xDSL Link Quality: *How to Ensure a Stable & Error-free Link for IPTV Deployments*

Steen Garbers Enevoldsen, *Product Manager, COE (Telco/ISP), ZyXEL Communications A/S*

***Abstract*—IPTV types of services require an error free link to ensure a good customer experience. This paper discusses in detail the noise characteristics, and how to optimize the xDSL link parameters to reduce bit errors. The xDSL tool set is presented and a strategy for configuring the line profiles are proposed.**

***Index Terms*—xDSL, Bit error rate, Quality of Service, Impulse Noise Protection, Noise types, Forward Error Correction, Cyclic Redundancy Checksum.**

I. Introduction

OPERATORS are currently deploying IPTV services on xDSL based access infrastructure making the sometimes less than perfect link quality more apparent than ever before, since IPTV types of services put much more strict requirements on the quality of the link, compared to normal internet browsing and even VoIP.

The present document discusses the problem at hand and provides the reader with a tool set to optimize the link quality in terms of reducing the probability of bit errors occurring, as well as providing means to correcting the bit errors that occur anyway.

The advantages of using the forward error correction features of ADSL, ADSL2/2+ and VDSL2 in relation to protection against impulse noise, and other transient types of noises is described in detail.

Note: The scope of this paper is optimizing the xDSL physical layer performance and noise resilience. Higher layer properties such as IP-QoS are not covered.

II. The Problem at Hand

Traditional internet usage, such as browsing and file transfers use TCP (Transmission Control Protocol), which include a retransmission scheme. In practice this mean that a subscriber will most likely never experience the effects of packet loss unless it is really severe. Time-sensitive applications such as IPTV and VoIP use UDP (User Datagram Protocol) that has less overhead and impose less delay to the loop. For these kind of applications the delay is of critical importance, and a dropped packet is preferable to a delay in deliverance of data. Especially for IPTV services one of the major challenges is to keep the 'Zapping-time' at a minimum, as well as ensuring that the delay on a live broadcast show does not increase too much compared to the live feed from other sources, such as DVB-x and analog teristical services.

The quality of a link can be measured on a number of parameters, depending on preference and application. Throughout this paper the focus is on the two measures that have the largest impact on IPTV service:

1) Bit Errors – A measure for how many bits on a link that are erroneous compared to the total number of transmitted bits.
2) Latency – A measure for the delay introduced by the ADSL link.

A. Latency

The latency is not directly affected by the loop itself[1], but can viewed as a consequence of the ADSL characteristics which is highly dependent on the actual profile that is configured. In a nutshell the latency introduced by the link, should always be sought to be minimised, but as further reading of the present paper will show the measures used to minimise the probability of errors on the link, will as a consequence introduce greater latency on the link. The minimum latency introduced by the DSL link is approximately 2 milliseconds, and this latency can increase up to several hundreds of milliseconds depending on the configuration parameters.

B. Bit Error Rate

Whenever a bit is erroneous on the DSL link, the result is a CRC error which in turn mean that a packet is dropped on the IP-layer. A single IP packet can contain 7 MPEG packets, so the visible effect of a bit error for the subscriber watching his IPTV service can be anything from a short glitch in the picture over macro block effect to in some cases (when several consequetive IP datagrams are dropped) a freeze of the picture. So the goal is really to (1) minimize the number of bit errors, and (2) to make sure that the bit errors that occur anyway is evenly spread in time. For a detailed walk through of BER see appendix A.

Document version: 1.0, March 2008
ZyXEL Communications A/S
Columbusvej 5, DK-2860 Søborg
Phone: +45 3955 0700
Fax: +45 3955 0707
http://www.zyxel.dk

[1]Strictly speaking the latency is indeed affected by the loop length, but since the VOP (Velocity of Propagation) on copper is in the order of 200 meters per μs it can be considered to be negligible compared to the delay introduced by the coding, etc.

III. What is good enough?

Traditionally xDSL systems have been designed with a target BER[2] of 10^{-7}, i.e. maximum one bit must be erroneus for every 10 million bits transmitted. Opinions on what is a tolerable level of BER ranges from 10^{-4} for data transmission over ATM to 10^{-12} for high-quality compressed video. The 10^{-7} level was the compromise value that was agreed upon in the ANSI T1E1.4 working group [1]. When discussing the acceptable level of BER it must be remembered that BER is most of all a statistical term. Consider these two scenarios:

1) A subscriber line (@$20Mbps$) suffer from one bit error every second for an hour $\Rightarrow BER = 5 \cdot 10^{-8}$
2) A subscriber line (@$20Mbps$) runs with no errors for 59 minutes but during the last minute a burst of 3600 bit errors occur $\Rightarrow BER = 5 \cdot 10^{-8}$

Which scenario is worst? The resulting BER is exactly the same[3], but the experience for the user will be very different. Assuming that bit errors are non-bursty and evenly distributed over time[4], table I show the expected time interval ΔT between consequitive bit errors on a loop for various bit rates.

TABLE I
TIME ΔT BETWEEN BIT ERRORS AT DIFFERENT BER LEVELS AND BIT RATES.

$R\,[Mbps]$	BER	$\Delta T\,[hh:mm:ss]$
20	10^{-7}	00:00:00:500
20	10^{-9}	00:00:50
20	10^{-12}	13:53:20
10	10^{-7}	00:00:01
10	10^{-9}	00:01:40
10	10^{-12}	27:46:40

Clearly $BER = 10^{-7}\,@\,20\,Mbps$ is unacceptable since this would potentially result in visible errors in the IPTV signal two times every second. So why is xDSL[5] designed at a $BER \leq 10^{-7}$ target, and is this even acceptable?

The answer lies in the fact that a DSL line profile is always defined at a certain target SNR margin (a typical line profile specifies 6 dB margin for both up- and downstream), which mean that assuming pure Gaussian noise distribution, the link will theoretically exhibit a $BER = 10^{-24}$ [2] – in other words 1.5 billion years between bit errors occurring. Alas in the real world, noise is rarely Gaussian and therefore a more realistic number is $BER \leq 10^{-9}$ at 6 dB margin, which empirical data show to be true in 99% of real life installations [2].

A 10^{-9} BER might be sufficient for an IPTV deployment, depending on a number of other factors, and assuming that the predominant source of noise on the local loop is WSS type (see section IV), and in this case the typical line profile with 6 dB target SNR margin would suffice. However if the link suffer from (frequent) transient noise types as well, a higher margin might be desirable. I am deliberatly being vague in this section, since it is really impossible to make a universally valid recommendation. It is extremely important for an operator considering deploying IPTV over DSL to do practical experiments before a full scale deployment.

And it is not really a viable solution to just increase the target margin, since for every dB of margin there is a penalty of several hundred kilobits in achvievable bit rate.

