Decoding-based channel estimation for selective cooperation diversity protocols

Stefan Valentin*, Tobias Volkhausen*, Furuzan Atay Onat†, Halim Yanikomeroglu†, and Holger Karl*
*University of Paderborn, Germany, {stefan.valentin|volk|holger.karl}@upb.de
†Carleton University, Ottawa, Canada, {furuzan|halim}@sce.carleton.ca

***Abstract*— We describe and analyze Minimum Path Difference (MPD), a metric to improve threshold-based Selection Decode-and-Forward (SDF) protocols in cooperative wireless networks with channel coding. By observing the decoding process, MPD provides a more frequent channel quality estimation than CRC or SNR and, thus, allows SDF to forward only the *parts* of a message that are likely to be correct. In this paper, we describe MPD's efficient integration into Viterbi decoders and SDF protocols, analyze its accuracy and threshold selection, and study its end-to-end Bit Error Rate in practical cooperative WLANs where ideal block fading and ideal channel codes cannot be assumed. Our studies validate that, unlike all prior methods, in these practical scenarios, MPD significantly improves SDF's performance and experiences only a marginal performance penalty if suboptimal, constant thresholds are selected.**

I. Introduction

Cooperative relaying can provide substantial diversity gains by exploiting the broadcast nature of the wireless channel even when multiple antennas per node are not applicable. To organize the relaying process, cooperation protocols are required that achieve user cooperation diversity [1] as a special form of spatial and temporal transmit diversity. In their seminal work [2], Laneman et al. show that so-called Selection Decode-and-Forward (SDF) protocols can reach full diversity (in order of cooperating nodes) if the relay perfectly prevents error propagation by forwarding only correct messages.

This ideal case, where the relay always makes a perfect forwarding decision, was assumed in many follow-up papers. It implies (1) ideal channel coding, meaning that an outage (in terms of SNR falling below a threshold) can be directly translated into a decoding error at the relay, and (2) message-wise block-fading (i.e., constant channel state during a message's transmission). While these assumptions provided tremendous insight in the performance regions of SDF protocols, they are not adequate for actually *achieving* these bounds in practice.

First, in practical systems, ideal channel codes cannot be assumed. Consequently, an outage does not directly map to a decoding error and SNR at best roughly estimates the Bit Error Rate (BER) of decoded messages [3]. Second, in realistic mobile scenarios, fading is a continuous time-correlated process where the channel state may change at any time [4]. Consequently, message-wise block fading cannot be assumed. Instead, the channel may vary several times per message; causing errors only in parts of a message.

So far, many papers assume that the relay uses either Cyclic Redundancy Check (CRC) [2] or an SNR-threshold [5, 6] to decide if there are any bit errors in a received message. Both methods show considerable shortcomings under the above realistic assumptions. While CRC-based decision performs excellent with message-wise block fading, it becomes ineffective if only parts of a message are likely to be corrupt. In such a case, a CRC-based SDF protocol ineffectively drops the *complete* message even though *partial forwarding*, i.e., forwarding the correct parts of the message, would improve the end-to-end performance at the destination.

Forwarding only these correct parts requires frequent error tests inside a single message. But such a frequent, at best, continuous observation must not decrease the data rate. This is not possible with either SNR or CRC: CRC is efficient only when calculated over large message blocks, as a checksum adds overhead to each block and CRC's detection rate decreases for shorter blocks [7]. Measuring SNR comes at the cost of training symbols reducing throughput. This makes frequent SNR measurement infeasible which is, therefore, only performed at the beginning of each message in most systems [8]. With this practical limitation, SNR can neither characterize all parts of a message nor, as discussed above, can accurately estimate decoding errors with practical Forward Error Correction (FEC) codes (cf. Sec. III of this paper).

To this end, we proposed Minimum Path Difference (MPD) as a *decoding-based* metric for continuous error estimation in [9]. MPD estimates decoding errors by only observing changes in the FEC-redundancy of a received message. As in trellis-coded systems this redundancy is already interleaved over the complete message, MPD estimates errors even continuously for each code symbol without decreasing the data rate by adding training symbols or checksums. By observing several code symbols, MPD provides an accurate, continuous BER estimation for each small message block similar to the A Posteriori Probability (APP) provided by ideal MAP decoders [10]. Unlike APP, however, it merely uses standard decoding functions of cellular and Wireless Local Area Network (WLAN) receivers and can thus be implemented without significantly increasing complexity.

