Challenges and Solutions in Vectored DSL

Raphael Cendrillon, Fang Liming, James Chou, Guozhu Long, and Dong Wei

Huawei Technologies Co., Ltd.
Bantian Longgang District, Shenzhen 518129, P.R. China
`Raphael_2008@huawei.com`

Abstract. VDSL2 is the latest generation of DSL technology, and aims to provide data-rates in excess of 100 Mbps to the home to enable next generation, high bandwidth services. Such high data-rates are achieved by transmitting at frequencies up to 30 MHz. Unfortunately transmitting at such high frequencies over twisted-pair leads to strong crosstalk between the lines. To address this issue crosstalk cancellation and precoding were developed, techniques known collectively as vectoring. The ITU is now finalizing the first version of the G.vector standard which specifies how vectoring should be implemented. This paper presents some of the more interesting problems that have been encountered in the development of this standard, and shows how creative solutions, which take advantage of the unique characteristics of the DSL environment, have helped develop vectoring from theory into a practical technology that will soon be ready for commercial deployment.

Keywords: DSL, Crosstalk Cancellation, Vectoring, DSM.

1 Introduction

VDSL is the latest generation of DSL technology, and aims to provide data-rates in excess of 100 Mbps to the home to enable next generation, high bandwidth services. Such high data-rates are achieved by transmitting at frequencies up to 30 MHz. Unfortunately transmitting at such high frequencies over twisted-pair, which was originally designed for voice-band transmission only, leads to strong crosstalk between the lines within a cable binder, an effect known as crosstalk.

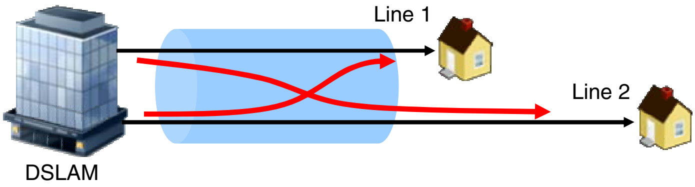

Fig. 1. The crosstalk problem

X. Jun Hei and L. Cheung (Eds.): AccessNets 2009, LNICST 37, pp. 192–203, 2010.

Crosstalk is typically 20 dB larger than any other noise source in the system and leads to large reductions in data-rate and service stability as shown in Figure 2.

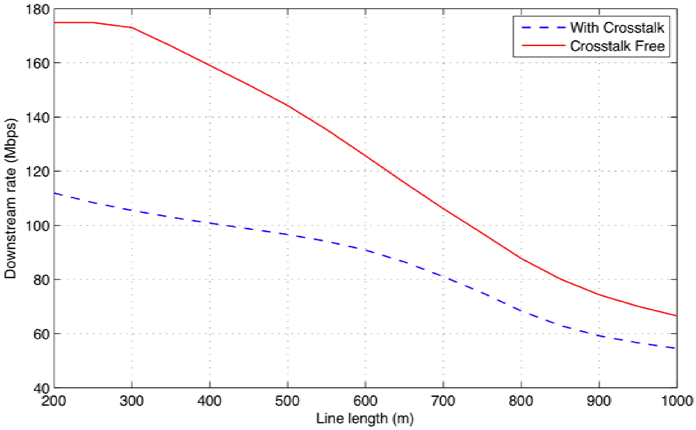

Fig. 2. Crosstalk leads to significant loss of data-rate

Crosstalk transforms the convention single-user wireline communications channel into a multi-user channel. Because of this the crosstalk problem can be solved by applying multi-user techniques like multi-user detection, power optimization and interference precoding[1][2][3][4]. These techniques exploit the fact that the central office modems from different lines are located in a common DSLAM. Since upstream receivers are co-located (a multi-access channel from the information theory perspective) joint reception is possible allowing crosstalk cancellation to be applied. In the downstream direction transmitters are co-located (a broadcast channel from the information theory perspective) allowing crosstalk precoding to be applied.