[2]at zero dB SNR margin

[3]Assuming that the BER is monitored over the period of an hour!

[4]This is in practice a false assumption, bit errors tend to come in bursts - See appendix A

[5]The $BER \leq 10^{-7}$ constraint applies to both ADSL, ADSL2, VDSL, VDSL2

IV. Noise Types

Compared to other last mile technologies such as fiber and even copper-based Ethernet, DSL technology must exist in a very uncontrollable and to a degree even unpredictable environment that is subject to a large number of different noise sources. Figure 1 depicts an overview of the local loop environment, where both intrinsic and extrinsic noise sources are mentioned.

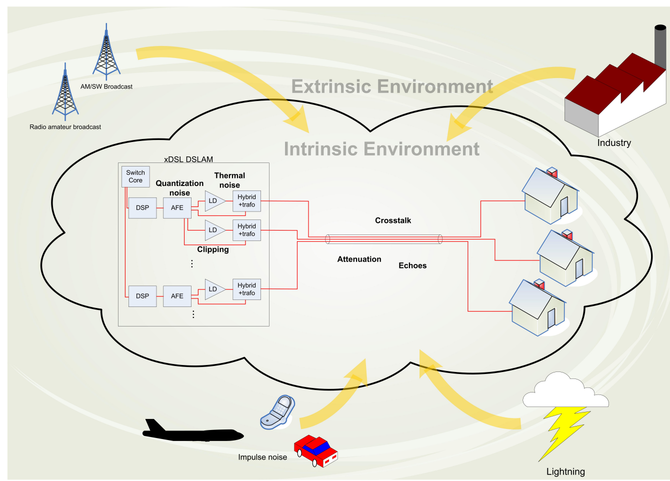

Fig. 1. Intrinsic and extrinsic sources of noise. Source: DSL Forum, TIWG.

Examples of Intrinsic sources of noise:

- Crosstalk from DSL services in adjacent binders or pairs (intrinsic to the DSL-system, but extrinsic to the DSL transceiver)
- clipping in line driver (intrinsic to the DSL transceiver, and hence the DSL-system.)

Examples of extrinsic sources of noise:

- RF interference from broadcast services
- Lightning
- Noise from industrial processes

The noise types in question can be divided into two main groups; the Wide-Sense Stationary (WSS) noise and the transient noise types. WSS noise can be characterized by the fact that it is by nature stationary, or slowly varying, whereas the transient noise types exhibit a very dynamic behavior. The methods used to cope with the two different noise types are very different.

As an example of the sort of noise which has been seen in real life installations, the graph depicted in figure 2 shows a time-domain plot of the noise measured differantially between

Tip and Ring[6] of a twisted pair in the vicinity of a TV set which was affecting ADSL broadband service within about 200 meters of the set's location. The set had a faulty reservoir capacitor in its PSU, but the owner would have been unaware of this, as the set still produced a picture, with just a little ripple visible.

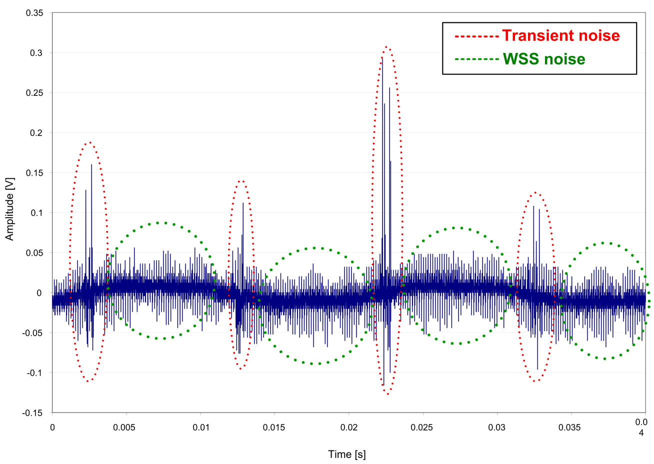

Fig. 2. An example of repetitive impulse noise (REIN), viewed in the time-domain. Source: BT plc

The noise plot consist of both WSS and transient noise, and it is noteworthy that the transient part is repeting it self. This is a quite common behaviour when the transient noise originates from faulty electrical equipment, and therefore some effort have been put into describing this kind of noise in a general way (REIN = Repetitive Electrical Impulse Noise) and specific test cases have been added in the DSL Forum test specifications that address how CPE's handle this particular type of noise.

Noise of transient character will always occur on the line; both induced from external sources, such as faulty switch mode power supplies, lightening surges, POTS signaling etc. Transients will also occur from the transceiver itself, since due to the intrinsic properties of the DMT modulation, clipping will occur at a certain probability, which also exhibits transient properties.

V. XDSL NUTS & BOLTS IN RELATION TO NOISE RESILIENCE

Basically when a DSLAM and CPE is connected and put in active state the two transceivers go through three different states, see figure 3:

1) Handshake - described in ITU-T G994.1, this is the common start procedure for all DSL types. Using PSK modulation the purpose of this phase is detect the DSL type (ADSL or ADSL2, which annex, etc) and some basic properties of the modem.
2) ADSL Training - described in ITU-T G992.3 (or the relevant standard for the mode that was chosen in the Handshake phase), this phase consist of three sub phases:
   - Transceiver training - Used to adapt e.g. the AGC and the Echo canceler
   - Channel analysis - used to analyze the line transfer function and noise characteristics.
   - Exchange - Used to transfer framing parameters, bit loading, etc.
3) Showtime - this is when data traffic can be passed through the link.

[6]Tip and Ring is a common term for the two wires of an ordinary telephone pair. It originates from the good old days, from the operators' phone plug which were used to manually switch calls. The names of the wires are derived from the part of the plug to which they are connected.

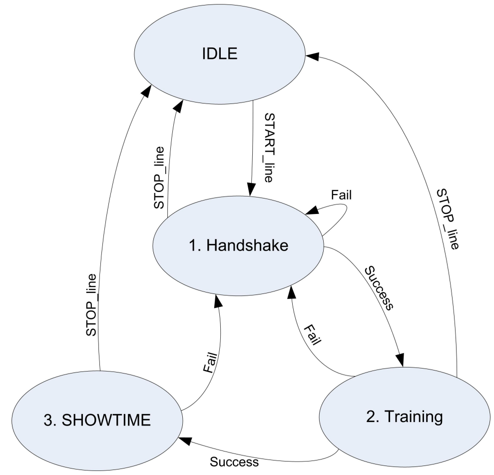

Fig. 3. xDSL transceiver initialization state machine.

As can be seen from the above description the ADSL transceiver's first method of coping with noise is done before even reaching SHOWTIME, in the "channel analysis" part of the "ADSL training" phase. In fact the receiver takes a snapshot of the current noise conditions on the line, and subsequently bases the choice of bit's and gain's tables on this information.

