SECURING AUDIO TRANSFER

Sebastian Gabler, NOA Audio Solutions, Austria

1. Is recording safe?

Digital transmission and storage of information is widely assumed to function free of unwanted alteration and corruption. Barring an error message, we usually assume digital information to be complete and authentic.

Audio professionals however frequently report potential error conditions that often go unnotified by operators as they occur. Jargon words such as "drop-out", "glitch", or "click" were coined in that context. Often ambiguous, these terms refer to symptoms that may not be apprehended easily, but from all of which information loss, or information corruption, may be inferred.

As this affects equally the transfer of archival collections in the capturing of physical media to digital files, curators should be particularly sensitive with this phenomenon. From start to end of the transfer chain, established Quality Assurance efforts begin with controlling the conditions of legacy reproduction equipment, such as cleaning and alignment. Only the best Analogue-to Digital Converters (ADC) will be acceptable[92] in digitisation. The transmission paths of linear, digital signals are typically secured using Error Detection Code (EDC) and Error Correction code (ECC).[93] For digital documents, Fixity information such as cryptographic hashes can prove authenticity.[94]

The transition from continuous protocols to packets is a tipping point in Quality Management of digital information. When capturing the signal to a file, or when sending it through a network, the continuous signal stream has to be packetized. This is because these systems use packets, memory pages, or storage blocks. For the specific case of Hard Disc Recorders that are built on general-purpose operating systems (Linux, Windows, or Mac OS), interrupt handlers make it necessary to use comparatively large packets. Essentially, these interrupt handlers work on time slices. Specifically Second Level Interrupt Handlers scheduling kernel processes have become notorious under their Microsoft brand name *Deferred Procedure Calls*, or DPC[95], for triggering glitches when recording Video and Audio.

Notably, for most packetized domains of a transfer chain, standardised error detection methods do not exist so far. This is true for capturing to file and for some network transfer protocols based on User Datagram Protocol (UDP). Using ECC-protected Transfer Control Protocol (TCP) instead, an error would lead to re-transmission, which incurs latency that is often not acceptable due to service quality requirements. Thus, there is concern about potential corruption of the information passing through these components. Popular audio driver models such as ASIO[96] do not provide any flow control, error pointers, or detection codes. Further errors may be caused by shortcomings as trivial as protocol-specific handling of intermittent contact. Googling "firewire audio loose contact" returns more than 300,000 hits, potentially pointing to an issue too widespread to ignore.

Recently, FADGI, the Federal Agencies Digitization Guidelines Initiative, has released studies on the quality aspects of digital audio.[97] This includes a study on errors that may occur between the analogue to digital converter and the recorded file, titled "Performance Impairments

92 IASA TC-04, Second Edition, Chapter 2, Key Digital Principles.
93 AES3 CRC: See https://tech.ebu.ch/docs/tech/tech3250.pdf; SDI CRC and EDH, see: https://en.wikipedia.org/wiki/Serial_digital_interface#Line_counter_and_CRC.
94 http://blogs.loc.gov/digitalpreservation/2012/03/file-fixity-and-digital-preservation-storage-more-results-from-the-ndsa-storage-survey/.
95 Analyzing Interrupt and DPC Activity, see: https://technet.microsoft.com/en-us/library/cc938646.aspx.
96 https://www.steinberg.net/de/company/developer.html (see: "ASIO SDK").
97 http://digitizationguidelines.gov/audio-visual/documents/Interstitial_Error_Report_2013-04-08.pdf.

Caused by Interstitial Errors." The study includes a field survey leading to the conclusion that the matter of information corruption on computers is of substantial relevance. Fifty-six (56) out of 83 respondents have encountered such errors previously.[98] The occurrence rate of such errors has been stated with approximately once per 100 hours of recorded audio. This is an observation which is shared by the author. As on the other hand, audiovisual collections amount to hundreds of thousands of hours in large institutions, the issue should be of substantial importance for curators.

Against this background, the problem needs a solution that is both reliable and low-threshold for adoption and operation.

2. Error patterns

Taking a generic information technology (IT) approach, errors occur as single bit, or burst, errors. IT typically uses the bit error rate, or BER, which counts false bits per total bits transmitted. For Audio, the severity of a corruption can be judged by the audibility of a disturbance, which is not always in line with the amount of information corrupted. For instance, a single bit flip can cause a disturbance that is more obvious than hundreds or thousands of missing samples, depending on which position in the digital word it occurred.[99] Still, for the archivist, none of that is acceptable.

2.1 Missing information

Missing samples occur as the absence of a part of the signal on the time line. For instance, the picture below shows a 256 sample buffer missing from the signal. It will result in a click.

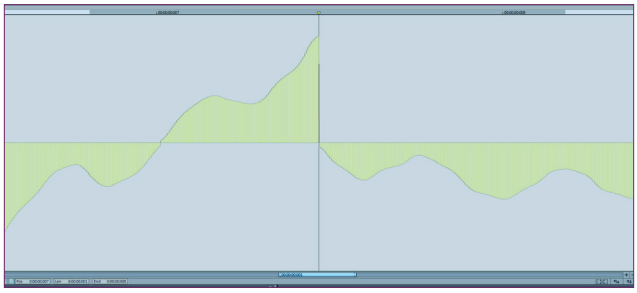

Figure 1 Missing audio buffer visualized

The author of the FADGI Study, Chris Lacinak, has coined the term "Interstitial Error" for this phenomenon, defining them with the statement, "Interstitial errors consist of lost or altered samples within the recorded file, resulting in the loss of content and integrity."[100] Occurrences have been reported as 1/100 h, or more. If caused by DPC queue overruns, they may occur as often as 1/1min.

