accepted for publication in: IFAC Annual Reviews in Control

# HOW TO EXPLOIT SPATIAL DIVERSITY IN WIRELESS INDUSTRIAL NETWORKS

**Andreas Willig** *

* *Telecommunication Networks Group, Technical University of Berlin*

Abstract: A key challenge for wireless industrial networking is to successfully transmit a packet within a prescribed deadline despite the unfriendly properties of the wireless transmission medium. A very promising class of approaches exploits the concept of spatial diversity to improve the robustness of wireless transmission. The concept of *relaying* belongs to this class and in this paper we discuss how protocols for wireless industrial networks can be designed to include relaying approaches. An example protocol design is presented and the achievable probability for not missing a prescribed deadline is assessed in an example scenario, demonstrating the significant improvements possible with spatial diversity techniques.

Keywords: Wireless industrial communications, error control, spatial diversity, MIMO, relaying, cooperative diversity

## 1. INTRODUCTION

Wireless industrial LANs differ from other wireless LANs, like the ones used in home or office environments, mainly in that they have to satisfy stringent requirements in terms of real-time and reliability at the same time, for example when important alarms have to be transmitted (Willig *et al.*, 2005). It is well-known that the wireless channel can introduce channel errors at significant and time-varying rates due to phenomena like interference, noise, path loss, shadowing or fading (Tse and Viswanath, 2005), and these channel errors pose a significant challenge for the required real-time and reliability properties.

It is clear that this challenge needs to be addressed at the lower layers of a communication stack, namely the physical layer (PHY), the medium access control (MAC) sublayer and the link layer. The MAC layer is central for the timing aspect, whereas the link layer, and especially its error control strategy, has an immediate impact on the reliability. When the error control scheme employs redundancy in time like packet retransmissions or coding overhead, it also affects the real-time behavior. For the design of wireless industrial communication protocols, the MAC and link layer are the major playground, whereas on the PHY layer often transceivers for standardized technologies like IEEE 802.11 (LAN/MAN Standards Committee of the IEEE Computer Society, 1999) or IEEE 802.15.4 (LAN/MAN Standards Committee of the IEEE Computer Society, 2006) are preferred because of their commercial availability.

A great variety of error control schemes has been developed for use over a single wireless channel. They can be broadly subdivided into error-control coding and retransmission-based automatic-repeat-request (ARQ) schemes. In the last years coding schemes have been developed that, over certain types of transmission channels, are able to operate very closely to information-theoretic limits and can be considered state-of-the-art in coding theory. Examples are low-density-parity-check codes and turbo codes (Hanzo *et al.*, 2007; Biglieri, 2005). However, the usefulness of these codes in industrial applications is limited. One reason for this is their relatively high computational complexity at the receiver, which translates into higher system costs. Secondly, many of these coding schemes can play out their advantages only for large packet sizes, which does not fit well together with the (very) short packets and the limited number of packets that a node has to transmit per cycle in industrial real-time applications. Furthermore, even these codes are not able to compensate prolonged times of large channel attenuations as they are frequently encountered on wireless channels that suffer either from fading (so-called *deep fades* or *channel outages*) or from obstacles moving into the line of sight between transmitter and receiver. The same is true for ARQ schemes as well: immediate retransmissions on a channel that is currently in a deep fade are often useless. The sometimes advocated idea of postponing retransmissions until the deep fade (maybe) ends is obviously not a good idea for packets with real-time deadlines.

A fundamental approach to circumvent this problem is the exploitation of *spatial diversity* (Diggavi *et al.*, 2004). In this class of mechanisms the single-channel restriction between a wireless transmitter and receiver is removed and information is transmitted over multiple spatial channels. The hope behind this is that these different channels are stochastically independent and with only small probability in a deep fade at the same time.

In this paper a brief introduction to spatial diversity techniques is given and it is discussed how one of them, namely relaying, can be exploited in industrial communication systems with their specific requirements (short packets, deadlines, high reliability). Relaying is one example of cooperative transmission techniques, which can be implemented even with simple and cheap single-antenna nodes having limited computational facilities, and are therefore very interesting for industrial applications.

The paper is structured as follows: in Section 2 some fundamentals of spatial diversity are briefly reviewed. In Section 3 the concept of relaying is explained in general, and some of its most important protocol design aspects (especially for industrial applications) are discussed. Following this, in Section 4 a relaying protocol framework suitable for small packets is proposed together with different approaches for the selection of relayers. The performance of these approaches, measured in terms of the probability to successfully deliver a packet within a prescribed deadline, is assessed for an example scenario. Conclusions are presented in Section 5.