Here we have one of the main points regarding the difference between stationary and impulsive noise types - since the above channel analysis assumes that the noise scenario does not change over time.

In order to cope with the fact that the noise do change over time, several measures are taken into use, of which the most important are:

1) SNR margin – The default configuration of the transceiver allow for a target margin of 6dB. In other words the noise can increase by 6 dB, without causing the BER to increase above 10^{-7}. See section III.
2) Bit-swap – An algorithm is provided in both ITU-T G992.1 (optional) and G992.3/5 (mandatory) that enables the transceivers to reduce the bit loading on a sub carrier in case the SNR on that particular sub carrier decreases.
3) Seamless Rate Adaptation (SRA) – Reconfigures the total data rate of a line by modifying framing parameters as well as bit loading and fine gains on sub carriers, thereby adapting to changing noise conditions.

A. SNR margin

This method provides protection primarily against wide-band noise, but the level of the noise cannot increase much before bit errors will occur anyway, in particular in the case where a narrow-band noise-spike emerges from the noise floor[3].

B. Bit swap

An important limitation in the bit swap algorithm is that, since the overall number of bits per symbol shall be constant during showtime, the bit loading shall be correspondingly increased on another sub carrier (see figure 4). This method provides good protection against narrow-band noise, but with an important limitation: Changing the bit loading requires extensive synchronization between the transmitter and the receiver, and the protocol implemented for this purpose is quite slow (we're talking in the order of 10's to 100's of milliseconds per bit swap).

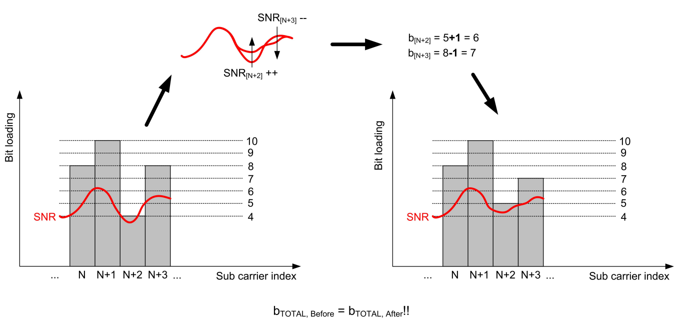

Fig. 4. Example of a bit swap. Noise on sub carrier N+2 decreases, while noise on sub carrier N+3 increases. This result in one bit being reallocated from sub carrier N+3 to sub carrier N+2.

Therefore bit-swap only works if:

- The noise amplitude does not increase too rapidly.
- The power spectral density of the noise is narrow enough such that not too many sub carriers are simultaneously affected by the noise.
- The SNR on an other sub carrier is good enough to support an increase in bit loading.

Impulse noise usually has a wide spectral content, and the amplitude is by nature rapidly increasing. Therefore bit swap provides little protection against impulsive noise types.

C. Seamless Rate adaptation

Seamless Rate Adaptation (SRA) is used to reconfigure the total data rate of a line by modifying framing parameters as well as bit loading and fine gains on sub carriers, see figure 5.

Four management parameters are defined for SRA:

1) Upshift Margin: Threshold for increasing the data rate.
2) Upshift Time: Minimum time for which Current SNR $>$ Upshift Margin.
3) Downshift Margin: Threshold for decreasing the data rate.
4) Downshift Time: Minimum time for which Current SNR $<$ Downshift Margin.

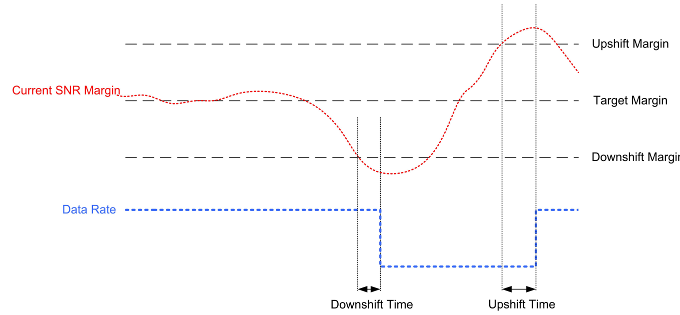

Fig. 5. SRA functionality described. The overall data rate is decreased and increased autonomously in order to keep the current margin close to the target margin.

If the current SNR margin drops below the defined threshold (downshift margin) for a predefined time (downshift time) the data rate is reduced in order to increase the margin. Similarly if the margin increases above "Upshift Margin" for at least "Upshift Time" the data rate is increased.

Using this functionality the link can be trained to the maximum acheivable data rate, and the link will in SHOW-TIME automatically adapt to changing line conditions without interrupting the data flow.

SRA is not really efficient against impulse noise, but is targeted towards varying crosstalk levels from adjacent pairs or binders, which is typically the case when subscribers turn on/off their CPE's causing the crosstalk level to jump from time to time, as seen by a CPE that is always on.

D. Reed Solomon Forward Error Correction

Where the stationary noise has a deterministic character (i.e. it's always present), the transient or impulse noise has a nondeterministic character, making the protection from it more difficult. One choice is of course to configure the transceivers to a high target margin to provide a sufficient "buffer" to cope with sudden transients, but this approach has the obvious drawback that the attainable bit rate decrease with higher target SNR.

Another approach is to accept the fact that bit errors will occur from time to time as a result of transients on the line, and apply some kind of error correction to cope with the bit errors. In ADSL/ADSL2/ADSL2+/VDSL2 this mechanism is provided by means of applying Reed-Solomon coding to the data.

Reed-Solomon coding (for details see appendix B) is a forward error correction (FEC) technique where redundant information is added to each frame, and based on this the receiver is capable of correcting a certain number of bit errors. As a consequence the user will not experience any packet loss, even though a bit error occur on the line. This means that instead of the ES (errored seconds) counter the ECS (error correction second) will increase (more details on O&M counters in appendix C). This counter means that during that particular second, one or more bit errors occurred on the line, but was corrected by the FEC algorithm. In other words the subscriber will not see the bit error.

The limitation in RS-coding is that while single bit errors are easily corrected; a burst of bit errors (i.e. many consecutive erroneous bits) cannot be corrected.

Impulse noise usually results in bursts of bit errors, so in order to improve the efficiency of the RS coding algorithm the data can be passed through a convolutional interleaver (figure 6 depict a block interleaver, which demonstrates the interleaving principle better than the convolutional type), which spreads the bytes from several RS code words into one DMT symbol. The advantage in this approach is that even though a burst of bit errors occurs on the line, the bit errors are spread across several RS codewords, once the data stream is de-interleaved by the receiver, and hence the RS algorithm is capable of correcting the (now single) bit errors.

