Protecting the robustness of ADSL and VDSL DMT modems when applying DSM

Jan Verlinden, Etienne Van den Bogaert and Tom Bostoen
Research and Innovation
Alcatel Bell
B-2018 Antwerp, Belgium
Email: jan.vj.verlinden@alcatel.be

Raphael Cendrillon and Marc Moonen
Department of Electrical Engineering
Katholieke Universiteit Leuven
B-3001 Leuven, Belgium
Email: raphael.cendrillon@esat.kuleuven.ac.be

***Abstract*— When transmitting data over an ADSL or VDSL link, performance is very important. In order to improve the bit rate that can be achieved over the copper link, a lot of techniques like power backoff (PBO) and dynamic spectrum management (DSM) focus on the crosstalk and try to operate with lower noise margin.**

Today's ADSL and VDSL modems are very robust. Both in ADSL and VDSL there exist reconfiguration protocols that take care of changing noise environments. The intent of the paper is to know what the impact is of DSM on the robustness of these systems. If the noise increases, the modem may lose showtime, unless the modem can adapt its PSD to compensate for the increase of noise.

In this paper, we investigate for DSM the speed and robustness of various online reconfiguration protocols that exist today. We will consider a worst case noise : a noise that also impacts the communication channel that is needed to reconfigure the modem. Since reconfiguration is essential to recover from a degraded environment, the speed and the robustness of this reconfiguration protocol is very important.

I. Introduction

A. DMT modulation

Numerous books can be found with respect to DMT modulation (see e.g. [2, chapter 1]). A big advantage of DMT is the flexibility to configure its spectrum and bit loading in accordance with the loop and noise conditions of the line. Each carrier (DMT is often called MCM - Multi Carrier Modulation) is considered as an individual channel. It is characterized by the number of bits (b_i) it transports per tone and its relative gain setting (g_i) with respect to the average PSD. This means that on carriers with high SNR, a high number of bits can be loaded, whereas on carriers with low SNR, only a small number of bits are loaded. Because noise conditions on the line can change, there are protocols foreseen to reconfigure the bit loading and the gains on each of the carriers.

Currently, there is a tendency to reduce the noise margin to a minimum value, either to save power, to reduce crosstalk or to use this power on other carriers to increase the bit rate. The last two reasons to reduce the margin refer to DSM - Dynamic Spectrum Management (see [3], [9] and [10]). When operating with reduced noise margin, it is very important to restore the noise margin in case a new noise source becomes active. When operating the line with a negative noise margin, the bit error rate (BER) will be above the target value of 10^{-7}. If the BER remains high for a certain period of time, the modem will shut down the link and will re-initialize. For normal internet traffic a re-initialization is not dramatic : it will annoy the user, but chances are big that he won't even notice the interruption of service. Other services like voice, video and gaming require however a high Quality of Service (QoS) and re-initializations should be avoided as much as possible. This means that the speed to reconfigure the bit loading is very important.

II. Various Bit swap Protocols

A. ADSL[1]

The ADSL bit swap protocol uses the ADSL overhead control (AOC) channel. The exact details of the AOC protocol can be found in [4, Chapter 11]. All AOC messages are transmitted 5 consecutive times and they will only be processed if 3 identical messages in a time period spanning 5 of these particular messages are received. This means that the AOC protocol uses majority voting on message level.

Every message can contain only a limited number of bit swap commands. The normal bit swap message can have 4 commands, and an "extended" bit swap message can have 6 commands. Every b_i or g_i change requires a separate command. The length of a normal message is 9 bytes, the extended message consists of 13 bytes.

After the transmission of the bit swap request, the near-end modem will start a timer and wait for about 500 ms. When no acknowledgement has been detected in this time-out interval, the near end modem will re-send a bit swap request message. So each time a failure in the protocol occurs, the actual bit swap is postponed with approximately 500 ms.

If the far end modem received the bit swap request without errors, it can answer the request with an acknowledge message, which has to be sent within 400 ms after the reception of the request. This acknowledge message also contains a counter to indicate when the bit swap is to take place. The specified counter needs to take into account a minimum wait time of 800 ms, which means that the actual bit swap can be performed at the earliest 800 ms after the request.

Due to tone ordering the AOC channel is always allocated to the tones with the lowest number of bits. Since in ADSL no

[1] ADSL or ADSL1 to make the distinction with ADSL2.