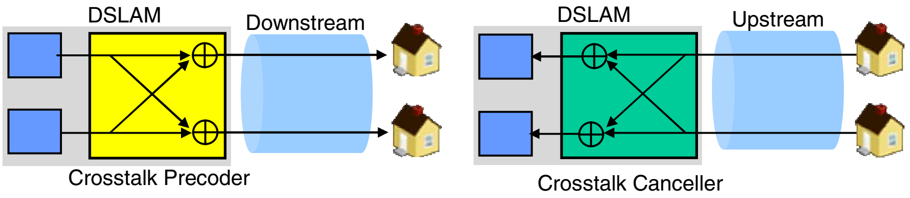

Fig. 3. Crosstalk cancellation and precoding architectures

Crosstalk cancellation and precoding are known collectively as vectoring in the DSL community. Significant work is now being done by industry on vectoring, with much of this work focused on the development of the ITU G.vector standard[5]. The goal of this standard is define the features, capabilities and protocols required to implement vectoring, and to ensure that equipment from different vendors is interoperable.

This paper presents some of the challenges faced in the development of G.vector, with specific focus on crosstalk precoder training. It is shown how creative solutions, which take advantage of the unique characteristics of the DSL environment, have helped develop vectoring from theory into a practical technology that will soon be ready for commercial deployment.

2 Crosstalk Precoding

In the downstream direction lines are synchronized allowing transmission to be modeled independently on each tone

$$\underbrace{\begin{bmatrix} y_k^1 \\ \vdots \\ y_k^N \end{bmatrix}}_{\mathbf{y}_k} = \underbrace{\begin{bmatrix} h_k^{1,1} & \cdots & h_k^{1,N} \\ \vdots & \ddots & \vdots \\ h_k^{N,1} & \cdots & h_k^{N,N} \end{bmatrix}}_{\mathbf{H}_k} \underbrace{\begin{bmatrix} x_k^1 \\ \vdots \\ x_k^N \end{bmatrix}}_{\mathbf{x}_k} + \underbrace{\begin{bmatrix} z_k^1 \\ \vdots \\ z_k^N \end{bmatrix}}_{\mathbf{z}_k} \qquad (1)$$

Here u_k^n denotes the signal transmitted on tone k of line n, y_k^n denotes the signal received on tone k of line n, z_k^n denotes the noise on tone k of line n, and $h_k^{n,m}$ denotes the crosstalk channel from line m into line n on tone k.

In a vectored system the downstream signals from the different lines on tone k are precoded with a precoding matrix $\mathbf{P}_k$ prior to transmission. This precoding introduces some distortion into the transmitted signals on each line, where the distortion is selected such that it cancels with the crosstalk introduced in the channel. Denote the signal intended for line n on tone k as x_k^n. The output of the crosstalk precoder $\mathbf{u}_k$ depends on $\mathbf{x}_k$ and the crosstalk precoding matrix $\mathbf{P}_k$

$$\mathbf{u}_k = \mathbf{P}_k \mathbf{x}_k \ . \qquad (2)$$

The received signal is then

$$\mathbf{y}_k = \mathbf{H}_k \mathbf{P}_k \mathbf{x}_k + \mathbf{z}_k \ . \qquad (3)$$

Under a zero-forcing design, the precoding matrix $\mathbf{P}_k$ is chosen such that $\mathbf{H}_k\mathbf{P}_k$ is diagonal. Hence each receiver sees only its intended signal, completely free from crosstalk.

In a vectored system the downstream transmitters are co-located in the DSLAM as shown in Figure 1. Note that the crosstalk from line 1 into line 2 must propagate the full length of line 2's signal path. Similarly the crosstalk from line 2 into line 1 must propagate the full length of line 1's signal path. As a result in the downstream the crosstalk channels have a much lower magnitude than the direct channel of the corresponding victim.

$$h_k^{n,n} >> h_k^{n,m} \ . \qquad (4)$$

In fact empirical models have been developed for the worst 1% case crosstalk channels and show that with co-located transmitters, the downstream crosstalk channels can be bounded as

$$\left|h_k^{n,m}\right| \le \alpha_k \left|h_k^{n,n}\right| , \tag{5}$$

where $\alpha_k = \sqrt{8\times 10^{-20}\cdot (49)^{-0.6}\cdot 3.28\cdot f_k^2 \cdot \min(l_n, l_m)}$, f_k is the frequency of sub-carrier *k*, and l_n is the length of line *n* in meters[1]. This property of the crosstalk channel causes the diagonal element of the crosstalk channel matrix to always have a larger magnitude than the other elements on its row, a characteristic referred to as row-wise diagonal dominance (RWDD). RWDD allows a simple linear crosstalk precoder, known as the diagonalizing precoder, to achieve near-optimal performance[2]. With the diagonalizing precoder

$$\mathbf{P}_k = \mathbf{H}_k^{-1}\,\mathrm{diag}\{\mathbf{H}_k\} , \tag{6}$$

where diag{$\mathbf{H}_k$} is the matrix $\mathbf{H}_k$ with the off-diagonal elements set to zero. Using (3) the received signal is then

$$\mathbf{y}_k = \mathrm{diag}\{\mathbf{H}_k\}\mathbf{x}_k + \mathbf{z}_k . \tag{7}$$

