combining can be designed in a “practical” way (i.e. on top of commercially available wireless transceivers) and that this merger can significantly improve the performance metric of success probability, which is of prime importance in industrial, time-critical applications. In the future we will investigate further packet combining schemes, and explore how such “learning” relay selection schemes like the genetic schemes will behave when the channels become less stationary, for example as a result of shadowing or significant changes in path loss.

REFERENCES

- [1] J.-P. Thomesse, “Fieldbus technology in industrial automation,” *Proceedings of the IEEE*, vol. 93, no. 6, pp. 1073–1101, 2005.
- [2] A. Willig, K. Matheus, and A. Wolisz, “Wireless technology in industrial networks,” *Proceedings of the IEEE*, vol. 93, no. 6, pp. 1130–1151, 2005.
- [3] S. N. Diggavi, N. Al-Dahir, 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, Feb. 2004.
- [4] A. Scaglione, D. L. Goeckel, and J. N. Laneman, “Cooperative communications in mobile ad hoc networks,” *IEEE Signal Processing Magazine*, pp. 18–29, Sept. 2006.
- [5] A. Nosratinia, T. E. Hunter, and A. Hedayat, “Cooperative communication in wireless networks,” *IEEE Communications Magazine*, vol. 42, no. 10, pp. 74–80, Oct. 2004.
- [6] J. N. Laneman, D. N. C. Tse, and G. W. Wornell, “Cooperative diversity in wireless networks: efficient protocols and outage behaviour,” *IEEE Transactions on Information Theory*, vol. 50, no. 12, pp. 3062–3080, Dec. 2004.
- [7] T. M. Cover and A. A. E. Gamal, “Capacity theorems for the relay channel,” *IEEE Transactions on Information Theory*, vol. 25, no. 5, pp. 572–584, Sept. 1979.
- [8] H. Zhu and G. Cao, “rdcf: A relay-enabled medium access control protocol for wireless ad hoc networks,” *IEEE Transactions on Mobile Computing*, vol. 5, no. 9, pp. 1201–1214, Sept. 2006.
- [9] L. M. Feeney, B. Cetin, D. Hollos, M. Kubisch, S. Mengesha, and H. Karl, “Multi-rate relaying for performance improvement in IEEE 802.11 WLANs,” in *Proc. International Conference on Wired/Wireless Internet Communication (WWIC)*, Coimbra, Portugal, May 2007.
- [10] J. N. Laneman and G. W. Wornell, “Distributed space-time coded protocols for exploiting cooperative diversity in wireless networks,” *IEEE Transactions on Information Theory*, vol. 49, no. 10, pp. 2415–2425, Oct. 2003.
- [11] T. S. Rappaport, *Wireless Communications – Principles and Practice*. Upper Saddle River, NJ, USA: Prentice Hall, 2002.
- [12] A. Willig, “How to exploit spatial diversity in wireless industrial networks,” in *Proc. 7th IFAC International Conference on Fieldbuses and Networks in Industrial and Embedded Systems (FET’2007)*, Toulouse, France, Nov. 2007, invited paper.
- [13] S. Kallel, “Complementary punctured convolutional codes (cpc) and their applications,” *IEEE Transactions on Communications*, vol. 43, no. 6, pp. 2005–2009, 1995.
- [14] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*. Bonn, Germany: Addison Wesley, 1989.
- [15] Chipcon, *2.4 GHz IEEE 802.15.4 / ZigBee-ready RF Transceiver, CC2420 Data Sheet*, Chipcon Products from Texas Instruments, 2004.