## 2. SPATIAL DIVERSITY FUNDAMENTALS

In wireless technologies information is conveyed by the transmission of radio waves through space. The transmitted waveforms are subject to reflections, diffractions or scattering, and as a result several delayed copies of a waveform are superposed at the receiver and create constructive or destructive interference. When the position of the transmitter, of the receiver or of some object in the propagation environment changes, the number of propagation paths, their respective delay and attenuation can change as well, leading to a change in the interference situation. The signal strength at the receiver hence can vary over time, thus creating a *fading channel* (Biglieri *et al.*, 1998). The fading process on a wireless channel is hardly predictable and therefore considered *random*. Following (Diggavi *et al.*, 2004), diversity is defined as "the method of conveying information through multiple independent instantiations of these random attenuations". The independency has a positive effect: as the number of instantiations is increased, the probability that none of them is of sufficient quality to allow successful decoding decreases.

In *spatial diversity schemes* the independent realizations are obtained from multiple antennas placed at geographically sufficiently separated locations. In the *single-user case* only a single transmitter and receiver are considered, and at least one of them has multiple antennas. Recent MIMO (multiple-input, multiple-output) techniques (Bölcskei, 2006; Paulraj *et al.*, 2004) like the upcoming IEEE 802.11n belong to this class. In the *multi-user case* further (geographically separated) nodes are involved in a transmission between a transmitter and receiver – this is also often referred to as *cooperative diversity* (Scaglione *et al.*, 2006). One example are relaying techniques (discussed below), another one are cooperative MIMO approaches (see for example (Cui *et al.*, 2004; del Coso *et al.*, 2007)). In cooperative MIMO, two groups of nodes form a virtual transmit and receive antenna array, respectively. When a node wants to transmit a packet to another node, the transmitting node sets up a virtual transmit array of neighbored nodes, disseminates the packet to the array members, and each array member transmits a copy of the packet. On the other side, the receiver sets up a receive array. The receive array members receive (parts of) the incoming packets and forward their observations to the ultimate receiver, which then can try to decode the packet.

Different types of gains can be achieved with spatial diversity techniques. We explain them for the example of single-user MIMO techniques, but many relationships are similar in the multi-user case. A *capacity gain* is achieved when the achievable transmission rate between transmitter and receiver is higher with spatial diversity techniques than without. To maximize the rate of a MIMO system, the transmitter could send independent data streams over its multiple antennas. Information-theoretic results for the capacity of MIMO channels as well as results for more practical receiver structures show that the achievable rates grow asymptotically linearly with $M = M_r = M_t$ where $M_t$ is the number of transmit antennas and $M_r$ is the number of receive antennas.

On the other hand, a *diversity gain* is achieved when thanks to spatial diversity techniques the bit error probability between transmitter and receiver can be reduced. This could be achieved when the multiple transmit antennas do not transmit independent information streams, but when coding is used to introduce correlation among them. The design of appropriate *space-time codes* (Liew and Hanzo, 2002) is currently a lively research area.

Both types of gains are available with spatial diversity techniques, but there is a tradeoff among them (Zheng and Tse, 2003).

In practical terms, for industrial applications multi-user techniques like the relaying approach discussed below are attractive. An attractive feature of multi-user approaches is that the individual nodes need only a single antenna, which reduces system complexity. True MIMO systems require significant complexity at the receiver side. Secondly, in multi-user techniques the spacing between the antennas can be larger than it is for true multi-antenna nodes, where the typically small size of the node puts practical constraints on the number of mountable antennas. This can be beneficial when obstacles block the direct line-of-sight between transmitter and receiver, since a third-party node can be used as a "detour" for transmitting information.

## 3. RELAYING FOR INDUSTRIAL APPLICATIONS

In this section we introduce the concept of relaying and discuss the issues that need to be resolved when relaying approaches are integrated into link-layer protocols.

### *3.1 The concept of relaying*

The concept of *relaying* is not new, the first theoretical works date back to the seventies (see for example (Cover and Gamal, 1979), see also (Cover and Thomas, 2006, Chap. 15)). In relaying schemes, there are a number of *relay nodes* that help in the transmission between a sender and a receiver – all involved nodes can be single-antenna nodes. These relay nodes possibly receive the senders packet and can assist with performing retransmissions when the receiver has not received the packet. [1] Relaying is hence tightly coupled to ARQ protocols. Since the sender and relayers have different geographical locations, the receiver gets information over different spatial channels, thus exploiting spatial diversity. A lot of information-theoretic research has been carried out to investigate capacity and diversity gains achievable with relaying (e.g. (Kramer *et al.*, 2005; Laneman *et al.*, 2004)). In the last years, there have also been significant activities towards practical integration of relaying into wireless protocols, see for example (Zhu and Cao, 2006), or (Willig, 2003) for a proposal in an industrial setting.