While we introduced the basic idea of MPD in a preliminary paper [9], in the present paper we extend this idea, describe implementation details of this metric, and extensively analyze its properties when applied in threshold-based cooperation protocols. In particular, we (1) provide an algorithmic description and implementation details for MPD's efficient integration into practical hard and soft-decision Viterbi decoders (Sec. III-A),

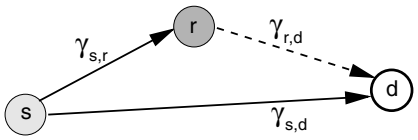

Fig. 1. Cooperation scenario with three single antenna nodes s,r,d and the *instantaneous* SNR values γ for the half-duplex channels used during phase 1 (solid line) and 2 (dashed line) of a cooperation cycle.

(2) study MPD's error estimation accuracy for messages and small message blocks (Sec. III-B), (3) show how MPD can support partial forwarding in simple threshold-based SDF protocols without decreasing data rate due to signaling overhead (Sec. IV-A), (4) study MPD's expression of SDF's end-to-end BER (BER_{e2e}), (5) use this expression to select ideal and heuristic MPD thresholds (Sec. IV-B), and (6) compare SDF's performance with these thresholds to other practical and ideal variants of SDF (Sec. V). These studies show that, unlike SNR and CRC-based decision, MPD-based SDF does not suffer under realistic fading and coding assumptions. Instead, even with suboptimal thresholds it closely reaches the theoretical end-to-end performance bound in practical cooperative WLANs.

II. System model and basic SDF operation

We consider the scenario in Fig. 1 where a single relay node r may employ SDF to cooperate with source node s for reaching the destination node d. For comparison, direct transmission from s to d is considered. Each node is equipped with a single omnidirectional antenna and may perform coding and decoding of messages. As in IEEE 802.11a/g transceivers, we assume the following coding procedure: First, an (inner) CRC code is applied and the resulting checksum is appended. Second, the resulting message is encoded using an (outer) convolutional FEC code with rate R_c. Finally, the resulting coded message, called codeword c, is modulated and transmitted.

Each transmission cycle is split into two phases. In the first phase, s encodes its message m and broadcasts the resulting codeword c over channels (s,r) and (s,d) to r and d. Relay r now demodulates the signal received from channel (s,r) and obtains codeword $c_{s,r}$. After decoding, the possibly erroneous message $m_{s,r}$ is available at the relay. With SDF protocols, in phase 2, r may either forward a newly encoded version of $m_{s,r}$ to d via channel (r,d), remain silent, or transmit other, unrelated data. This forwarding decision depends on an estimate of the transmission errors in $m_{s,r}$. In this paper, we study several metrics to be used for this estimate. For example, as $m_{s,r}$ is CRC-coded, r can test the complete message for errors not corrected by FEC decoding. If the message contains no such errors, r encodes, modulates, and forwards $m_{s,r}$ in phase 2. In this case, d can receive and decode the resulting codeword $c_{r,d}$ to message $m_{r,d}$. Finally, d achieves diversity gain by combining this message's signal with the signal of message $m_{s,d}$, received during phase 1.

To combine the signals received during both phases at d, we assume the common Maximum Ratio Combining (MRC) scheme [3]. For all nodes, we consider Physical layer (PHY) functions and parameters according to IEEE 802.11a WLAN transceivers [8]. At the transmitters, we assume a standard CRC-32 code polynomial, convolutional FEC coding with generator polynomial $g_0 = 133_8; g_1 = 171_8$, code rate $R_c = 1/2$, and $4\,\mu$s BPSK modulation symbols. On the receiver side, coherent detection and Viterbi decoding are assumed. Furthermore, we assume a constant message flow and a medium access control scheme assuring an orthogonal channel (e.g. a separate time slot) for each cooperation phase.

All studied channels reflect a non-line of sight situation with moving nodes. We consider frequency-flat, half-duplex, i.i.d. Rayleigh fading channels where small-scale fading is parameterized by a maximum Doppler shift according to a velocity v. Instead of assuming block-fading per message, we model fading *per modulation symbol* using the common "Jakes-like" method with the "land mobile" autocorrelation function (Table 2.1 in [4]). This widely-used model is suitable for mobile indoor or urban scenarios with many stationary, uniformly distributed scatterers.

III. Decoding-based channel estimation with MPD

We define Minimum Path Difference (MPD), describe its integration into hard and soft-decision Viterbi decoding, and show that its BER estimation is more accurate than SNR measured in WLAN receivers and close to the ideal case.