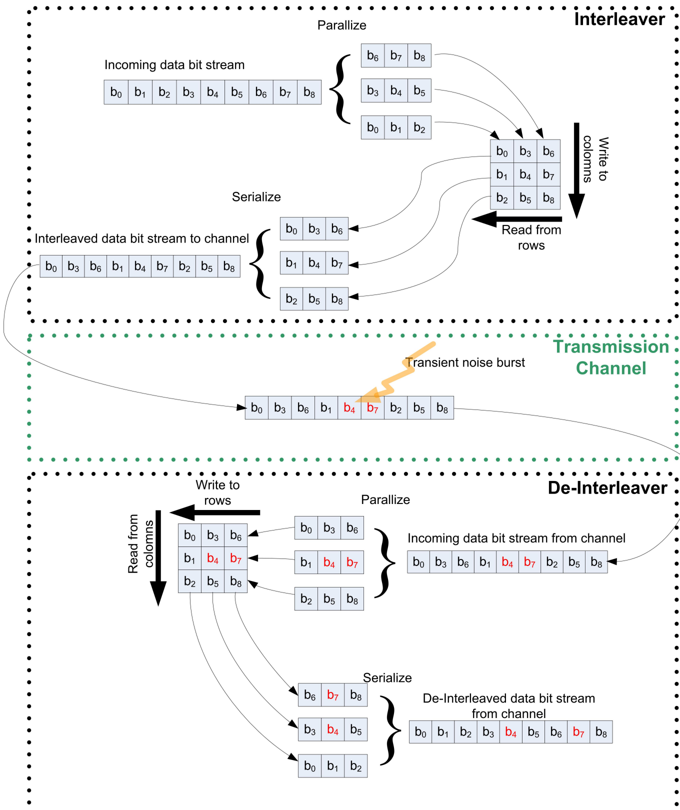

Fig. 6. The basic principle of operation for a block interleaver/de-interleaver. Note that the two adjacent bits errored due to a transient on the line is seperated after the de-interleaving. Note that xDSL use a convolutional interleaver which is 2-4 times as memory efficient and introduces less delay, compared to the block interleaver [2].

The drawback in using this method is that an additional latency is introduced to the transmission path, depending on the selection of coding and framing parameters. The delay introduced by the interleaver between the PMS-TC and PMD sub layers of the reference model (see [4], Figure 5-2), i.e. between reference points α and β can be expressed as (applies to ADSL2 and ADSL2+):

$$delay_{\alpha-\beta} = 3.75 + \frac{\lceil S_p \cdot D_p \rceil}{4}\, ms \qquad (1)$$

Where:

- $\lceil x \rceil$ denotes rounding to higher integer.
- S_p denotes the number of PMD symbols over which the RS code word (also called the FEC data frame) spans.
- D_p is the interleaving depth in latency path $\#p$.

For VDSL2 the delay is calculated differently due to the different framing construction [5, section 9.7]:

$$delay_p = \frac{S_p \cdot (D_p - 1)}{q_p \cdot f_s} \cdot \left(1 - \frac{q_p}{N_{FECp}}\right) ms \qquad (2)$$

Where:

- D_p denotes the interleaving depth in latency path p.
- S_p denotes the number of PMD symbols over which the RS code word (also called the FEC data frame) spans for latency path p.
- q_p is the number of interleaver blocks in an FEC codeword for latency path p.
- N_{FECp} is the FEC codeword size for latency path p.
- f_s is the data symbol rate in $ksymbols/s$.

Common for VDSL2 and ADSL2+ the relation between the parameters can be crudely expressed as: *The "R-S Coding + Interleave" scheme will be capable of protecting against impulses of longer duration for higher values of S_p and D_p.*The prize to be paid is that the resulting latency will increase accordingly.

E. Impulse Noise Protection

The forward error correction scheme is basically the same for ADSL, ADSL2(+) and VDSL2, but by request from operators a new handle has been added to the ADSL2/2+/VDSL2 standards: INP_{min}. This handle addresses a concern expressed by many operators regarding the delivery of the new high-bandwidth video services. The shortcoming in ADSL is that one can only specify the upper boundary on the interleaving delay, hence ensuring that the resulting latency through the total transmission path stays below a certain limit[6].

The problem is that this does not guarantee that any usable impulse noise protection is actually applied to the line. This makes it very difficult to create a generic line profile to be used for video services that require a stricter maximum BER in order to guarantee a fair quality perceived by the user.

To solve this ADSL2, ADSL2+ and VDSL2 provides a handle for the minimum impulse noise protection that is required for the line. The maximum interleaving delay parameter from ADSL is still provided as well, so now it is possible to specify both a minimum impulse noise protection that is desired for the link as well as specifying an upper boundary on the resulting latency.

The INP_{min} factor is described in terms of DMT symbols, in other words an $INP_{min} = 1$ means that the errors originating from an impulse transient of one DMT symbol duration can be corrected. One DMT symbol correspond to 250 μs. It can be argued that any values of INP_{min} less than 1 makes little sence, and even one does not guarantee error recovery from one DMT symbol, since the impulse noise might at a certain probability hit the boundary between two frames, hence causing erasure of two DMT symbols.

Therefore an ammendment to ADSL2(+) was made allowing for higher values of D (which is required in order to support INP_{min} without decreasing the bit rate)[7].

Practical experiments suggest that $INP_{min} \geq 2$ will provide the best impulse noise protection.

VI. MONITORING THE PERFORMANCE OF AN XDSL LINK

Monitoring the quality of the line makes sence both before and after the xDSL service has been enabled, however the methods are quite different.

A. Pre-deployment performance evaluation

Pre-deployment performance evaluation is used to estimate if the loop quality is sufficient for a subscriber to reveive a particular DSL service. The most accurate way of doing this is to place specialized measurement equipment on both the central and the customer side, but this is a too expensive method. Therefore a tool based on SELT (Single-Ended Line Testing) is commonly used. Many DSLAMs have this kind of functionality built-in. The technique is based on time-domain reflectometry, where a pulse is sent from the CO side and the received echo is analyzed. This can give a good estimation on the loop length, as well as the channel frequency response. Furthermore the steady-state (WSS) noise level can be measured using this tool. Based on the results the tool can estimate the maximum attainable bit rates for DSL service on the loop. The accuracy of the loop length estimation is typically quite good (3-6 %), but the estimation of the maximum attainable bit rate is difficult, since you basically only have the central side point-of-view (especially if the tool is built into the DSLAM it can only monitor the upstream frequency range). Therefore it is normal to have a rather large error-margin on those kind of estimations (of course depending on the actual tool used - some might themselves have this margin incorporated in ther results).

B. Post-deployment performance evaluation

Monitoring the xDSL line performance counters while the service is deployed means that the operator might be able to proactively react on emerging problems, before the subscriber is even aware of a service degradation. Furthermore line monitoring can be used to keep the operator aware of how far the cable binder is from it's capacity limits. I.e. before offering DSL service to a new subscriber existing data from other modems already deployed in the same cable binder can be used to evaluate if there is sufficient capacity (in terms of noise margin on existing modems) for the service to be deployed without affecting existing customers.