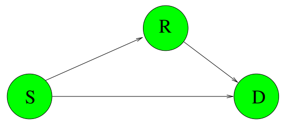



Fig. 1. Basic relaying operation

In its simplest form, the relaying ARQ channel consists of three nodes $S$, $D$ and $R$, see also Figure 1. The source $S$ wants to send a packet towards the destination $D$. A third node, the relayer $R$ picks up $S$'s signals and forwards its observations to $D$, which can combine $R$'s observations with his own ones to decode the packet. Some fundamental variants of relaying are:

- Decode and forward: it is required that node $R$ successfully decodes the packet (i.e. finds a correct packet checksum) before it forwards it further towards $D$.
- Amplify and forward: node $R$ samples the waveform incoming from $S$ without trying to decode it. After this, node $R$ forwards the sampled waveform to $D$ which can combine it with his own sampled waveform for joint decoding.
- A variant of decode and forward rests on the abilities of an *ignorant transceiver*: a relayer can accept (possibly erroneous) packets from its transceiver and forward them. This way the destination node receives more information to work on than in pure decode-and-forward schemes. On the other hand, the relayers transceiver already makes hard decisions on the received bits and some information is lost as compared to amplify-and-forward.

[1] It should be noted that relaying, as described in this paper, is considered a link-layer technique, in which third-party nodes (the relayers) are used to stabilize a link between a transmitter and receiver. This usage of the term should not be confused with the usage in multi-hop networks, which often refers to simple forwarding.

In many cases wireless industrial communication systems will have to rely on commercially available wireless transceivers and have to use their hardware interfaces. This means that amplify-and-forward schemes are not usable, as commercial transceivers do not (easily) allow users to sample an analog waveform, nor to combine own samples with (digitally represented) samples of other nodes for purposes of joint decoding. It is, however, possible to use the ignorant-transceiver mode, when the hardware allows to switch off automated CRC checking (at the price of having to check the CRC in software later on).

In the following subsections we discuss some of the issues that arise in the design of practical wireless relaying protocols for industrial applications.

### *3.2 Controlling relayers and their activities*

In general, any node $R$ other than the source $S$ or the destination $D$ becomes a *relayer candidate* when it has received the packet once (from $S$ or from any other relayer that worked on the same packet). But how should it behave then?

The very first question is whether relaying is at all *desired*. For example, when a packet's deadline is very close or its retransmission budget has been exhausted, no relaying should happen, since otherwise a relayer's activities might interfere with any activities that the source starts after deadline expiration. The source must signal the relayer candidates to inhibit any activities. It is thus necessary to add header fields to the packets that are related to relaying. This inhibition information might be represented by a single bit, but other relaying-related information might require larger representations. This means that the packet size increases, and for the typical industrial case of short packets the increase might be substantial. The increase in packet size takes away channel bandwidth from other packets, allows fewer retransmissions within the given deadline, and bears the risk of a higher packet loss rate – larger packets are in general more susceptible to channel errors.

The second question is whether relaying is *required*. In the context of an ARQ protocol, relaying is not required when the destination sends out an immediate acknowledgement and the source receives it. How could a relayer check these conditions? And how could it be ensured that all relayers and the source see the same results, so that their actions do not interfere with each other? We refer to this issue as the *consistency issue*. One possibility is to let the source transmit a dedicated signaling packet when it has received the acknowledgement. However, consistency is harmed by loss of these packets. Another possibility is to let all nodes check for the presence of signal energy at the point in time where the receiver should send its acknowledgement. Lack of signal energy is interpreted as lack of acknowledgement and as a sign that relaying is required. However, the relayer candidate and the destination could be hidden terminals to each other, so that the relayer candidate does not sense a signal when the source node actually does. On the other hand, the presence of signal energy does not necessarily imply that: a) the energy belongs to an acknowledgement packet, since it could be interference from a co-located wireless system operating in the same frequency band; and b) the source has successfully received the acknowledgement. In all these approaches it is hard to achieve consistency.