1-bit constellations are used, the AOC channel will probably be transported on tones having 2-bit or 3-bit QAM constellations.

B. DMT VDSL

The DMT VDSL bit swap protocol uses the VDSL overhead control (VOC) channel. The exact details of the VOC protocol can be found in [6, Section 10.7]. The VOC protocol has all functionality of the AOC protocol. On top of this there exists an additional ”express” bit swap message. Since the AOC messages only support a few commands per message, this express bit swap mechanism was added to VDSL because there are much more tones in VDSL than in ADSL. With the express bit swap 4096 tones can be updated using a single message. In this case majority voting is not used : the message is only sent once, but the message contains a 16 bit CRC for error detection. There is no express swap acknowledge command, so the receiver that initiates the bit swap request is responsible for monitoring the returned signal to determine if the bit swap request has been implemented by the far-end transmitter or not. This requires of course additional complexity at the receiver. Since there is no acknowledge message to indicate when the bit swap is to take place, this is already indicated by some additional fields in the express bit swap message. It can be either the ”next bit swap frame” or the ”next-to-next bit swap frame”. Since a bit swap frame consists out of 16 DMT symbols, this means that the bit swap will be performed approximately (duration of a DMT symbol is not fixed in VDSL) after less than 4 ms or after less than 8 ms. If the near end modem receiver detects that the bit swap request was not implemented by the far end transmitter, it can choose to re-send the express bit swap message.

In VDSL the same tone ordering is applied as for ADSL, so the VOC channel is also allocated to the tones with the lowest number of bits. For VDSL these tones will have 1-bit, 2-bit or 3-bit QAM constellations.

C. ADSL2

In ADSL2 the bit swap messages are encapsulated in an HDLC based frame structure. Exact details of this protocol can be found in [5, 7.8]. The HDLC frame contains 16 bits that are used as CRC. The maximum allowed message size of the HDLC frame is 1024 bytes. In ADSL2 the operator can set the bit rate used for the overhead channel $C_{overhead}$ from 4 kbps to 64 kbps. A typical value is 8 kbps. After the transmission of a bit swap message, the transmitter will wait for 400 ms to get a response. This response will affirm the message, defer the message or reject the message. If no response was received, the transmitter will re-transmit the same message. Immediately after detection of a affirmative response, the bit swap is executed.

In ADSL2 a different tone ordering mechanism is used than for ADSL or VDSL. The overhead channel is not allocated explicitly to tones with a certain QAM constellation, so it could be allocated to any QAM constellation ranging from 1-bit to 15-bit QAM constellations.

III. Expected time needed for the complete execution of a single bit swap message

A. Geometric model

A geometric model can be used to model the exchange of messages and their probability of success (see [12]). It indicates the time (expressed as the number of trials) until the first success occurs.

$$\begin{aligned}
p_m &= \text{probability of successful transmission of a single message} \\
p_{em} &= 1 - p_m = \text{probability of error/failure for transmitting a single message} \\
P(x) &= p_{em}^{x-1} p_m = \text{probability that the first successful transmission occurs after x trials} \\
E[x] &= \frac{1}{p_m} = \text{expected number of trials until first success} \\
E[x] - 1 &= \frac{1}{p_m} - 1 = \text{expected number of unsuccessful trials}
\end{aligned}$$

B. ADSL

This results in the following formula for ADSL, indicating the average time needed for the complete execution (transmission, successful reception, implementation) of a single bit swap message :

$$T_{\mathrm{m}}(ADSL) = 800 + 500 \times (\frac{1}{p_m} - 1) \quad [ms] \qquad (1)$$

C. VDSL

For VDSL, one has the same formula as for ADSL if no express bit swaps are used. When only express bit swaps are used, the calculation is more complicated. Since the response time is much shorter, and the message size could be much longer than for regular bit swap messages, one also needs to take into account the delay for transmitting the message. Also the processing time needed to determine if the bit swap was executed or not is taken into account. Based on [8] it can be calculated that for a simple receiver 4 ms should be enough to guarantee reliable detection of the express bit swap.

The average time needed for the complete execution of a single bit swap message can then be expressed as follows :

$$T_{\mathrm{m}}(VDSL) = (T_{delay} + T_{message}) \times \frac{1}{p_m} \quad [ms] \qquad (2)$$

with T_{delay} being the sum of the time given to the transmitter to implement the bit swap (4 ms or 8 ms) and the time needed for the receiver to detect the bit swap (4 ms). $T_{message}$ is the time needed to communicate the actual bit swap message via the overhead channel.