A. MPD definition and Viterbi decoder integration

MPD estimates the BER by observing the trellis decoding process of a received codeword $c_{s,r}$ and can be calculated using only the already present FEC redundancy. Specifically, MPD expresses the minimum distance of $c_{s,r}$ to the closest valid codeword in the trellis diagram. This expression includes the full decoding history and is calculated over all symbols of a codeword while traversing the path with the minimum weight in the trellis structure. Since this path is found and traced-back during normal Viterbi decoding, MPD can be calculated *during* the second step (traceback) of this standard algorithm. For hard-decision Viterbi decoding, this calculation can be illustrated by the example in Fig. 2. Here, an $R_c = 1/3$-coded message c is transmitted via channel (s,r), received as $c_{s,r}$ (including a channel-dependent error term $e_{s,r}$), and trellis-decoded. Fig. 2 illustrates the end of this decoding process where the Hamming distance already weights each trellis edge and the weight-minimizing path is already found and traversed-back (bold line in Fig. 2). During this traceback, a standard Viterbi decoder returns the decoded message $m_{s,r}$. Additionally, our MPD-extended decoder returns a path weight per received code symbol as vector $mpd_{s,r}$ (dashed arrows in Fig. 2). For hard-decision decoding, this vector contains the Hamming distance per code symbol, thus, representing the number of errors for each successfully decoded symbol in $c_{s,r}$.

We now formally describe MPD's calculation and generalize it for soft-decision variables (aka soft-bits). For hard *and* soft-decision decoding, Algorithm 1 shows a condensed version

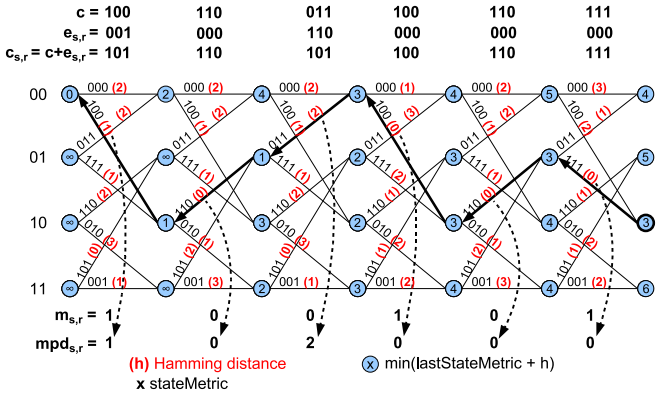

Fig. 2. Example of MPD-extended hard-decision Viterbi decoding of the received codeword $c_{s,r}$. Each trellis edge is weighted by the Hamming distance (in brackets) which is returned per code symbol in vector $mpd_{s,r}$.

Algorithm 1: MPD-extended Viterbi decoding

Input: Codeword $c_{s,r}$ with u code symbols: $c_1,\ldots,c_u$;
Codeword $\tilde{c}_{s,r}$ with u code symbols: $\tilde{c}_1,\ldots,\tilde{c}_u$
Output: Message $m_{s,r}$ with u message symbols: $m_1,\ldots,m_u$;
Metric values $mpd_{s,r}$ per code symbol: $mpd_1,\ldots,mpd_u$

```
// Viterbi (1): Search minimum-weight path
1 edge_{1,...,u} = findPath(c_{s,r});
// Viterbi (2): Traceback over path
2 for i = u,...,1 do
3     m_i = messagesymbol(edge_i);
      // MPD calculation adds line 4
4     mpd_i = diff(c̃_i, codesymbol(edge_i));
5 end
6 return m_{s,r}, mpd_{s,r}
```

of our MPD-extended Viterbi algorithm[1]. The algorithm performs Viterbi decoding of a message m encoded at rate $R_c = k/n$ with n coded bits per k message bits. In total, message $m_{s,r}$ consists of $u = l/k$ message symbols or l (uncoded) message bits and is decoded from codeword $c_{s,r}$, which consists of u code symbols or l/R_c code bits. Viterbi decoding is performed in two steps: First, the algorithm searches for the trellis path which minimizes the accumulated weight (function findPath() in line 1). Second, for all u edges of this path, a traceback is performed (line 2–5) and one k-bit message symbol is returned per edge (function messagesymbol() in line 3). Finally, the decoded message $m_{s,r}$ is returned.