Given that relaying is desired and required, the third question to ask is whether it is a good idea to have $R$ as a relayer or better use another node. This is the issue of *relayer quality*. The key question concerns the channel quality between $R$ and the destination. If this channel quality is constantly bad, then obviously $R$ is not particularly attractive as a relayer. If the channel fluctuates, then $R$'s qualities as a relayer are time-varying. This means that $R$ should constantly monitor the channel towards $D$ and maintain up-to-date *channel-state information*. In industrial applications the networks often have a centralized topology where a number of sensors transmit information to a central controller. In addition, the central controller frequently transmits packets, for example acknowledgements towards the sensors or poll packets to request data from the sensors. This arrangement can be nicely exploited, since it means that all the possible source nodes $S$ (i.e.: the sensors) continuously receive packets from their destination node (the central node) and hence can compute channel-state information. In other types of networks with more scattered communication relationships it is harder to obtain up-to-date channel state information to all possible destination nodes.

Furthermore, the notion of a "good relayer" is unnecessarily restrictive. It is intuitively clear and not hard to derive theoretically that there is a geographical region between source and destination (termed the "good region" in Figure 2) where it is beneficial to have a relayer. When a relayer is placed in the complement of this region it is actually harmful to use him, for example when the single relayer is even farther away from the destination than the source node is. However, even if the good region is empty, it might well be that there exists a chain of relayers among which a packet could be successfully forwarded (compare the lower part of Figure 2). Relaying protocols should possibly include the usage of such chains,

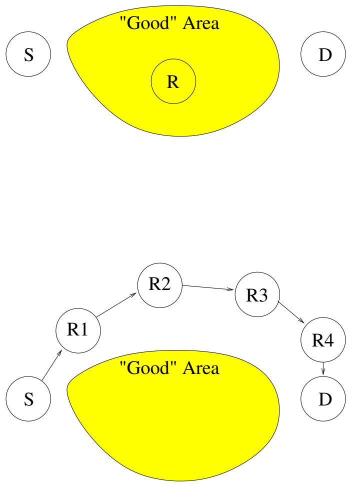



Fig. 2. Good relayer versus good chain

and the identification of good chains is a task similar to route setup in routing protocols.

Given that relaying is desired, required and that $R$ is a reasonable relayer, the fourth question is whether there are other good relayers as well and whether $R$ is *selected* to become a relayer. When the source transmits a packet, the broadcast property of the wireless medium might lead to a situation where multiple "good" relayers $R_1, R_2, \ldots, R_n$ receive the packet. The time-varying nature of the wireless medium implies that this "good-relayer subset" is of random size and varies over time. Without any coordination the relayed packets issued by $R_1, \ldots, R_n$ would collide at the destination, rendering the whole relaying procedure useless. A further complication is that the relayers might well be hidden nodes to each other, which is a problem for coordination schemes involving carrier-sensing operations. When there are multiple relayers, it should especially for fading channels be avoided to use one relayer all the time (its channel to the destination could also be in a deep fade), instead different relayers should be selected for subsequent retransmissions.

The coordination can be done in different ways. For example, the relayer candidates could use explicit *control packets* to select one among them. However, since such a procedure is susceptible to hidden-node situations and time-consuming in general (as compared to the packet deadlines), and since furthermore in industrial applications the control packets will often not be much smaller than the data packets, this kind of solutions appears to be wasteful and solutions without extra control packets are preferrable. Three fundamental approaches can be conceived for relayer selection: source-controlled relayer selection, relayer-controlled relayer selection and destination-controlled relayer selection.

In *source-controlled* relayer selection schemes 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. Examples of such information are:

- The source could include directly the MAC address of relayers. With this approach collisions and hidden terminal situations among relayers are naturally avoided, at the cost of flexibility and increased sizes of the MAC header. Furthermore, $S$ has to make some choice of relayers, and it could be a bad one. The choice of a relayer for $S$ can be preconfigured, or $S$ could try to *learn* about the best relayer. This learning problem is similar to the k-armed bandit problem, a standard benchmarking problem for reinforcement learning methods (Kaelbling *et al.*, 1996).
- The source can include other control fields. For example, when the relayers possess channel state information towards the destination, the source $S$ can include a threshold value which restricts the set of potential relayers to those having a channel quality which is better than the threshold. When the source notices collisions among relayers, it could increase the threshold for the next cycle.

In *relayer-controlled* relayer selection schemes each candidate decides itself about whether it retransmits the packet or not, without being explicitly controlled by any other network node. On the one hand, this class of strategies does not require $S$ to add additional MAC addresses to the packet header, on the other hand there is a higher potential for collisions at the destination, for hidden-terminal situations and consistency issues, since the set of relayers cannot as easily be controlled. To circumvent this, the relayers could use a MAC protocol with collision-avoidance features to reduce the contention among them. However, this contention resolution process again can take significant time (which in general depends on the node density) as compared to the length of short packets.