The time needed to transmit the message $T_{message}$ depends on the number V of overhead bytes that is used per DMT symbol. A typical value is $V = 1$ byte. So the time needed for an express bit swap request can be expressed as [6, 10.7.3.9]:

$$L_{message}(VDSL) = 8 \times (5 + 2.5 \times N_{tones}) \quad [bits] \tag{3}$$

$$T_{message}(VDSL) = \frac{L_{message}}{8 \times V} \times 0.250 \quad [ms] \tag{4}$$

D. ADSL2

The size of a bit swap request (on-line reconfiguration command of type 1) can be expressed as [5, 9.4]:

$$L_{message}(ADSL2) = 8 \times (9 + 3 \times N_{tones}) \quad [bits] \tag{5}$$

so the time needed for a bit swap request in ADSL2 is :

$$T_{message}(ADSL2) = \frac{L_{message}}{C_{overhead}} \quad [ms] \tag{6}$$

If the bit swap command is accepted or rejected then immediately a message will be send back. So, the time out of 400 ms is only used when the response message would not be received correctly. The length of the response message is fixed to 9 bytes. As such, the average time needed for the complete execution of a single bit swap message is expressed as follows :

$$T_{\mathrm{m}}(ADSL2) = T_{message} \times \frac{1}{p_m} + 400 \times p_{em-re} \times (\frac{1}{p_m} - 1)[ms] \tag{7}$$

with p_{em-re} the chance to have an error in the response message.

IV. Error probability of a message : p_{em}

Trellis and Reed-Solomon coding are not taken into account for these calculations. It is assumed that coding can give additional protection against noise. However, these codes are used to increase the number of bits for data transmission and not to lower the BER. With coding the target BER is still 10^{-7} for a noise margin equal to zero. Since the effects of burst errors are not investigated in this paper, it means that we can work as if the physical layer is uncoded without affecting the general conclusions.

A. Symbol Error Probability for QAM Modulation

ADSL and DMT VDSL use QAM modulation with constellation sizes ranging from $k = 1$ to $k = 15$. For $M = 2^k$, the probability of a symbol error (symbol error rate) for k even is given by [2, section 6.3.4]

$$p_{es}(M) \leq 4\left(1 - \frac{3}{2M}\right) \operatorname{erfc}\left(\sqrt{\frac{6\gamma_{\text{av}}}{2M-1}}\right) \tag{8}$$

and for k odd by

$$p_{es}(M) \leq 4\left(1 - \frac{1}{\sqrt{M}}\right) \operatorname{erfc}\left(\sqrt{\frac{3\gamma_{\text{av}}}{M-1}}\right) \tag{9}$$

where γ_{av} is the average SNR per symbol.

B. Bit Error Probability

For ADSL Gray coding is used to minimize the amount of bit errors when a symbol error occurs. This means that the bit error probability can be approximated with ([1, 4.2.6]) :

$$p_{eb} = \frac{1}{k} p_{es}(M) = \frac{1}{k} p_{es}(2^k) \tag{10}$$

For VDSL no Gray coding is used and one can assume that in the event of a symbol error, all erroneous bit combinations are equally likely to occur. This assumption yields following formula ([1, 4.2.2]) :

$$p_{eb} = \frac{2^{k-1}}{2^k - 1} p_{es}(2^k) \tag{11}$$

C. Protection of the bit swap protocol

Protection of the bit swap protocol was introduced to have a higher protection against errors. The reason is very clear. When the noise on a tone increases, its margin will decrease, so the probability to have an erroneous bit on this carrier will also increase. At the moment the margin drops below a certain level, a "bit swap" is needed in order to change the b_i and/or g_i settings of this carrier and to restore the margin. However, during this period, the carrier has a low margin and a higher probability to produce errors, than during normal operations. So if the bit swap message is transmitted on carriers that are also affected by this noise, then this mechanism (which should restore the margin) is extra vulnerable to errors. That is why the overhead protocols that take care of the bit swap mechanism, need extra protection.

Thanks to this extra protection, the probability to have an error in the bit swap protocol is reduced. This is very important, because in case a wrong bit swap command is executed by mistake, the complete demapping will be wrong and the communication link is lost, resulting in a re-initialization.