C. Diagnostics mode

A diagnostics mode is offered by ADSL2(+) (it is coming for VDSL2 as well) enabling a detailed loop measurement that can be used to evaluate if e.g. a service upgrade to higher bit rates is possible, but also this tool is useful when diagnosing a problem at a customer. The advantage in this tool is that the CPE and DSLAM is used for the measurements, so it is not necessary to have a technician visit the customer to do the diagnosis of the loop.

Furthermore the accuracy of this measurement is very good, since the results are what the actual devices see. Also when performing the loop diagnostics the COE and CPE go through a process very similar to a normal initialization sequence, so the results are quite dependable.

VII. CONCLUSION: THE PERFECT LINEPROFILE

Now that the basics of xDSL transmission and noise scenarios are described we will try to define the “perfect” line profile for IPTV types of service. Hopefully by now the reader will have realized that no such thing can be defined, due to the complexity of the problem at hand.

To recap the task we wish to configure a line profile that facilitates IPTV in terms of noise resellience and at the same time try to keep the imposed latency at a minimum, while ensuring a sufficient achievable data rate to be able to offer the service to a maximum of potential customers.

The default SNR margin of 6 dB might be insufficient, and the operator is encouraged to consider increasing the SNR margin to 9 dB, if the resulting loss in loopreach is acceptable. Roughly speaking if you increase the SNR margin by one dB the achievable performance will decrease by 200 kbps.

xDSL offer both fast path and interleaved path types of lineprofiles, but considering the importance of keeping the BER at an acceptable level, only the interleaved path makes sense. Furthermore to minimise the probability of errors on the link, the INP_{min} factor should be used, and it is advisable to configure the INP_{min} to at least one DMT symbol or even two.

SRA can be used to allow the line to adapt on-the-fly to changing noise conditions, but as a consequence the bit rate perceived by the user will also change. This could be a problem depending on the type of service offered to the subscribers (i.e. guaranteed minimum bit rate, or best effort).

Configuring the line to use interleaving, and INP_{min} of at least one DMT symbol (and preferably 2) will introduce a noticable latency compared to a line configured using the fast path. As an example, an ADSL2+ line running at 10 Mbps with $INP_{min} = 2$ will have an increased latency of 10 ms[8], compared to the same line configured to fast path operation.

Note that using the INP_{min} and traditional Maximum latency parameters makes it possible to make an illegal line profile. If using $INP_{min} > 0$ a consequence is that the resulting delay will be > 0, and hence the Maximum Latency parameter shall be set to a value > 0 to allow for this - otherwise the line will never enter SHOWTIME.

If an increased latency is a problem for the subscribers requiring a gaming-type of service, multiple latency path should be considered. This method allows the operator to configure the line to use more than one latency path, thereby

[7]See document HA-R17A3 from ITU-SG15 for details on this subject.

[8]A practical experiment. The resulting latency can differ depending on the framing parameters that are autonomously chosen by the COE and CPE in the actual situation.

making specific profile for IPTV and for gaming/normal internet surfing. The latency paths is usually seperated on the ATM layer by assigning different PVC's to the two latency paths, and on the CPE by assigning different physical ports to each PVC. Note that this functionality is optional and not necessarily supported on all devices.

The importance of performing field tests before a full deployment should be apparant - this test should not only be used to test the IPTV STB et cetera, but also be used to test the chosen lineprofile, and monitor the line quality in order to evaluate the effectiveness of the noise resellience.

APPENDIX A
BER DEFINED

BER is an acronym for Bit Error Rate, which in short can be described as the number of erroneous bits per error-free bits on a link. Let N be the number of transmitted bits, and τ the number of errored bits, then:

$$BER = \frac{N}{\tau} \tag{3}$$

The BER definition origins from 'old times' and actually assume a serial data stream, hence the correct way to accurately measure BER is to use an external instrument transmitting a serial data stream into the Tx-transceiver and measuring the serial data stream from the Rx-transceiver. The instrument then compares the transmitted to the received data stream and calculates the corresponding BER. Although a serial interface is actually described for ADSL transceivers it is very rarely used, making the above described method un-usable for typical xDSL equipment (there is no way to connect a BER test tool).

Some ADSL chipsets have a BER measuring tool integrated that circumvents this limitation, but this is neither common nor a mandatory feature. The only mandatory 'error measurement tool' that is provided by xDSL is CRC error counting, and therefore in the DSL Forum Testing & Interoperability Working Group (TIWG) it was proposed by Texas Instruments to use the CRC error reporting to estimate the BER. This method was adopted for the TR-067 (ADSL interoperability performance test specification) for the margin verification tests and is subsequently used in TR-100 (ADSL2+ interoperability performance test specification).

A. How to estimate BER: Use CRC error counters

The challenge in using CRC errors to monitor BER is that the CRC is calculated over an entire ADSL super frame, and consequently one CRC error might correspond to one single or one thousand bit errors. This problem has been addressed by doing practical measurements to find the average correlation between CRC errors and bit errors.

Firstly we establish the basic formula for the BER in relation to monitored CRC errors:

If R is the data rate in bits per second (bps), T the observation time in seconds (s), CRC the counted CRC errors during the observation time and E_{CRC} is the number of bit errors that correspond to a single CRC error then,

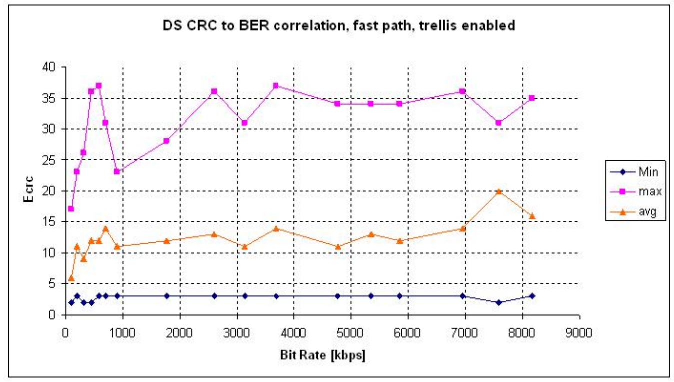

Fig. 7. Downstream CRC to BER correlation. Fast path, trellis enabled. Source: DSL Forum, TIWG.