So $y_k^n = h_k^{n,n} x_k^n + z_k^n$ and each user receives their signal completely free from crosstalk. Due to the RWDD of $\mathbf{H}_k$ it can be guaranteed that application of the diagonalizing precoder causes negligible increase in the transmit power, so the spectral masks constraints and analog front end limitations will not be violated[2].

3 Precoder Training

In order to apply crosstalk precoding the coefficients $\mathbf{P}_k$ must be set to their appropriate values. This could be done by first identifying the crosstalk channel and then using a closed form design, such as the diagonalizing precoder (6). However in practice an adaptive training algorithm is preferable since it has lower complexity and allows the crosstalk precoder to track variations in the crosstalk environment as the temperature changes and lines activate and de-activate.

In the draft G.vector standard precoder training is implemented by the VDSL2 Transceiver Unit at the Operator side (VTU-O) transmitting a pilot sequence during its sync symbol. The sync symbol is a special symbol transmitted once every 257 discrete multi-tone (DMT) symbols and is traditionally used to carry a control flag between the VTU-O and VDSL2 Transceiver Unit at the Remote side (VTU-R[1]) to indicate when certain events take place, for example when a change in the bitloading table or data-rate of a line is applied.

When an initializing line first activates it transmits only during the sync symbol time slots to avoid disrupting the vectored lines in the binder that are already active and carrying data. This is shown in Figure 4. Note that the 257 symbol super-frames of all lines are synchronized, so the initializing line only creates crosstalk during the

---

[1] Note that ‘VTU-R’ is simply VDSL terminology for customer premises equipment (CPE).

sync symbols of the active vectored lines. The initializing line is silent during the data symbol time slots of the active vectored lines.

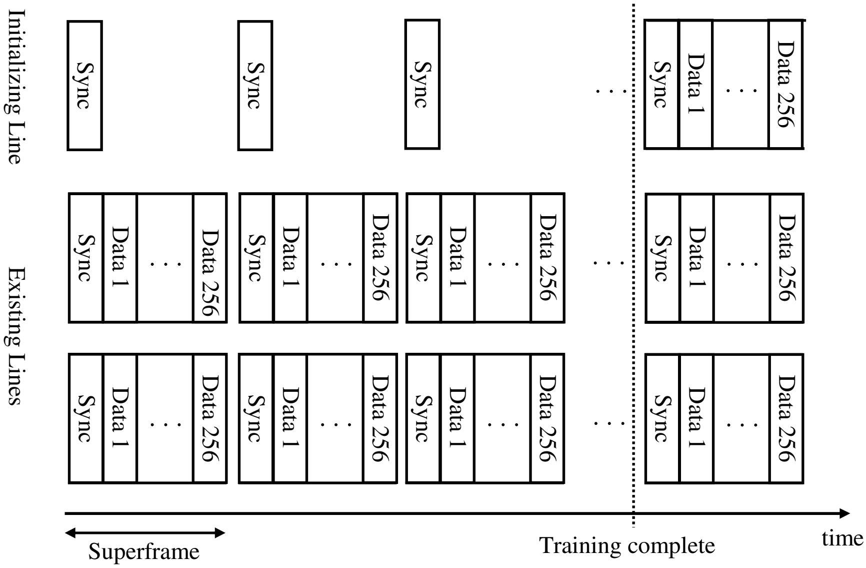

Fig. 4. Initializing lines transmit only during sync symbols to avoid disrupting active lines

The initializing line continues to transmit only during the sync symbol time slots until training of the crosstalk precoder is complete. Once the precoder coefficients for the initializing line have been determined it's crosstalk can be cancelled. At this point the initializing line may begin transmitting data during the data symbol time slots. Denote the symbol transmitted by line *n* on tone *k* during sync symbol time slot *t* as $d_k^n(t)$. The pilot sequence of line *n* on tone *k* can then be written as

$$\mathbf{d}_k^n = [d_k^n(1) \ldots d_k^n(T)] \; , \tag{8}$$

where T denotes the periodicity of the pilot sequence and is chosen to be greater than the number of lines in the binder. In order the identify the crosstalk channels the pilot sequences on different lines are chosen to be orthogonal across time.