A calculation of the reliability (what is the chance that an error occurs in the overhead channel, but is not detected) can be found in [8]. In contrast with [8], this paper focuses on detected errors and the delay that is caused by such an error since they require re-transmission. The chance of an undetected error is very small and considered here to be negligible.

For ADSL and VDSL the protection of the bit swap protocol is done either by means of majority voting or cyclic redundancy check (CRC).

1) Majority voting: For the AOC (ADSL overhead control) and VOC (VDSL overhead control) channel, majority voting is used [4, section 11.1.2] and [6, section 10.7.2]: all AOC (VOC) messages are transmitted 5 times and the receiver will accept the message only if within these 5 messages, 3 identical messages are detected and it will use that one. If no 3 identical messages can be detected, no action shall be taken.

The chance to reject the message (having at least 3 errored submessages) for ADSL and VDSL using normal bit swapping is given by :

$$p_{em} = p_{esubm}^5 + 5p_{esubm}^4(1 - p_{esubm}) + 10p_{esubm}^3(1 - p_{esubm})^2 \quad (12)$$

with p_{esubm} the chance to have an errored submessage :

$$p_{esubm} = 1 - (1 - p_{eb})^L \quad (13)$$

with L the length of submessage : 72 (9 bytes) for a normal message or 104 (13 bytes) for an extended message.

2) Cyclic Redundancy Check: Both ADSL2 and VDSL express bit swap use a cyclic redundancy check to detect errors in the messages. In both cases, each message contains a 16-bit CRC protection.

The chance for ADSL2 and VDSL using express bit swaps to reject the message (having at least 1 error) is given by :

$$p_{em} = 1 - (1 - p_{eb})^{L_{message}} \quad (14)$$

with $L_{message}$ the size of the message.

D. Explanation of the expression "single bit swap message"

The "single bit swap message" is defined as follows : for ADSL it is a normal bit swap message. This message contains 4 commands, allowing a change of b_i and/or g_i, with a maximum increase of 3 dB and a maximum decrease of 2 dB for the g_i values per command. For ADSL2 it is an online reconfiguration message with reconfiguration parameters (exact value for the new b_i and g_i) for 200 tones (typical amount of tones for downstream). For VDSL express swapping the message contains configuration parameters for 1000 tones (typical amount of tones used in upstream or downstream) : the b_i can be set to any value, the g_i can be modified with a value ranging from -4 dB to 3 dB indicating the amount of decrease or increase.

Suppose for example that a modem is deployed on a line at a rather low bit rate using DSM. It will do power back off in order to reduce the possible crosstalk towards other users and the noise margin will be 6 dB. Suppose that due to some disturbance the noise on all tones would be increased with 4 dB. For ADSL this requires $(200 \times 2)/4 = 100$ messages, as the maximum increase in g_i is limited to 3 dB. For ADSL2 this requires only 1 single message, updating all tones at once. For VDSL it would require 2 messages, when using express bit swapping ; using normal bit swapping it requires $(1000 \times 2)/4 = 500$ messages.

V. Simulation Results

A. Quantification of crosstalk noise

Based on [7] simulations can be done to quantify the effect of crosstalk. Two reference cases have been simulated, one for ADSL on a loop of 4000 m, one for VDSL on a loop of 700 m. In each simulation the SNR is calculated for three different scenarios : (a) no crosstalkers, so only with AWGN at -140dBm/Hz, (b) 1 worst case self crosstalker combined with AWGN at -140dBm/Hz, and (c) 2 worst case self crosstalkers combined with AWGN at -140dBm/Hz. As such, one can see the degradation in SNR in case a first or a second self crosstalker comes up.