MPD's calculation is integrated into the second step of Viterbi decoding. Here, the traceback iterates over the complete weight-minimizing path and MPD can be calculated per code symbol (line 4). As described above, with hard-decision decoding the Hamming distance can be used to express the distance between a received code symbol to a symbol from the edge of the trellis. Here, MPD can be directly calculated from $c_{s,r}$. With soft-decision decoding, $\tilde{c}_{s,r}$ is required as an offset-free copy representing codeword $c_{s,r}$ as *normalized* soft-bits $\in [-1,1]$. For these normalized soft-bits, the Euclidean distance can be calculated by $\text{diff}(a,b) = \sum_{j=0}^{n}(a_j - b_j)^2$, where a_j stands for one of n soft-bits in symbol a from the received codeword and b_j corresponds to one of n soft-bits in code symbol b from the trellis edge (as returned by function codesymbol() in line 4). After this distance is obtained for all symbols of the weight-minimizing path, Algorithm 1 returns vector $mpd_{s,r}$ containing one value for each symbol of $\tilde{c}_{s,r}$.

[1]For clarity, implementation details like quantization and pipelining are omitted. Details of Viterbi decoding are provided in standard literature [3].

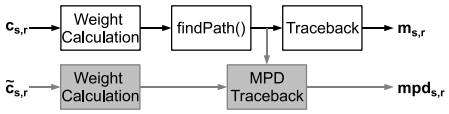

Fig. 3. Basic steps of the MPD-extended Viterbi decoding algorithm.

Fig. 3 summarizes our MPD extensions to the soft-decision Viterbi algorithm. As shown, Algorithm 1 decodes $c_{s,r}$ (white blocks) and calculates MPD of $\tilde{c}_{s,r}$ (shaded blocks) in parallel. Hence, although soft-decision decoding requires an additional codeword $\tilde{c}_{s,r}$ for MPD calculation, this does not significantly increase decoder complexity. Due to the parallel formulation of Algorithm 1, MPD calculation adds no iterations over the trellis and even an MPD-extended Viterbi algorithm has still complexity $\mathcal{O}(u)$. A further important observation for applying MPD is that an MPD value is calculated for *each* decoded symbol. In theory, this enables *symbol-wise* adaptation (e.g. an SDF decision) but in practice "smoothed" MPD values averaged over blocks of N symbols would be used.

B. MPD accuracy study

We now study the accuracy of MPD's BER estimation for transmitting over channel (s,r) in a vehicular WLAN scenario (Sec. II). We focus on two cases for MPD calculation. First, MPD is averaged over a complete message providing a single estimation per message. Second, multiple MPD values per message can be provided by calculating MPD over many small blocks of a message. We select 4 Byte-blocks as after this period the trellis path becomes stable [3] for the applied code (Sec. II) and called this case *block(4)-MPD*. We compare this block and message-wise MPD calculation to the ideal case where SNR is *continuously* measured for each message symbol. In this best yet unrealistic case for SNR measurement, called *ideal* $\gamma_{s,r}$, *each* message symbol represents a training symbol and, thus, no data is transmitted. Further, we compare MPD to a realistic metric, where $\gamma_{s,r}$ is extracted only from the first 4 preamble symbols of a message. This corresponds to SNR measurements in IEEE 802.11a/g WLAN transceivers [8] and will be called *realistic* $\gamma_{s,r}$.

To analyze the accuracy of these metrics we study the real number of errors in $c_{s,r}$ (equivalent to the actual BER of this message or its blocks) as a function of the observed metric values. For each metric, we illustrate this expression as a scatter plot in Fig. 4. Each plot shows a single point for each single 500 Byte message or, with *block(4)-MPD*, for each 4 Byte message block; in total 16×10^3 messages were transmitted. The results show how accurately the studied

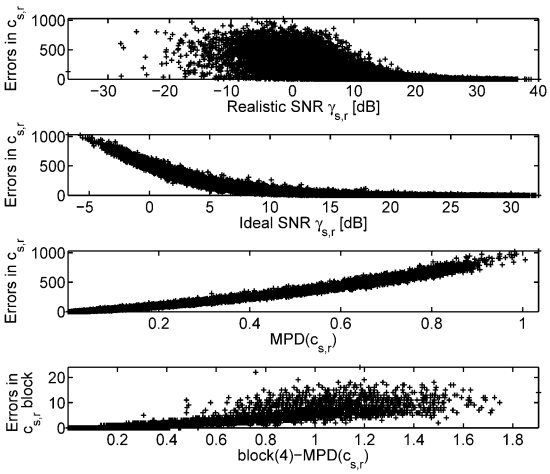

Fig. 4. Scatter plots for the no. of bit errors in a received message $c_{s,r}$ as expressed by SNR and MPD-based metrics. Each metric is observed during the same transmission via channel (s,r) at $v = 20$ m/s and mean SNR $\in [0,30]$ dB.