$$\mathbf{d}_k^n \mathbf{d}_k^{m^H} = 0, \; \forall n \neq m \; . \tag{9}$$

Different adaptive algorithms could be applied in order to train the crosstalk precoder. This paper will present an approach based on LMS for its simplicity and for illustration, although any other adaptive algorithm could also be used. With LMS the error at the output of the frequency-domain equalizer (FEQ) is first calculated by the VTU-R.

$$e_k^n(t) = d_k^n(t) - w_k^n y_k^n(t) \; . \tag{10}$$

Here w_k^n is the FEQ coefficient for line n on tone k and is ideally set to $(h_k^{n,n})^{-1}$. The pilot sequence for line n on tone k is $d_k^n(t)$ and is communicated between the vectoring control entity (VCE) and the VTU-R before training begins. The VCE is a special entity that is responsible for managing the initialization and operation of the crosstalk precoder.

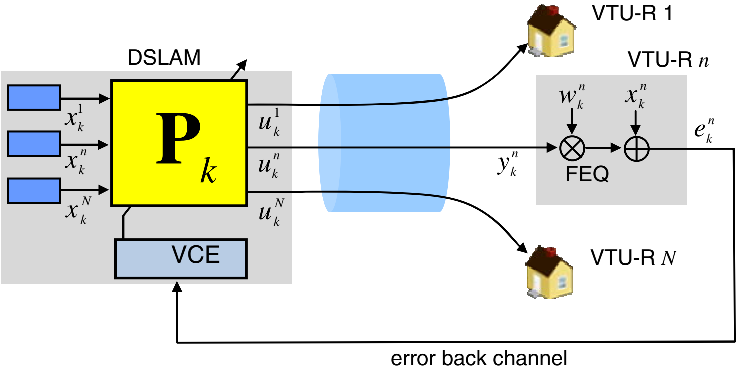

Fig. 5. Crosstalk precoder training

After the error is calculated by the VTU-R, it is encoded and sent back to the VCE through an upstream channel. The VCE then uses the error samples to update the crosstalk precoder coefficients. For example using LMS the update rule would be

$$p_k^{n,m}(t+1) = p_k^{n,m}(t) + \mu d_k^m(t)^* e_k^n(t) \ , \tag{11}$$

where $p_k^{n,m}$ is the element on the nth row and mth column of the precoding matrix $\mathbf{P}_k$.

4 Challenges and Solutions

4.1 Challenge: Limited Feedback Bandwidth

Although the method for crosstalk precoder training is simple in theory it does present a number of practical challenges. To begin with, when precoder training takes place the upstream channel has not been fully initialized, and so only a very low data-rate special operations channel (SOC) is available to feed back the error samples. In its standard configuration the SOC supports only 16 bits of feedback per DMT symbol. Error samples are measured during sync symbols, which occur once every 257 DMT symbols. So per sync symbol only 257x16 or approximately 4000 bits are available for error feedback (Note that error feedback is sent continuously over the SOC in between sync symbols).

In VDSL2 there are up to 2884 downstream sub-carriers. Each sub-carrier will have its own error sample, which means that in this configuration only 1.4 bits are available to feed back each complex error sample. This corresponds to only 0.7 bits per error sample dimension.

Using such a low number of bits leads to an extremely high quantization error. This high quantization error will in turn lead to a high asymptotic error for the LMS algorithm and low data-rates at convergence. Alternatively the step size for LMS can be reduced leading to a higher data-rate, but the training time will be intolerably long as a result. Neither of these options are acceptable in practice since a subscriber must have access to high data-rates within a matter of seconds after a VDSL2 modem is switched on.

4.2 Solution: Error Subsampling

One way to reduce feedback overhead is by subsampling the error samples in frequency. Figure 6 shows a typical set of crosstalk channels taken from measurements done in the Huawei laboratory. As can be seen despite the large variations the crosstalk channels vary quite smoothly with frequency. This means that the crosstalk precoder can be accurately initialized by feeding back the error on every *n*th sub-carrier, running LMS on the subsampled channel, and then using interpolation to determine the crosstalk precoder coefficients for the intermediate tones[6].