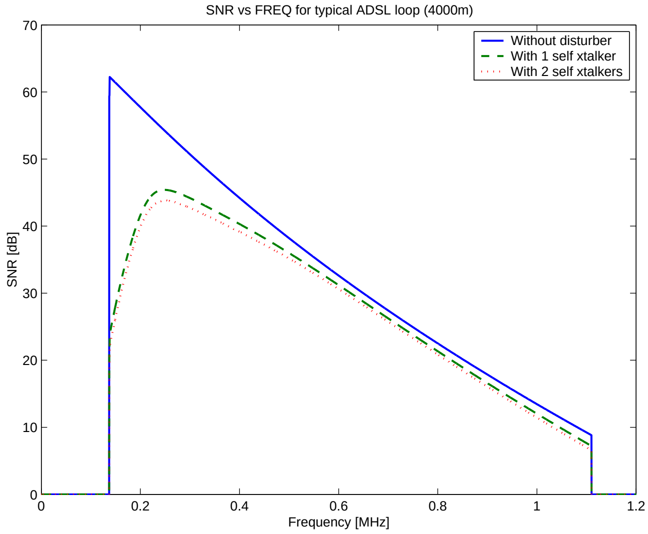

Fig. 1. SNR vs frequency for ADSL with various disturbers on 4000m loop

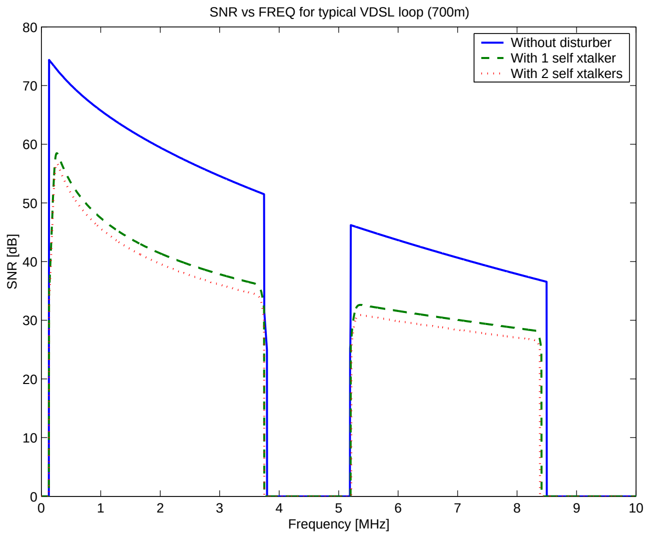

Fig. 2. SNR vs frequency for VDSL with various disturbers on 700m loop

These two cases are examples in order to quantify the degradation in SNR. Other loops will give different results, but it is mainly the order of magnitude that is important for the rest of this paper. One can see in Fig. 1 that for ADSL a first disturber will degrade the SNR on tones lower than 250 kHz with more than 9 dB, on tones higher than 250 kHz with less than 9 dB. For VDSL (see Fig. 2) a first disturber will degrade the SNR on all tones with more than 9 dB on all tones. A second disturber will degrade the SNR for both ADSL and VDSL with less than 1.8 dB. This can be calculated from [7, A.3.2] : $n^{0.6}|_{n=2} = 1.5 = 1.8dB$.