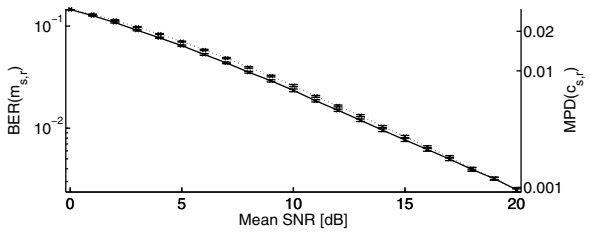

Fig. 5. Mean MPD of $c_{s,r}$ (dotted line) compared to mean BER of $m_{s,r}$ (solid line) for channel (s,r). Confidence intervals are shown for 95%.

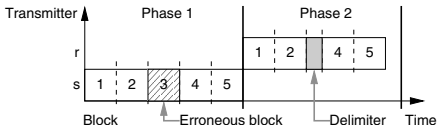

Fig. 6. Example of a partial forwarding cooperation cycle: Block 3 is received erroneously and, thus, not forwarded but replaced by a short delimiter.

metric estimates the number of errors. An ideal metric would clearly express the number of bit errors as a function of metric values. Realistic $\gamma_{s,r}$ shows a high standard deviation and can, thus, not accurately express a distinct number of bit errors. As expected with a continuously fading channel, this metric shows no similarity to ideal $\gamma_{s,r}$ and no clear structure. In consequence, realistic $\gamma_{s,r}$ is not a useful indicator of message quality. This situation is completely different for MPD. With message-wise MPD, the scatter plot falls in a very small region, i.e., has low standard deviation, very similar to ideal SNR. Moreover, message-wise MPD and number of bit errors can be (almost) injectively mapped onto each other by virtue of monotonic increase. Block-wise MPD, however, is not calculated over all message symbols and, thus, is less similar to the ideal $\gamma_{s,r}$ case. Nevertheless, compared to realistic $\gamma_{s,r}$, block-wise MPD still reaches high accuracy and, unlike all other metrics, provides many BER-estimates per message.

MPD's high accuracy results from the fact that each MPD value is averaged over many (continuously observed) samples. In addition to this statistical benefit, MPD is calculated *during* FEC decoding and, thus, takes the actual decoding status into account. None of these benefits are achieved with realistic $\gamma_{s,r}$ as it is measured only over a short part of a message. Due to these benefits, Fig. 5 shows a close qualitative match of BER and message-wise MPD averaged over all messages in the above scenario. Hence, MPD is an excellent estimator for the actual number of errors in a decoded message as well as for the BER of many messages. While message and block-wise MPD enables more accurate adaptation decisions than realistically measurable SNR, block-wise MPD also enables even multiple adaptations per message.

IV. Using MPD in threshold-based SDF protocols

In this section, we discuss how to apply MPD in threshold-based SDF cooperation protocols to support partial forwarding and how to select the ideal MPD thresholds.

A. MPD-based SDF with partial forwarding

It is straightforward to profit from MPD's accurate BER estimation by simply applying it instead of SNR in threshold-based SDF protocols. In this case, the relay makes only a single threshold-based forwarding decision per message; now based on the MPD averaged over all u message symbols.

However, even more decisions per message can be made by averaging MPD over small blocks of a message (Sec. III). For each of these blocks the relay may choose a length $N \in [1,u]$, calculate the corresponding MPD value, and make an independent, MPD threshold-based forwarding decision. This allows *partial forwarding* of messages, which is beneficial in scenarios where fading does not equally affect an entire message or the relay may reconstruct parts of a completely faded message by FEC. According to its threshold, this cooperation protocol, called *MPD-based SDF*, drops blocks that are likely to be erroneous while still forwarding correct blocks (Fig. 6).

To properly combine the signals received from s and r, the destination requires to know which blocks were dropped. The relay can signal these blocks to the destination by replacing each dropped block by a delimiter (Fig. 6). As shown, using delimiters does not increase the length of the message if block length N is selected such that it is always larger than the delimiter. Compared to a complete retransmission, this method never decreases the data rate as delimiters are "hidden" inside the dropped blocks.