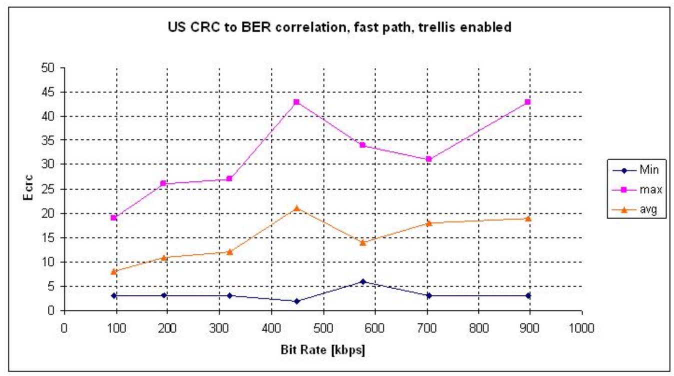

Fig. 8. Upstream CRC to BER correlation. Fast path, trellis enabled. Source: DSL Forum, TIWG.

$$BER = \frac{E_{CRC} \cdot CRC}{R \cdot T} \tag{4}$$

The tricky part is to identify the constant E_{CRC} as an expression for the correlation between CRC and BER, which is done by a combination of transmission path analysis and practical measurements.

Since the correlation between CRC errors and bit errors is very dependent on the used latency path, the subsequent description is split into separate sections for the fast and the interleaved data path.

However practical experience with the method reveal a good dependability, it is important to note that the methods described in the subsequent sections is not an accurate BER measurement - It merely provides an estimate of the BER.

1) BER to CRC correlation for Fast latency path: Figures 7 and 8 show a compilation of a number of tests where the CRC error counter has been monitored on a link. The accurate BER have been measured for the same link, hence showing the numbers of bit errors that one CRC error correspond to as a function of bit rate.

As can be seen the E_{CRC} is largely independent from the bit rate, and furthermore the average E_{CRC} is between 10 to 20 bit errors per CRC error. To decrease the risk of underestimating the BER it was decided in the TIWG to define, that for the fast path one CRC error correspond to 20 bit errors.

Using this we can refine our formula from equation 4:

$$BER_{FASTPATH} = \frac{20 \cdot CRC}{R \cdot T} \tag{5}$$

And to make things practical for measurements we refine the formula to state the minimum required time to monitor the link to ensure that a BER threshold is not exceeded. Furthermore to increase the reliability of the test we monitor not one, but 10 consecutive CRC errors:

Let

- ΔT = The minimum time to monitor the link for 10 CRC errors to occur.
- BER = The assumed BER for the test.
- E_{CRC} = The number of bit errors per CRC error.
- R = The data rate of the link.

Then,

$$\Delta T = \frac{10 \cdot E_{CRC} \cdot \frac{1}{BER}}{R} \tag{6}$$

I.e. for $E_{CRC} = 20$, and $BER = 10^{-7}$: $\Delta T = \frac{200 \cdot 10^{-7}}{R}$

So for a practical example, a link trained at a data rate of 10 Mbps (Fast path!) can be considered to have a BER of 10^{-7} or better, if no more than 10 CRC errors are detected during a time interval of:

$$\Delta T = \frac{200 \cdot 10^{-7}}{10 \cdot 10^{6}} = \underline{\underline{200s}} \tag{7}$$

Obviously from equation 6, the test duration depend on both the BER level and the data rate. The lower the BER the longer the test, and similarly the lower the data rate, the longer the test time.

Table II show calculations for the estimated test duration for various combinations of bit rate and target BER.

TABLE II
ESTIMATED TEST DURATION FOR BIT RATES BETWEEN 0,5 TO 25 MBPS FAST PATH, AND FOR $BER = 10^{-7}$ AND 10^{-9} RESPECTIVELY.

BER	$R\,[Mbps]$	$\Delta T\,[s]$	BER	$R\,[Mbps]$	$\Delta T\,[s]$
10^{-7}	25	80	10^{-9}	25	8000
10^{-7}	20	100	10^{-9}	20	10000
10^{-7}	18	111	10^{-9}	18	11111
10^{-7}	15	133	10^{-9}	15	13333
10^{-7}	12	167	10^{-9}	12	16667
10^{-7}	10	200	10^{-9}	10	20000
10^{-7}	8	250	10^{-9}	8	25000
10^{-7}	6	333	10^{-9}	6	33333
10^{-7}	4	500	10^{-9}	4	50000
10^{-7}	2	1000	10^{-9}	2	100000
10^{-7}	1	2000	10^{-9}	1	200000
10^{-7}	0.5	4000	10^{-9}	0.5	400000

Clearly this test methodology is not suitable for lower bit rates, which cause the needed test time to skyrocket. This applies to any type of BER test, since at lower bit rates the line must be monitored for a very long time for any bit error to occur.

2) BER to CRC correlation for interleaved latency path: When estimating the BER to CRC correlation for the interleaved path we need to take the Reed-Solomon algorithm into consideration. A Reed-Solomon codeword with R redundant check bytes can correct up to $R/2$ byte errors per codeword. A codeword with $(R/2+1)$ or more byte errors is uncorrectable and will result in bit errors in the user data.

With 16ms of interleaving latency, the byte error distribution can be considered fairly random, and therefore the most common error event will occur when there are exactly $(R/2+1)$ byte errors per codeword. As a result, the expected (i.e. average) number of bit errors per uncorrectable codeword is given by:

$$E_{RS} = \left(\frac{R}{2} + 1\right) \cdot \Delta\tau \tag{8}$$

Where $\Delta\tau$ is the average number of bit errors per byte errors. This number varies between 1 and 2 depending on the constellation size and is for these calculations assumed to be $\Delta\tau = 2$.

ADSL(2)/VDSL2 use a self synchronized scrambler to minimize the likelihood that a long sequence of zeros will be transmitted over the channel, hence ensuring that the PSD on the line is constant and independent from the actual data passing over the line. The descrambler is characterized by the equation (G993.2 9.2):

$$d'_n = d_n \oplus d'_{n-18} \oplus d'_{n-23} \tag{9}$$

Where d'_n is the n^{th} input to the scrambler and d_n is the n^{th} output from the scrambler. Obviously for each error in d' there will three errors in d except for the rare occasions where errors in d' happens exactly 5, 18 or 23 bits apart. Therefore the self synchronized scrambler can be considered to multiply the error-rate by approximately 3.

A CRC error will typically be caused by one single uncorrectable codeword, since at low bit error rates the uncorrectable codewords will be evenly spaced in time. If the number of codewords per symbol S is greater than one occasionally an uncorrectable codeword will span two superframes, and consequently cause two CRC errors.

The expected number of bit errors per CRC error for the interleaved data path can then be expressed as:

$$E_{CRC} = 3 \cdot E_{RS} = 6 \cdot \left(\frac{R}{2} + 1\right) \tag{10}$$

Figure 9 depict the result from a large number of practical measurements (covering many loop lengths and data rates) compared to the analytical results.

As can be seen the average measurements correspond quite nicely to the numbers derived from the above analysis. The graph show that the number of bit errors per CRC error increase as the coding overhead R increases. Since it is assumed that when using interleaved operation the typical framing parameters will select preferably the larger ($R \geq 8$) values for R, the agreed compromise is to correspond one CRC error to 50 bit errors.