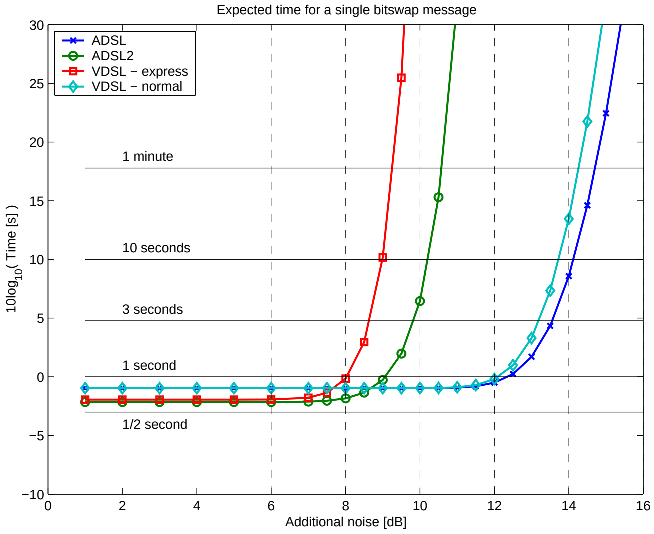

Fig. 3. Expected time for a single bit swap message

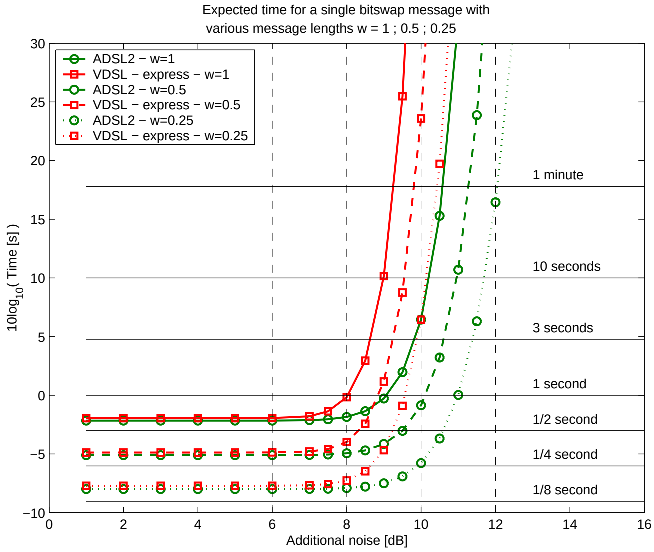

Fig. 4. Expected time for a single bit swap message for various message lengths

B. Estimated time for a single bit swap message

Before the noise of a crosstalker is added it is assumed that the modem is working with an initial noise margin of 6 dB. This means that if the noise would increase with 6 dB, then the resulting noise margin would be 0 dB and the modem would still operate with a BER of 10^{-7}.

Since T_m also depends on the constellation size k of the carrier(s) that are used for the overhead communication, tone ordering applied in the various xDSL flavours and its effect on the overhead channel is taken into account (see [4] ; [5] and [6]). The tone ordering determines which tones are used for the overhead channel. So for ADSL1 T_m is calculated as the average $T_m(k)$ for $k = 2, 3$. For ADSL2 $k = 1, 2, ...15$ is used and for VDSL $k = 1, 2, 3$ is used.

In Fig. 3 the estimated time for successful transmission of a single bit swap message (T_m), depending on the amount of additional noise is plotted. The same amount of additional noise is added on all tones. As one could expect, the addition of 6 dB of noise or less has no impact on T_m. For higher noise injections however, there seems to be some kind of threshold. At this point the communication link is degraded that much, that it is almost impossible to exchange bit swap messages : T_m is increasing very fast towards infinity. This means that for such a noise the modems are not even able to exchange messages to improve the quality of the communication channel and the modems will have to re-initialize.

It is very obvious that ADSL2 and VDSL express bit swapping require less time to perform the complete update. However, due to their large message size, they are more sensitive to bit errors on the overhead channel.

C. Effect of message length on T_m

In Fig. 4 the effect of the message length on the expected time for a single bit swap T_m is shown, in case of ADSL2 and VDSL using express bit swap (ADSL and VDSL using normal bit swap use fixed message length). The parameter w indicates the relative size of the message, compared to the original size (e.g. $w = 0.5$ means that for ADSL only half of the tones, so 100 tones are reconfigured in a single bit swap message). For low noises it turns out that the time needed for successful bit swap message is mainly determined by the length of the message, so when this length is divided by 2, also the required time is divided by 2 (3 dBt difference, with $[dBt] \equiv 10log_{10}([s])$). In fact this means that the total time remains more or less the same. This statement is valid as long as the time needed to transmit the message is still much larger than the time for the other components like acknowledgement and processing. The robustness however improves a lot when smaller messages are transmitted. For ADSL2, the protocol is capable of successful recovering from a situation with approximately 0.8 dB more noise than before, for each factor 2 of message size reduction. For VDSL this is approximately 0.6 dB more noise than before, for each factor 2 of message size reduction.

D. Effect of the speed of the overhead channel on T_m

As can be seen in Fig. 5 increasing the speed of the overhead channel reduces the speed of the time needed for a successful transmission. The robustness against additional noise however, is only slightly improved, mainly because the curve of T_m vs additional noise is very steep for higher values of additional noise.

E. Improvements

Robustness can be significantly improved by sending smaller messages. However, when sending smaller messages, also the number of reconfigurations per message is limited.

It would in fact be better if one could send a single message indicating that the g_i should be increased by x dB for a range of tones, by just indicating the first tone of the range, the last tone of the range and x, the amount of increase for g_i.