B. Selecting ideal MPD thresholds

Improving $\mathrm{BER_{e2e}}$ with a threshold-based SDF protocol requires to select thresholds which are not too aggressive, i.e., by forwarding even incorrect messages, and not too conservative, i.e., by erroneously dropping correct messages. Although SDF's local forwarding decision cannot actually *minimize* the global $\mathrm{BER_{e2e}}$, an ideal SDF threshold avoids any $\mathrm{BER_{e2e}}$ increase caused by the relay. Selecting such a threshold depends on (1) how the metric expresses decoding

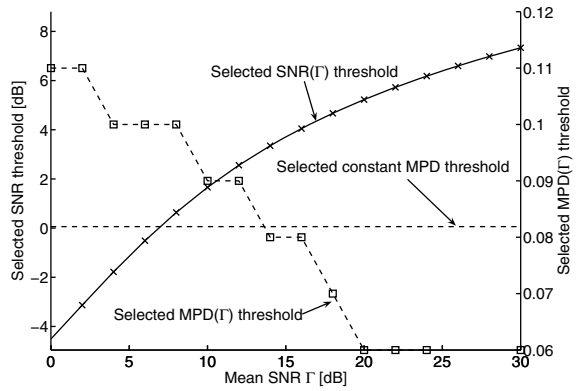

Fig. 7. Selected SNR thresholds (solid line) and MPD thresholds (dashed lines) either depending on mean SNR Γ or constant. Example for $v = 20$ m/s; Threshold values are *not* quantized

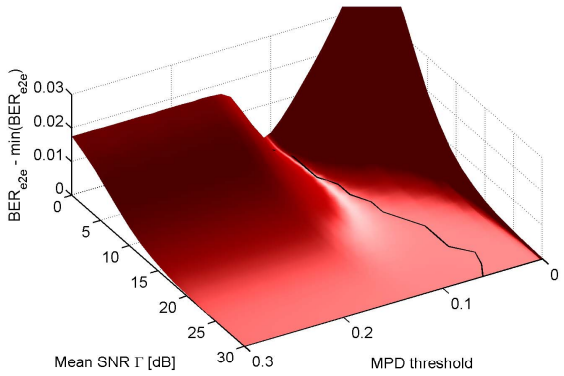

Fig. 8. Ideal MPD thresholds (black line) selected as the Γ-dependent MPD threshold set which minimizes BER_{e2e}. Example for $v = 20$ m/s, 4 Byte blocks, and 10^5 transmitted messages; Threshold values are *not* quantized

errors and (2) if the metric is a function of further channel and system parameters. We now study the effect of these two characteristics on the selection of ideal MPD thresholds by comparing (1) ideal SNR to ideal MPD thresholds and (2) ideal MPD thresholds to a heuristic, constant MPD threshold.

Fig. 7 shows the MPD and SNR thresholds selected for the vehicular scenario in our end-to-end performance study (Sec. V). We study a symmetric situation where all links have equal mean SNR Γ and assume Γ to be known for threshold selection. Selecting SNR thresholds in such situations is discussed in [6] in detail. Based on this BER analysis, we obtained the shown SNR thresholds using Eq. (3), (14), and (22) in [6]. The resulting Γ-dependent SNR thresholds are denoted by SNR(Γ). The selected thresholds for MPD also depend on Γ and are denoted by MPD(Γ). Unlike SNR(Γ), MPD(Γ) always decreases with Γ as MPD expresses the number of errors instead of signal power.

We select the MPD(Γ) thresholds using a large set of MPD values calculated for many received message blocks. From this set, for each Γ, the MPD value providing the smallest BER_{e2e} is chosen. An example of this selection is illustrated in Fig. 8 showing an offset-free BER_{e2e} at the vertical axis for clearer presentation. As shown, for low Γ the ideal MPD thresholds (black line) can be found at the bottom of a clearly shaped BER_{e2e} "valley". This valley represents the threshold a local forwarding decision should choose to obtain the lowest achievable BER_{e2e} by forwarding the most correct blocks. If Γ increases, more blocks are correct thereby reducing the impact of an accurate forwarding decision. Here, the flattened valley allows a larger set of different ideal MPD thresholds. Finally, for large Γ ($\Gamma \geq 20$ dB in this example), the valley becomes a pure plane and any MPD threshold can be ideal. This behavior of MPD(Γ) simplifies threshold selection compared to SNR(Γ), where each threshold strongly depends on Γ (Fig. 7). For MPD(Γ) instead, with increasing Γ the actual threshold value becomes less significant to minimize BER_{e2e}. Further, even selecting an MPD threshold for a *region* of Γ can be still ideal as the BER_{e2e} valley does not flatten continuously vs. Γ (Fig. 7). Consequently, even a constant MPD threshold can considerably improve BER_{e2e} in practical setups (Sec. V).