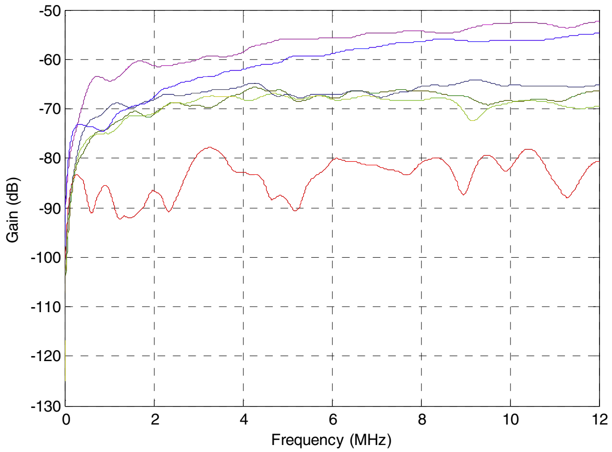

Fig. 6. Crosstalk channels vary quite smoothly with frequency

To evaluate the performance of error subsampling simulations were run in a binder consisting of 32 300m VDSL2 lines running profile 12a from Annex A. The background noise was set to -140 dBm/Hz, the coding gain was assumed to be 5 dB, and real crosstalk channel measurements were used. In all cases perfect feedback was assumed (no quantization) so that the impact of subsampling alone could be investigated.

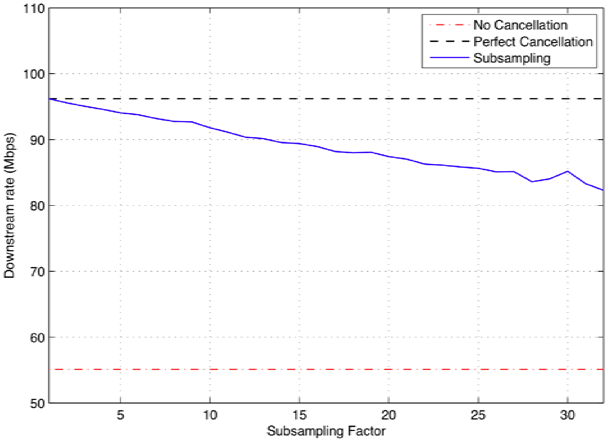

Fig. 7. Impact of error subsampling on data-rate

Figure 7 shows the average downstream data-rate as a function of the subsampling factor. As can be seen a subsampling factor of up to 8 can be used with only a minor reduction in performance.

Using a subsampling factor of 8 reduces the number of sub-carriers for feedback from 2884 to 360. This allows the number of feedback bits per error dimension to be increased from 0.7 to 5, significantly reducing the quantization error. Unfortunately though 5 bits is still not sufficient to achieve a high data-rate within a reasonable training time.

4.3 Challenge: Slow Convergence Speed

By limiting transmission of the pilot sequence to the sync symbol time slots only it is possible to ensure that initializing lines do not disrupt active vectored lines during the initialization process. The negative side to this is that since error samples are measured only during the sync symbols, which occur once every 257 DMT symbols, convergence can be extremely slow. As a result unless the LMS step size is set very high it is not possible to converge within an acceptable period of time (less than 10 seconds). Setting a high LMS step size is only possible if the quantization error of the error samples is very low.

Simulations were run to investigate the impact of the number of quantization bits per error sample dimension on the data-rate achieved at convergence of the LMS algorithm. The scenario consisted of 32 300m VDSL2 lines running profile 17a from Annex A. The background noise was set to -140 dBm/Hz, the coding gain was assumed to be 5 dB, and the beta crosstalk channel model was used[7].

Figure 8 shows the change in the achievable data-rate as the LMS algorithm converges. As can be seen converging to a high data-rate within 10 seconds requires at least 12 bits of feedback per error sample dimension. Unfortunately however only 5 bits are available with the conventional SOC, even if error subsampling is used.

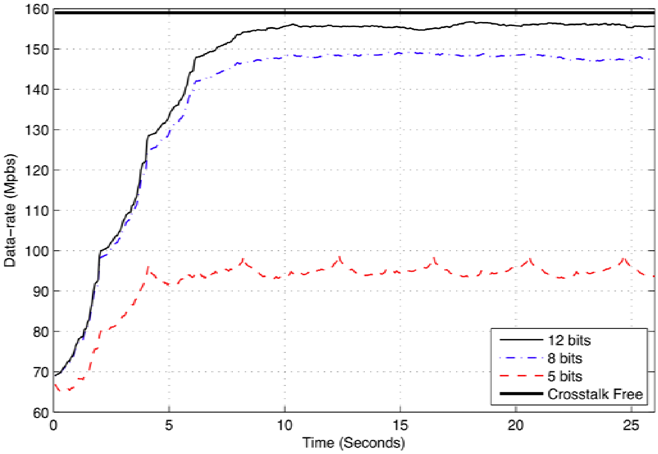

Fig. 8. Impact of number of quantization bits on data-rate at convergence

4.4 Solution: Error Scaling

The problem of high quantization error and slow training convergence can be solved by scaling the error samples prior to quantization[8]. Denote the maximum magnitude of the error samples across frequency for a particular sync symbol t and line n as