*Destination-controlled* relayer selection schemes can be an alternative in systems where much traffic goes to the same destination and where this destination polls the other nodes frequently – just like in many industrial communication systems. The destination computes channel-state information towards the sensors from the responses to polling packets and it can piggyback a relayer list onto frequently issued management packets. Whenever any of these pre-selected relayers receives a packet, it starts with relaying. However, the destination must not take packets coming from a relayer into account when updating the channel-state information belonging to the source node. To achieve this, the relayer must modify the packet. A minimal extension would be to use a single bit that marks the packet as a relayed one. A more significant extension adds the relayer MAC address to the packet, and as a result the destination can update the channel-state information belonging to the relayer. Another issue with destination-controlled schemes is the computational load that is put on the destination – this can be a significant burden especially in industrial cases where all traffic is directed to the central node.

The problem of relayer selection is discussed for example in (Nosratinia and Hunter, 2007).

### 3.3 Integration into existing ARQ schemes

In real wireless networks relaying schemes must be incorporated into practical protocols. This requires some adaptations to the ARQ protocols.

The ARQ protocol running on $S$ must be aware of the fact that one or more relayers transmit to $D$ after $S$ has finished its packet. This must be considered when $S$ decides about its timeouts for the arrival of acknowledgement packets and the point in time where $S$ starts its own retransmissions.

A more delicate issue concerns acknowledgement packets. Short data packets are not much longer than acknowledgements, and if on a fading channel a short data packet does not get through from source $S$ to destination $D$, then $D$'s acknowledgement packet does not have much better chances. Instead, relayers should also try to receive the acknowledgement packet from the destination and forward it back to the source.

## 4. A RELAYING FRAMEWORK FOR INDUSTRIAL TRAFFIC

In this section we present a simple relaying framework that is tailored to the case of small data packets and therefore useful for wireless industrial applications. Consistency issues are already avoided by construction.

### 4.1 Framework description

This framework adopts source-controlled relayer selection and does not require any lengthy contention process among relayer candidates. The source can use one or more relayers, or it can avoid the usage of relayers at all.

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* (separated by a small guard time for transceiver turnaround, processing times, etc.), 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 when they have picked it up.

More precisely, the framework operates as follows. 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's 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 $i \in \{1, \ldots, n\}$ the relayer $R_i$ listed at position $i$ possesses a correct copy of the packet, it simply transmits the packet in this slot. In addition, $R_i$ writes its 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 it and, using the slot number, it can check whether this slot has already passed or whether there is still a chance for it to transmit.

The $n$ data slots and the $n$ acknowledgement slots occur unconditionally – there is no additional carrier sensing by the relayers to check for the presence of an acknowledgement. This simplifies the design and eliminates consistency issues upfront. Furthermore, the lack of channel-sensing makes the protocol less vulnerable against interference from the outside. As a downside, however, when no relayer picks up the sources packet, the time for all the $n$ data slots and the $n$ acknowledgement slots is wasted. For this reason the framework is probably suboptimal for larger packet sizes.

### 4.2 Relayer selection

The source has any freedom to decide about $n$ and the relayers it wants to use. The choice of $n$ will in practice be determined by the packet deadline. The source schedules new rounds as long as the deadline is not expired.

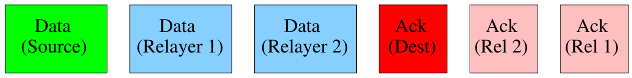



Fig. 3. Round structure for the smallpacket relaying framework ($n = 2$)

The choice of relayers is more delicate. When the deployment (i.e. the nodes, their geographical positions and mutual channel qualities) is static and known, then for each source node its set of relayers could be optimally configured. When the deployment is not known, a source could *learn* about relayers to use – more specifically, and adopting the terminology introduced in Section 3.2, a source should learn about good relayer chains of length $n$.[2] When a source has $m$ neighbor nodes, then a chain is an ordered selection of $n$ distinct out of $m$ neighbors of the source, and the total number of available chains given by $|C_{n,m}| = \frac{m!}{(m-n)!}$.

A learning scheme faces two difficulties: for larger neighborhoods $m$ the number of chains can become quite large (for example $|C_{2,40}| = 1560$, $|C_{3,40}| = 59280$ and $|C_{3,50}| = 117600$), and furthermore for each chain a number $T$ of tests must be made to have a reasonably reliable estimate of the quality of the chain. It would thus be practically unfeasible to test all chains and find the ones that really optimize the probability to successfully deliver a packet within a prescribed deadline. We therefore adopt two sup-optimal approaches, briefly described next, followed by the description of two non-learning schemes.