V. End-to-end performance study

We study the effect of the decision metric and threshold selection on the end-to-end BER (BER_{e2e}) performance of SDF cooperation protocols. In particular, we compare the following metrics: Message-wise *CRC* as assumed in [2], message-wise *realistic* $\gamma_{s,r}$ measured over 4 preamble symbols as in WLAN receivers, symbol-wise *ideal* $\gamma_{s,r}$ with ideal yet unrealistic SNR measurement as in [5, 6], and MPD with block-wise calculation and decision. We chose 4 Byte MPD blocks for sufficient decoder stability (Sec. III-B) and to avoid decreasing data rate due to signaling (Sec. IV-A).

For all above metrics except CRC, the ideal MPD(Γ) and SNR(Γ) thresholds are selected as in Sec. IV-B. For MPD, we study if selecting the Γ-dependent threshold is worth the effort by comparing it to the constant MPD threshold from Sec. IV-B. Furthermore, we include *direct* transmission and *genie* SDF in our study as a lower and upper BER_{e2e} performance bound, respectively. Unlike ideal $\gamma_{s,r}$, where only the local channel estimation at the relay is ideal, genie SDF always makes a perfect forwarding decision as the "genie" perfectly knows which symbols to forward to actually decrease BER_{e2e} at d.

For these cases we show BER_{e2e} results *prior* to decoding to allow comparison to studies for uncoded cooperation systems [2, 5, 6]. For our cooperative WLAN scenario we choose standard IEEE 802.11a parameters (Sec. II) and a constant data stream where each 500 Byte message is appended by a CRC checksum and, then, FEC coded at rate $R_c = 1/2$. Per message, this results in 8000 Bits transmitted at 6 MBit/s using BPSK modulation. All i.i.d. Rayleigh channels are parameterized with the same mean SNR and the simulations were performed at *symbol level*, i.e., a channel state change affects at least one modulation symbol. Two different fading scenarios are studied: The first *indoor* scenario represents a situation with low mobility, i.e., highly time-correlated fading channels, slowly varying according to a Doppler shift $f_d = 17.34$ Hz. At the carrier frequency of 5.2 GHz this corresponds to a maximum

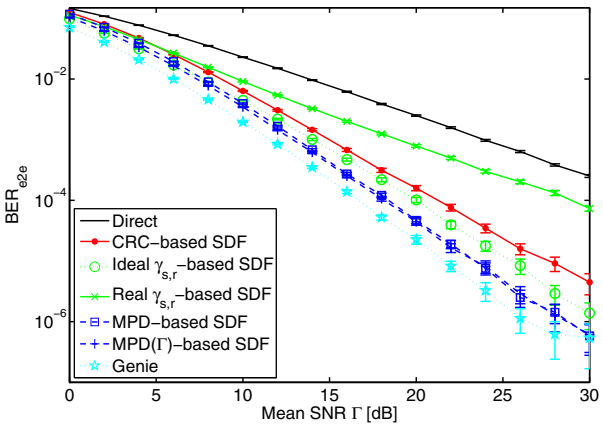

Fig. 9. End-to-end BER vs. mean SNR with direct transmission and different SDF metrics; $v = 1$ m/s, 10^5 transmitted messages per shown mean value, and 95% confidence intervals

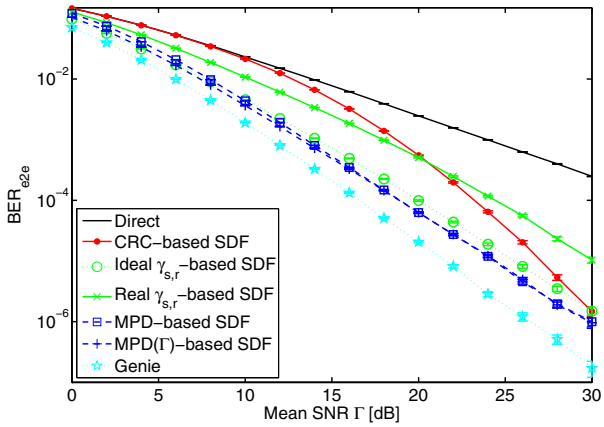

Fig. 10. End-to-end BER vs. mean SNR with direct transmission and different SDF metrics; $v = 20$ m/s, 10^5 transmitted messages per shown mean value, and 95% confidence intervals

velocity of $v = 1$ m/s. In the second *vehicular* scenario we assume $f_d = 350$ Hz corresponding to $v = 20$ m/s. Here, all channels vary faster and the state of each channel temporally decorrelates, i.e., the channels are "less stable" over time.