$$e_n^{\max}(t) = \max_k \left| e_k^n(t) \right| \ . \tag{12}$$

Figure 9 shows the decay of the maximum error magnitude $e_n^{\max}(t)$ as the LMS algorithm converges in a typical scenario.

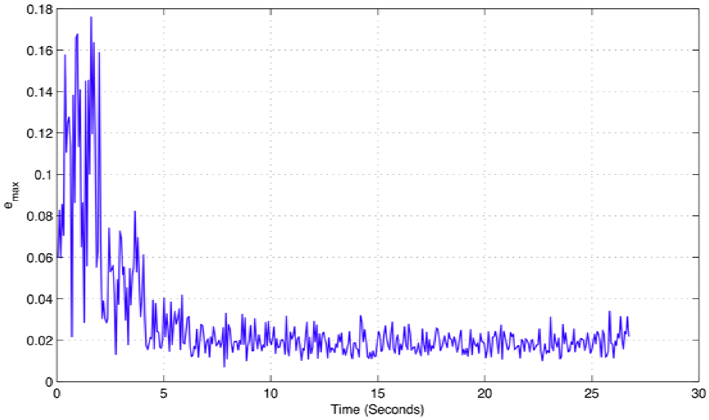

Fig. 9. The maximum error magnitude decays rapidly as LMS converges

As can be seen the error magnitude decays rapidly. This means that the quantization range can be reduced as the algorithm converges, ensuring a low quantization error even when very few feedback bits are used. In practice this is implemented with error scaling as follows. First the VTU-R determines the maximum error magnitude for the current sync symbol $e_n^{\max}(t)$. The scaling factor is then set as

$$s_n(t) = 2^{\left\lceil \log_2 (e_n^{\max}(t)) \right\rceil}, \tag{13}$$

where $\lceil . \rceil$ denotes the ceiling operation. The error samples on every tone are then normalized by the scaling factor $s_n(t)$.

$$\bar{e}_k^n(t) = e_k^n(t) / s_n(t) \ . \tag{14}$$

These scaled error samples are then quantized using the available number of bits (5 in our example) using a quantization range [-1,1]. Both the error scaling factor and the scaled, quantized error samples are then fed back to the VCE.

Using the scaling factor ensures that the scaled error samples are always in the range [-1,1] so that clipping of the error sample during quantization will never occur. In addition to this as the LMS algorithm converges the maximum error magnitude decays rapidly leading to a small value for the error scaling factor $s_n(t)$. This reduces the quantization range considerably, leading to a much lower quantization error. For example, if 5 bits are used per error sample dimension, then the maximum quantization error without error scaling is

$$e_{\text{quant}} \leq 2^{-5} \ . \tag{15}$$

This is large enough to reduce the convergence speed to an unacceptable level. With error scaling the error magnitude will rapidly fall to a level below 2^{-7}. This will result in an error scaling factor of 2^{-7}. The 5 available quantization bits will now span a much smaller quantization range of $-2^{-7} \ldots 2^{-7}$. This reduces the quantization error to a maximum of

$$e_{\text{quant}} \leq 2^{-12} \ . \tag{16}$$

Such a low quantization error leads to rapid convergence of the LMS algorithm, so the modem initialization time can be kept short without sacrificing data-rate. Since a common scaling factor is used for all sub-carriers it adds very little to the feedback overhead. For example, if 5 bits per error sample dimension are used, then feeding back the error samples for a single sync symbol with 360 tones requires 3600 bits. Feeding back the error scaling factor adds only 8 bits to the overall feedback requirement, or 3608 bits in total. This is an increase of only 0.2% and is negligible.

Note that the scaling factor is selected to be a power of 2 so error scaling can be implemented with a simple binary shift operation. This keeps complexity very low which is particularly important for the VTU-R (CPE).