*4.2.1. Selection scheme 1: Find good nodes* This scheme tries only to find good nodes (i.e. nodes in the good region) and does not look for good chains. In the initial training phase only one relayer $R$ is used in each round ($n = 1$), for $T$ successive trials. The number $T$ should be large enough to achieve a reasonably accurate estimate of $R$'s relaying qualities. At the end of the training phase the source sorts its neighbors in descending order according to their relaying quality. In the following steady-state phase the number of relayers $n$ can be configured to a higher value and the source simply uses the $n$ best nodes as relayers.

*4.2.2. Selection scheme 2: A genetic algorithm* Genetic algorithms are a well-known approach to find local extrema in large search spaces (Goldberg, 1989). Our approach for identifying good chains works as follows:

- The algorithm works on a *population* of individual chains (each of length $n$), having a fixed *population size*. Initially, the population is selected at random.
- Each chain in the 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 available for the current population. The $\alpha \cdot 100\%$ of the 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 $\beta \cdot 100\%$ of the members of the new population are created from mutations of randomly chosen members of the $\alpha \cdot 100\%$ survivors. Specifically, to create a mutated member one survivor chain $c_s = (N_1, \ldots, N_n)$ is picked randomly and one of the neighbor addresses $N_i$ is changed randomly. The next $\gamma \cdot 100\%$ of the members of the new population are created from crossovers of the survivors. Specifically, two survivor chains $c_1 = (N_1, \ldots, N_n)$ and $c_2 = (M_1, \ldots, M_n)$ are picked randomly and a new chain is created as $c = (N_1, \ldots, N_{n/2}, M_{n/2+1}, \ldots, M_n)$. The remaining $(1 - \alpha - \beta - \gamma) \cdot 100\%$ of the members are randomly chosen from the set of all possible chains.

*4.2.3. Selection scheme 3: Random selection* This scheme is provided as a baseline scheme. Specifically, for a given $n$ the source node selects $n$ distinct out of its $m$ neighbors at random.

*4.2.4. Selection scheme 4: Preconfigured with helper nodes* For some applications learning a good node or a good chain might not be an option, but instead a source node could be pre-configured with a list of relayers to use. Such a scheme can be useful when a number of helper nodes is placed around the central controller, which are not part of an industrial control system but which only help with relaying packets for other nodes.

[2] It appears that most of the industrial MAC- and link-layer protocols do not encompass learning phases. At first sight, learning phases appear to be unproductive time, but they could be put into work breaks or phases of machine maintenance. Another obvious criticism concerning learning schemes is that they are of limited value in the face of mobility and non-stationary wireless channels. This needs to be assessed in the future.

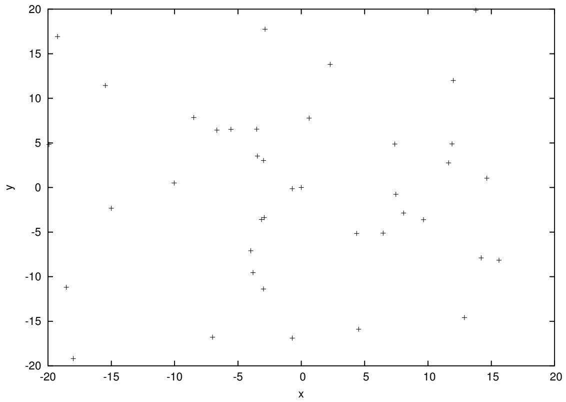



Fig. 4. Investigated node deployment (40 nodes, including one central controller in the middle of the square area)

### 4.3 Performance results

We provide here some exemplary performance results for the previously described relaying framework. The adopted performance measure is the probability of delivering a packet successfully within a prescribed deadline of 10ms from a source $S$ to the destination $D$ – this is called the *success probability*. We have chosen a two-dimensional deployment with a centralized controller, to which all packets are directed, and a number of source nodes which want to transmit packets to the controller. All nodes are placed in a square of $40 \times 40$ meters, centered at the origin, where also the central controller is placed. There are 40 nodes in total, including 39 source nodes. The source nodes are randomly placed in the square. The placement is for reference shown in Figure 4.

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 30,000 packets is simulated. Queueing effects and medium access control are not considered.

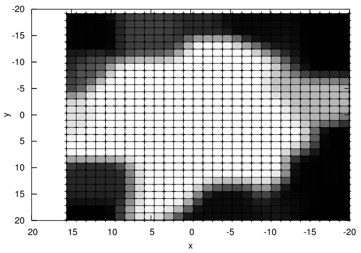
The channel model is very simple. Between each pair of nodes a separate wireless channel exists, which is stochastically independent of all other channels. Each channel $c$ is a binary symmetric channel (BSC), i.e. each bit on this channel is, independently of other bits, erroneous with a certain fixed probability $p_c$. The probability $p_c$ depends on the geographical distance between the two nodes through the path loss (the path loss exponent is three), and on the chosen modulation scheme (coherent BPSK). The relevant physical layer parameters are based on existing IEEE 802.15.4 transceivers (Chipcon, 2004). The data rate is 250 kbit/s, and the transceiver turnover time between transmit and receive modes corresponds to 40 bit times.