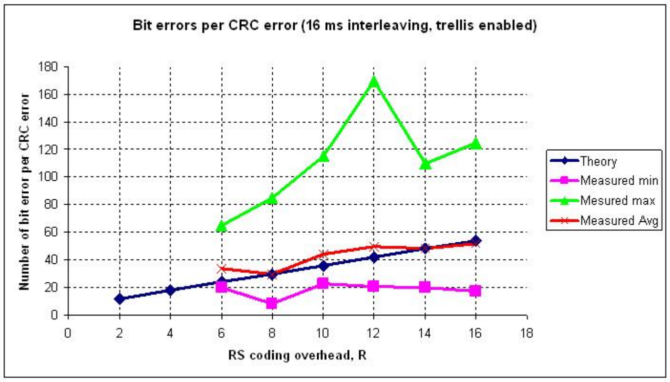

Fig. 9. Measured compared to calculated number of bit errors per CRC error for interleaved path operation. Source: TIWG.

This enables us to define the BER formula in the same way as for the fast data path:

$$BER_{INTERLEAVEDPATH} = \frac{50 \cdot CRC}{R \cdot T} \tag{11}$$

To increase the reliability of the test we monitor not one, but 10 consecutive CRC errors:

Let

- ΔT = The minimum time to monitor the link for 10 CRC errors to occur.
- BER = The assumed BER for the test.
- E_{CRC} = The number of bit errors per CRC error.
- R = The data rate of the link.

Then,

$$\Delta T = \frac{10 \cdot E_{CRC} \cdot \frac{1}{BER}}{R} \tag{12}$$

I.e. for $E_{CRC} = 50$, and $BER = 10^{-7}$: $\Delta T = \frac{500 \cdot 10^{-7}}{R}$

So for a practical example, a link trained at a data rate of 10 Mbps (Interleaved path!) can be considered to have a BER of 10^{-7} or better, if no more than 10 CRC errors are detected during a time interval of:

$$\Delta T = \frac{500 \cdot 10^{-7}}{10 \cdot 10^{6}} = \underline{\underline{500s}} \tag{13}$$

Table III show calculations for the estimated test duration for various combinations of bit rate and target BER for the interleaved data path.

TABLE III
ESTIMATED TEST DURATION FOR BIT RATES BETWEEN 0,5 TO 25 MBPS INTERLEAVED PATH, AND FOR $BER = 10^{-7}$ AND 10^{-9} RESPECTIVELY.

BER	$R\,[Mbps]$	$\Delta T\,[s]$	BER	$R\,[Mbps]$	$\Delta T\,[s]$
10^{-7}	25	200	10^{-9}	25	20000
10^{-7}	20	250	10^{-9}	20	25000
10^{-7}	18	278	10^{-9}	18	27778
10^{-7}	15	333	10^{-9}	15	33333
10^{-7}	12	417	10^{-9}	12	41667
10^{-7}	10	500	10^{-9}	10	50000
10^{-7}	8	625	10^{-9}	8	62500
10^{-7}	6	833	10^{-9}	6	83333
10^{-7}	4	1250	10^{-9}	4	125000
10^{-7}	2	2500	10^{-9}	2	250000
10^{-7}	1	5000	10^{-9}	1	500000
10^{-7}	0.5	10000	10^{-9}	0.5	1000000

APPENDIX B
REED SOLOMON FORWARD ERROR CORRECTION

All DSL flavours provide a means of both detecting, and correcting (to a certain extent) the bit errors that occur on a line. Common for all flavours are that Reed-Solomon coding is used for this purpose, providing coding gain against random errors[9].

The basic idea behind Reed-Solomon coding is to encode the data by constructing a polynomial based on the data. The receiving end can then reconstruct the data by exploiting a theorem from linear algebra that states that any k distinct points uniquely determine a polynomial of degree at most $k-1$. Using Reed-Solomon coding, if we have k data bytes and add R redundant bytes, the k original bytes can be reconstructed as long as no more than $R/2$ bytes are received in error.

[9]If correctly used it is possible to get approximately 3 dB coding gain[10] from the RS-coding[2].

A. Galois Field arithmetic

The arithmetic for DSL is performed byte-wise in the finite Galois Field $GF256$, which can be stated as successive powers of a primitive element:

$$GF256 = \{0, 1, \alpha, \alpha^2, \alpha^3, \cdots, \alpha^{254}\} \tag{14}$$

The element α is considered as a root of the binary polynomial

$$\alpha^8 + \alpha^4 + \alpha^3 + \alpha^2 + 1 = 0 \tag{15}$$

And since we're working in the Galois Field, and the arithmetic used in equation 15 is binary, then α is basically just a dummy-value, i.e. not 0 or 1 but just a polynomial variable. Using normal arithmetic the polynomial in equation 15 has in fact neither 0, nor 1 as a root and cannot be factored with binary arithmetic. But when viewed in the $GF256$ using $GF256$ arithmetic $\{\alpha, \alpha^1, \alpha^2, \alpha^4, \alpha^8, \alpha^{16}, \alpha^{32}, \alpha^{64}, \alpha^{128},\}$ are roots of the equation, and is factorable in $GF256$ into the products of factors of $(x - \alpha^i)$, i.e.:

$$\alpha^8 = \alpha^4 + \alpha^3 + \alpha^2 + 1 \tag{16}$$

reproducing α^8 in terms of powers of α less than 8 means that all the elements of $GF256$ can be expressed in terms of polynomials with binary coefficients in powers of α less than 8. In other words any data byte can be associated with an element in $GF256$ by considering the individual bits in the byte to be the binary coefficients in a polynomial with powers $\alpha^7, \alpha^6, \ldots, \alpha^0 = 1$, which can be written as a vector of coefficients $[b_7, b_6, \ldots, b_0]$. Addition is in $GF256$ simply a vector addition (element by element) and multiplication is done by by polynomial multiplication through substitution for any powers of α exceeding 7.

B. Reed-Solomon Coding for ADSL2

As an example the FEC procedure (i.e. RS-coding) implemented in ADSL2 is described. Basically the RS coding is ADSL and VDSL is the same with minor differences due to the different framing.

Input to the RS scheme are $M_p \times K_p$ Mux data frames from the scrambler described in section A, comprising the message bytes, $m_0, m_1, \ldots, m_{M_p \times K_p - 2}, m_{M_p \times K_p - 1}$. The procedure shall creater R_p redundancy bytes $c_0, c_1, \ldots, c_{R_p - 2}, c_{R_p - 1}$. Appending the redundancy bytes to the comprise the FEC codeword of size $M_p \times K_p \times R_p$ bytes.