Simulations were run to evaluate the performance of error scaling and its impact on the convergence of the LMS algorithm. The scenario consisted of 32 300m VDSL2 lines running profile 17a from Annex A. The background noise was set to -140 dBm/Hz, the coding gain was assumed to be 5 dB, and the beta crosstalk channel model was used. Performance with and without error scaling was evaluated. In both cases 5 bits of feedback per error sample dimension was used.

Figure 10 shows the change in the achievable data-rate as the LMS algorithm converges. As can be seen when error scaling is not used the high quantization error leads to a large loss in data-rate. With error scaling the data-rate is improved significantly, allowing high performance to be achieved with a convergence time of only 10 seconds.

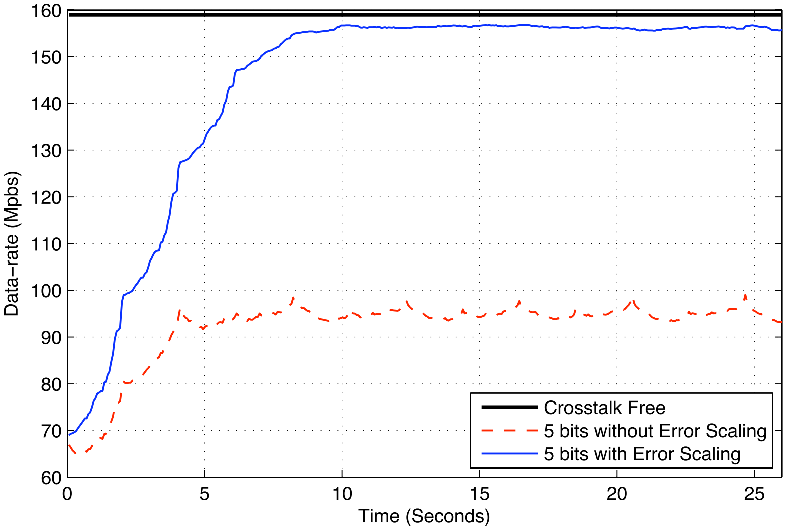

Fig. 10. Error scaling significantly improves the accuracy of crosstalk precoder training

5 Conclusions

Vectoring is an essential technology to enable next generation VDSL2 access networks to achieve their full potential and can drastically improve service data-rates, reach and reliability. Significant work is now being done by industry on vectoring, with much of this work focused on the development of the ITU G.vector standard. This paper introduced some of the challenges faced in creating this standard, and showed how creative solutions, which take advantage of the unique characteristics of the DSL environment, have helped develop vectoring from theory into a practical technology that will soon be ready for commercial deployment. The first issue of the G.vector standard is planned for completion in the October ITU meeting this year. It is believe that widespread adoption of vectoring technology in DSL networks will follow.

References

1. Ginis, G., Cioffi, J.: Vectored Transmission for Digital Subscriber Line Systems. IEEE J. Select. Areas Commun. 20(5), 1085–1104 (2002)
2. Cendrillon, R., Ginis, G., Van den Bogaert, E., Moonen, M.: A Near-optimal Linear Crosstalk Precoder for VDSL. IEEE Trans. on Communications 55(5), 860–863 (2007)
3. Cendrillon, R., Ginis, G., Van den Bogaert, E., Moonen, M.: A Near-optimal Linear Crosstalk Canceler for VDSL. IEEE Trans. on Signal Processing 54(8), 3136–3146 (2006)
4. Cendrillon, R., Yu, W., Moonen, M., Verlinden, J., Bostoen, T.: Optimal Multi-user Spectrum Balancing for Digital Subscriber Lines. IEEE Trans. on Communications 54(5), 922–933 (2006)
5. ITU-T Draft Standard G.vector: Self-FEXT Cancellation (Vectoring) for use with VDSL2 Transceivers, Temporary Document 09XC-R18R3 (2009)
6. Huawei Technologies Co., Ltd., Reducing the Back Channel Overhead of Crosstalk Channel Identification, ITU-T SG15/Q4 contribution 08CC-59 (September 2008)
7. ATIS pre-published Technical Report ATIS-PP-0600024, Multiple-Input Multiple-Output Crosstalk Channel Model (2009)
8. Huawei Technologies Co., Ltd., Speeding up the Convergence of Crosstalk Precoder Initialization with Error Scaling, ITU-T SG15/Q4 contribution 09AG-047 (February 2009)