The other important parameters have been chosen as follows: for all selection policies except the "preconfigured" policy the number $n$ of relay slots is $n = 4$, the number of test trials per chain or per node is $T = 30$, the width of a relayer MAC address is 8 bits (each relayer's MAC address must be added to the packet), 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. For the preconfigured selection scheme we have assumed four helper nodes at positions $(0, 6)$, $(0, -6)$, $(6, 0)$ and $(-6, 0)$, and each source was configured with the closest two ($n = 2$) of these.

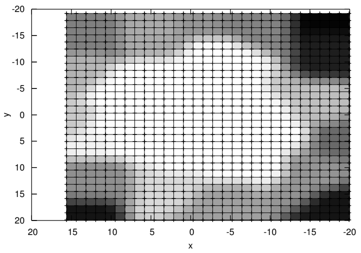
In Figures 5 and 6 we compare the success probabilities of the different relayer selection schemes and the pure ARQ scheme (i.e. a scheme where the source makes immediate retransmissions without using any relayer) by means of density plots. In the figures, brighter areas indicate a higher success probability, darker areas a smaller one. It should be noted that for visualization purposes the plots are smoothed. The results indicate that relaying clearly pays out in areas where the pure ARQ scheme works badly. In addition, in those areas where pure ARQ is doing very well, the relaying schemes do not loose performance. Visually, the genetic selection scheme slightly outperforms the find-good-nodes scheme, and these two outperform all other schemes. This is substantiated by the finding that the average success probability (taken over all source nodes) for the find-good-nodes scheme is $\approx 0.93$ (with minimum observed success probability of 0), for the genetic scheme it is $\approx 0.96$ (minimum 0.16), whereas the pure ARQ scheme has an average of $\approx 0.64$ (minimum 0) and the preconfigured scheme has an average of 0.81 (minimum 0). The advantage of the genetic scheme is due to its ability to find good chains when there are no good relayers. While not shown here, this advantage of the genetic scheme is also confirmed when many independent instantiations of the random source positions are investigated and becomes more pronounced for lower node densities.

## 5. CONCLUSIONS

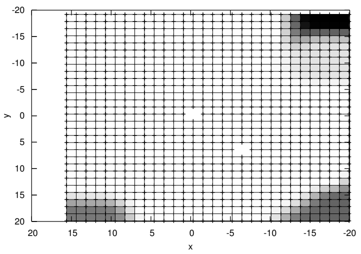
In this paper we have explored the usage of spatial diversity schemes, and in particular relaying, in industrial settings. It became apparent that the properties of industrial traffic (especially the small packet size) can successfully shape the design of protocols and that indeed significant improvements in terms of success probabilities can be



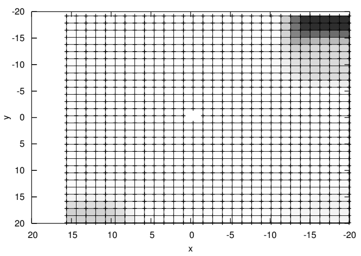
**(a)** Pure ARQ



**(b)** Random selection



**(c)** Find good nodes

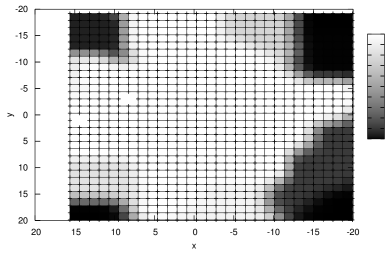


**(d)** Genetic

Fig. 5. Comparison of selection policies for their success probability

made. It should be noted that the proposed relayer selection schemes (and their parameters) are *not* the result of a long phase of experimentation and optimization but are more an "educated guess" of good schemes and parameters, and significant refinements and performance improvements should be possible.

More generally, the author is convinced that spatial diversity is a key component for the design



**(a)** Preconfigured

Fig. 6. Comparison of selection policies (cont'd)

of protocols for wireless industrial systems and should not be omitted from future standards.

## REFERENCES


Biglieri, Ezio (2005). *Coding for Wireless Channels*. Springer. New York.

Biglieri, Ezio, John Proakis and Shlomo Shamai (1998). Fading channels: Information-theoretic and communications aspects. *IEEE Transactions on Information Theory* **44**(6), 2619–2692.