The BER_{e2e} results for the *indoor* scenario are shown in Fig. 9. Even with slowly fading channels, poor performance is obtained if SDF adapts only once per message. A further effect results from the quality of the decision metric. As SDF with realistic $\gamma_{s,r}$ bases its decision only on a few symbols per message, it achieves significantly lower performance than SDF using CRC as an accurate error estimator for a complete message. Comparing these results to the ideal case and practical, block-wise MPD clearly demonstrates that directly applying CRC or SNR-based SDF in WLANs is not the best design choice. Instead, partial forwarding with MPD significantly improves the performance. In this scenario, even a simple constant MPD threshold is sufficient as selecting MPD(Γ) thresholds only insignificantly improves BER_{e2e}.

Fig. 10 shows the BER_{e2e} results for the *vehicular* scenario. With this "faster" fading process the performance of the realistic $\gamma_{s,r}$ case has improved. Since here the channel state decorrelates over the complete message, an SNR sample measured over its preamble provides a more accurate estimate than with high time-correlation (Fig. 9). For medium and low SNR, CRC-based SDF suffers from CRC's conservativeness. In this fading scenario, where channel outages are shorter but more frequent, with CRC the *complete* message is dropped even if only a small part is affected. MPD, instead, forwards the correct blocks which decreases BER_{e2e} after combining at the destination. Consequently, MPD(Γ) reaches the best BER_{e2e} performance which is not significantly decreased by simply choosing a constant threshold.

VI. Conclusion

We studied SDF's end-to-end BER (BER_{e2e}) using different metrics for channel quality estimation. We conclude that, in practice, SDF's *message-wise* forwarding decision based on CRC or SNR [2, 5] cannot realize the full performance of cooperation. Replacing these metrics by our decoding-based metric MPD provides an accurate and more frequent channel quality estimation. Such a frequent estimation allows the relay to forward only the *parts* of a packet that are likely to be correct, resulting in superior cooperation performance.

By using MPD, SDF cooperation protocols closely reach the BER_{e2e} of the ideal case even with practical channel and system assumptions. Similar to SNR, an MPD-based forwarding decision relies on thresholds. Unlike SNR, however, MPD-based SDF pays only a marginal performance penalty even if suboptimal, constant thresholds are selected. Even then, MPD achieves its superior performance at no cost in terms of data rate and can be efficiently integrated into widely employed Viterbi decoders. Altogether, this makes MPD the first known practical approach to realize SDF with partial forwarding.

References

[1] A. Sendonaris, E. Erkip, and B. Aazhang, "Increasing uplink capacity via user cooperation diversity," in *Proc. IEEE Int. Symp. on Inf. Theory (ISIT)*, Aug. 1998, p. 156.

[2] J. N. Laneman, G. W. Wornell, and D. N. C. Tse, "Cooperative diversity in wireless networks: Efficient protocols and outage behavior," *IEEE Trans. Inf. Theory*, vol. 50, no. 12, pp. 3062–3080, Dec. 2004.

[3] J. G. Proakis, *Digital Communications*, 4th ed. McGraw-Hill, 2000.

[4] M. K. Simon and M.-S. Alouini, *Digital Communications over Fading Channels*, 2nd ed. John Wiley & Sons, Inc., 2004.

[5] P. Herhold, E. Zimmermann, and G. Fettweis, "A simple cooperative extension to wireless relaying," in *Proc. Int. Zurich Sem. on Commun.*, 2004, pp. 36–39.

[6] F. A. Onat, A. Adinoyi, Y. Fan, H. Yanikomeroglu, J. S. Thompson, and I. D. Marsland, "Thereshold selection for SNR-based selective digital relaying in cooperative wireless networks," *IEEE Trans. Wireless Commun.*, Mar. 2008, accepted for publication.

[7] A. Willig, "Intermediate checksums for improving goodput over error-prone links," in *Proc. Vehicular Technology Conf. (VTC-Fall)*, Sept. 2004.

[8] B. O'Hara and A. Petrick, *IEEE 802.11 Handbook: A designers companion*. IEEE Press, 1999.

[9] S. Valentin, T. Volkhausen, F. Atay Onat, H. Yanikomeroglu, and H. Karl, "Enabling partial forwarding by decoding-based one and two-stage selective cooperation," in *Proc. IEEE CoCoNET Workshop, Int. Conf. on Commun. (ICC)*, May 2008.

[10] E. C. Strinati, S. Simoens, and J. Boutros, "Error rate estimation based on soft output decoding and its apllication to turbo coding," in *Proc. IEEE Wireless Commun. and Netw. Conf. (WCNC)*, Mar. 2007.