The redundancy bytes are generated from the message bytes using the equation:

$$C(D) = \mathcal{M}(D) \cdot D^{R_p} \, modulo \, \mathcal{G}(D) \tag{17}$$

Where:

- $\mathcal{M}(D) = m_0 D^{M_p \times K_p - 1} + m_1 D^{M_p \times K_p - 2} + \cdots + m_{M_p \times K_p - 2} D + m_{m_p \times K_p - 1}$ is the message polynomial.
- $C(D) = c_0 D^{R_p - 1} + c_1 D^{R_p - 2} + \cdots + c_{R_p - 2} D + c_{R_p - 1}$ is the check polynomial.
- $\mathcal{G}(D) = \prod_{0 \geq i \geq (R_p - 1)} \left(D + \alpha^i\right)$ is the generator polynomial.

In other words, the check polynomial can be found as the remainder from dividing the message polynomial $\mathcal{M}(D) \cdot D^R$ by the generator polynomial $\mathcal{G}(D)$.

APPENDIX C
O&M MANAGEMENT PRIMITIVES AND COUNTERS

Numerous counters and other O&M handles are defined in [7] for xDSL that enable the subscriber to monitor the link. It is important to note the syntax of the counters since these indicates if it is upstream or downstream and which latency path that the counter relates to.

An error/anomaly can be:

- Near-End, i.e. if the counter is polled on the DSLAM it relates to the upstream direction.
- Far-End, i.e. if the counter is polled on the DSLAM it relates to the downstream direction.
- Related to the Interleaved data path (where applicable)
- Related to the Fast data path (where applicable)

I will not walk through all counters and parameters here, but just briefly touch the ones that are of particular interest to the topic in this paper. For more information the reader is encouraged to read [7].

1) steady state noise monitoring: When monitoring the WSS noise conditions the best parameter is the Actual Signal-to-Noise ratio (SNR) margin [7, section 7.4.4-5] as well as the total output power [7, section 7.4.4-5], since these two parameters indicates how close the link is to the maximum threshold. If the actual margin drops below the target margin, and the output power is at the maximum level, this indicates that any further increase in noise will cause the BER of the link to increase, or even cause the link to drop.

2) Transient noise monitoring: The transient noise cannot be monitored, but the effects from the transients can be monitored in terms of CRC errors. The parameters that should be monitored are the CVI and CVF (Code Violation for the Interleaved and the Fast path). A code violation is basically a CRC-8 error computed on the Mux data frame. It is possible to estimate the Bit Error Rate (BER) from the CV-I/F counters, see appendix A for details.

When ever a CV-anomaly is reported it means that there has been an error in the data (i.e. perceivable by the subscriber).

If running in Interleaved mode the CVI counter means that an error has occurred, and the RS coding was unable to correct the error (i.e. the error was perceivable by the subscriber). If the error was corrected by the RS coding scheme the CVI counter does not increment, but the Forward Error Correction Count Line (ECI) counter will increase. An ECI event means that there was an error in the data, but the error was corrected by the RS coding scheme (i.e. the error is NOT perceivable by the subscriber). This counter also exist for the fast latency path, since the RS coding scheme also apply to this latency path, although not very effiently so the probability of ECF instead of CVF is not very good.

To analyze the types of errors it is advisable to use the "Second" counters. The Errored Second (ES) counter increments whenever one or more CRC-8 anomalies (CV events) has occurred during a running 1-second period. So if the CV counter is high, but the ES counter is low, it means that a single burst of errors have ocurred, and this could be anything - a passing car, or some other freak incident. Anyway no action can be taken to correct such incident.

However if the CV counter is high and the ES counter is also high, it means that repetitive transient errors are ocurring (See the REIN definition in section IV), and that action need to be taken to resolve this issue in terms of identifying the source of the noise, or alternatively altering the line profile to better cope with the transient noise.

Similar to the ES and CV-I/F duality there is an ECS - EC-I/F duality. ECS mean Error Correction Seconds, and this counter increase whenever an error has been corrected by the RS coding scheme. Since this counter does not constitute subscriber perceivable errors it might not be as important as the ES counter. However monitoring this counter and storing the data into a historical database gives a good indication on how the quality of the link evolves over time, which could give an early warning before the link quality degrades enough to actually cause CRC errors.

If the number of CRC-8 anomalies during a one second interval exceed 18, the Severily Errored Second (SES) counter increments instead of the ES counter. Differing between the ES and SES counters gives a rough indication on the severity of a problem.

If 10 consequetive seconds are SES, the UAS counter start to increase. This counter mean Unavailable Seconds, and during a UAS second no data can be passed through the line. The UAS counter stop incrementing after 10 consequetive seconds with no SES, whereafter user data again can pass through the line.

For practical use a term "Threshold Report" is defined.

This basically means that 15-minute and 24-hour counters for at least ES, SES and UAS (other counters are optional) are defined. For each interval a threshold can be defined, such that if the number of events within the 15-minute or 24-hour window is exceeded, an alert (usually in terms of an SNMP trap) is generated.

REFERENCES

[1] T1E1.4, "Network and customer installation interfaces – asymmetric digital subscriber line (adsl) metallic interface," ANSI, Standard T1.413-1995, 1995.
[2] T. Starr, J. M. Cioffy, and P. J. Silverman, *Understanding Digital Subscriber Line Technology*, 1st ed. NJ, USA: Prentice Hall, 1999.
[3] F. Xiong, *Digital Modulation Techniques*, 1st ed. Artec House, Inc., 2000.
[4] "Asymmetric digital subscriber line transceivers - 2 (adsl2)," ITU, Standard G992.3, 2005.
[5] "Very high speed digital subscriber line transceivers 2 (vdsl2)," ITU, Standard G993.2, 2006.
[6] J. A. C. Bingham, *ADSL, VDSL, and Multicarrier Modulation*, 1st ed. John Wiley & Sons, Ltd, 2000.
[7] "Physical layer management for digital subscriber line (dsl) transceivers," ITU, Standard G997.1, 1999.

Steen G. Enevoldsen received his MS degree in electronics in 1999 from Aalborg University. Steen is a member of the IEEE, and has previously designed xDSL CPE equipment for Intel Corporation, and xDSL IP-DSLAM equipment for Ericsson Corporation. Steen has been an active contributor to the DSL Forum, where he held a position as editor of TR-067 Issue II for some time. Working for Ericsson Steen has actively participated in numerous xDSL plugfests held by the ADSL Consortium at University of New Hampshire, as well as plugtest events held by the ETSI plugtest service. Before joining ZyXEL steen worked for YouSee A/S (formerly known as TDC Kabel TV - the largest Cable MSO in Denmark) with focus on HDTV DVB/IP hybrid Set-top-box design. Mr. Enevoldsen (Email: sge@zyxel.dk) currently hold a position as Product Manager for COE equipment for the ISP/Telco business segment covering the Nordic region.