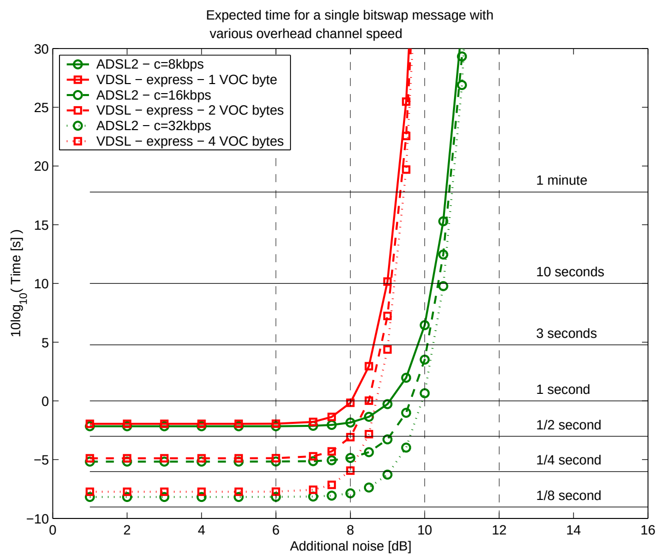

Fig. 5. Expected time for a single bit swap message for various speeds of the overhead channel

This would be a very short message, so the robustness for successful communication is very high. After transmission of such a message the individual tones can be tuned to their final value, but at least the communication link survived.

Such a bit swap command would require changes of the current standards.

VI. Conclusion

In this paper we examined the probability of error for the various bit swap protocols in case they are used in a modem using DSM. Based on crosstalk simulation we indicated the amount of additional noise that is injected to a modem if a new connection is established on a neighbouring modem. The average time needed for successful transmission of a single bit swap message was plotted versus the amount of additional noise injected into the modem.

Limitations of the various protocols in the scope of DSM are indicated : ADSL and VDSL normal bit swapping are the most robust ones, but rather slow. ADSL2 and VDSL express bit swapping are very fast but are less robust when the modem needs to deal with sudden high noise increases.

DSM can improve the reach and the rate of current modems, but we don't want to operate with less robustness compared to the current ADSL and VDSL modems who usually work with higher noise margins. When using DSM, the message size for ADSL2 and VDSL should be reduced, as it can significantly improve robustness. A short message dealing with a g_i increase on a range of tones, for which only start and stop tone need to be indicated, is even better as it can update a lot of tones in a robust way.

References

[1] J. G. Proakis, *Digital Communications*, 2nd ed. McGraw-Hill, 1989.
[2] T. Starr, J. M. Cioffi, P. J. Silverman, *Understanding digital subscriber line technology*, Prentice Hall , 1999.
[3] W. Yu, G. Ginis, J. M. Cioffi *Distributed power control for digital subscriber lines*, IEEE J. Select. Areas Commun. vol 20, no. 5, pp. 1105-1115, June 2002.
[4] ITU-T, *G.992.1 - Asymmetric Digital Subscriber Line (ADSL) transceivers*, ITU-T, 1999.
[5] ITU-T, *Draft G.992.3 AAP - ADSL2, Asymmetric Digital Subscriber Line*, ITU-T, 2002.
[6] Committee T1 - Telecommunications, *T1E1.4 VDSL - Very-high bit-rate Digital Subscriber Line (VDSL) Metallic Interface, Part 3 : Technical Specification of a Multi-Carrier Modulation Transceiver*, Committee T1 - Telecommunications, 2001.
[7] Committee T1 - Telecommunications, *T1E1.4/2003-002R3 Draft proposed American National Standard, Spectrum Management for Loop Transmission Systems, Issue 2*, Committee T1 - Telecommunications, 2001.
[8] L. Hoo, A. Salvekar, C. Aldena, J. M. Cioffi, *T1E1.4/99-119 Analysis of Express Swapping Speeds and Reliability*, Committee T1 - Telecommunications, 1999.
[9] R. Cendrillon, W. Yu, J. Verlinden, T. Bostoen, M. Moonen, *Optimal Multi-user Spectrum Management for Digital Subscriber Lines*, Submitted to IEEE Trans. Commun., 2003.
[10] T. Bostoen, *White Paper : Vision on the future of DSL : Dynamic Spectrum Management and Remote Terminal deployment*, Alcatel, 2003.
[11] S. Lacaille, *Reed Solomon Coding and Interleaving for Forward Error Correction in ADSL*, Université de Valenciennes et du Hainaut-Cambrésis (France), 1996.
[12] R. L. Scheaffer, *Introduction to Probability and its Applications*, 2nd ed. Duxbury Press, 1995.