Bölcskei, Helmut (2006). Mimo-ofdm wireless systems: Basics, perspectives and challenges. *IEEE Wireless Communications* **13**(4), 31–37.

Chipcon (2004). *2.4 GHz IEEE 802.15.4 / ZigBee-ready RF Transceiver*. Chipcon Products from Texas Instruments.

Cover, Thomas M. and Abbas A. El Gamal (1979). Capacity theorems for the relay channel. *IEEE Transactions on Information Theory* **25**(5), 572–584.

Cover, Thomas M. and Joy A. Thomas (2006). *Elements of Information Theory*. second ed.. John Wiley & Sons. New York.

Cui, Shuguang, Andrea J. Goldsmith and Ahmad Bahai (2004). Energy-efficiency of mimo and cooperative mimo techniques in sensor networks. *IEEE Journal on Selected Areas in Communications* **22**(6), 1089–1098.

del Coso, Aitor, Umberto Spagnolini and Christian Ibars (2007). Cooperative Distributed MIMO Channels in Wireless Sensor Networks. *IEEE Journal on Selected Areas in Communications* **25**(2), 402–414.

Diggavi, Suhas N., Naofal Al-Dhahir, A. Stamoulis and A. R. Calderbank (2004). Great Expectations: The Value of Spatial Diversity in Wireless Networks. *Proceedings of the IEEE* **92**(2), 219–270.

Goldberg, David E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison Wesley. Bonn, Germany.

Hanzo, Lajos, Jason P. Woodard and Patrick Robertson (2007). Turbo Decoding and Detection for Wireless Applications. *Proceedings of the IEEE* **95**(6), 1178–1200.

Kaelbling, Leslie Pack, Michael L. Littman and Andrew W. Moore (1996). Reinforcement Learning: A Survey. *Journal of Artificial Intelligence Research* **4**, 237–285.

Kramer, Gerhard, Michael Gastpar and Piyush Gupta (2005). Cooperative strategies and capacity theorems for relay networks. *IEEE Transactions on Information Theory* **51**(9), 3037–3063.

Laneman, J. Nicholas, David N. C. Tse and Gregory W. Wornell (2004). Cooperative diversity in wireless networks: Efficient protocols and outage behaviour. *IEEE Transactions on Information Theory* **50**(12), 3062–3080.

LAN/MAN Standards Committee of the IEEE Computer Society (1999). *Information technology – Telecommunications and Information Exchange between Systems – Local and Metropolitan Area Networks – Specific Requirements – Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications.*

LAN/MAN Standards Committee of the IEEE Computer Society (2006). *IEEE Standard for Information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Specific requirements – Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low Rate Wireless Personal Area Networks (LR-WPANs).* revision of 2006.

Liew, T. H. and Lajos Hanzo (2002). Space-Time Codes and Concatenated Channel Codes for Wireless Communications. *Proceedings of the IEEE* **90**(2), 187–219.

Nosratinia, Aria and Todd E. Hunter (2007). Grouping and Partner Selection in Cooperative Wireless Networks. *IEEE Journal on Selected Areas in Communications* **25**(2), 369–378.

Paulraj, Arogyaswami J., Dhananjay A. Gore, Rohit U. Nabar and Helmut Bölcskei (2004). An Overview of MIMO Communications – A Key to Gigabit Wireless. *Proceedings of the IEEE* **92**(2), 198–218.

Scaglione, Anna, Dennis L. Goeckel and J. Nicholas Laneman (2006). Cooperative communications in mobile ad hoc networks. *IEEE Signal Processing Magazine* pp. 18–29.

Tse, David and Pramod Viswanath (2005). *Fundamentals of Wireless Communications.* Cambridge University Press. Cambridge, UK.

Willig, Andreas (2003). Polling-based MAC Protocols for Improving Realtime Performance in a Wireless PROFIBUS. *IEEE Transactions on Industrial Electronics* **50**(4), 806 –817.

Willig, Andreas, Kirsten Matheus and Adam Wolisz (2005). Wireless Technology in Industrial Networks. *Proceedings of the IEEE* **93**(6), 1130–1151.

Zheng, Lizhong and David N. C. Tse (2003). Diversity and multiplexing: A fundamental tradeoff in multiple-antenna channels. *IEEE Transactions on Information Theory* **49**(5), 1073–1096.

Zhu, Hao and Guohong Cao (2006). rdcf: A relay-enabled medium access control protocol for wireless ad hoc networks. *IEEE Transactions on Mobile Computing* **5**(9), 1201–1214.