98 http://digitizationguidelines.gov/audio-visual/documents/Interstitial_Error_Appen_2012-09-11.pdf, p. 58.
99 John Watkinson, The Art of Digital Audio, p. 17.
100 http://www.digitizationguidelines.gov/guidelines/digitize-audioperf.html.

2.2 Positional disorder

Packetizing may lead to audio samples in the wrong order. Practically, this will most likely occur with scattered blocks, instead of scattered samples.

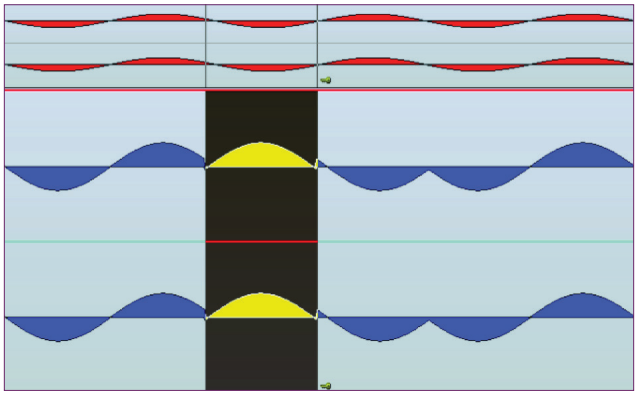

Figure 2: Swapped audio buffers

Figure 2 shows an identical signal recorded to two channels of the same interface. The lower channel exhibits audio 256 sample blocks A-D, in sequence of A-C-B-D. Note that there is no discontinuity in the time basis. The problem discovered by the author in 2008 was caused by a firmware bug in an audio interface (the blocks were already swapped when the signal hit any application), and the error would occur at least once in 10 h of continuous recording. Notably, the manufacturer acknowledged the issue, but refused to fix it. The solution was to downgrade the firmware, respectively the driver. One can imagine that with natural signals, the error may be rather subtle and hard to detect by established methods. When found, it can be completely reverted using a sample editor, however at a tremendous effort.

2.3 Corrupted information

Different error conditions may lead to wrong sample values. For example, a mute event may have occurred in the signal chain, issuing digital signal 0 instead of the actual signal values, or the signal path was affected by noise, leading to altered bits.

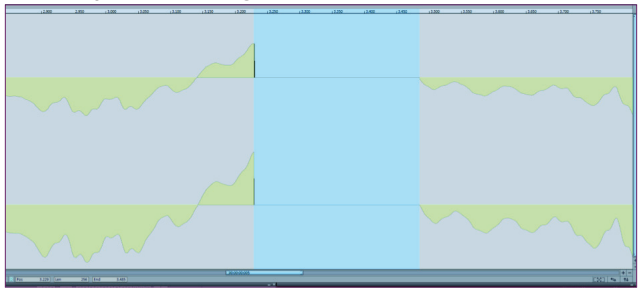

Figure 3: Mute Artifact.

Information could be corrupted by parasite signal processing that may have been introduced into the signal chain without the awareness of the system designer or operator.

While an archival signal chain will usually want to avoid the use of equipment that may alter the signal in the transcription process, such as analogue or digital mixers, specifically commodity components may have unexpected features. For instance, consumer sound cards may have a Sample Rate Converter (SRC) of unknown quality standard that will alter the signal.

On Microsoft platforms, following the implementation of the so-called *Universal Audio Architecture,*[101] legacy application programming interfaces (API) such as *DirectSound* and *Multimedia Extensions* (MME) were deprived of a direct access to the physical sound card. The API usually would still function, however the signal would be routed through the operating system's *Mixer*, detaching control over the actual sample rate and level of the signal. This may result in unwanted re-sampling, or peak limiting.

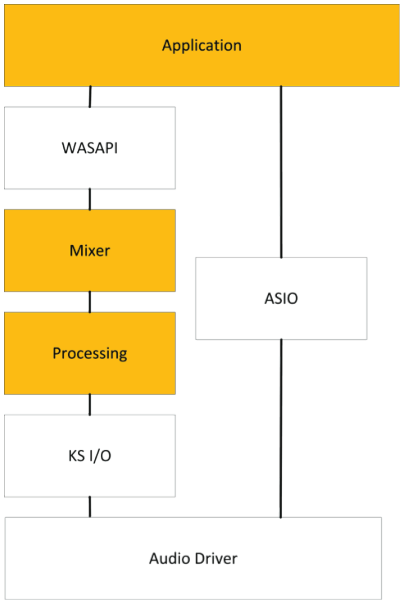

Figure 4: Microsoft Universal Audio Architecture.

The coloured elements in Figure 4 may introduce signal-altering processing.

3. Established quality assurance methods

As Quality Assurance is paramount in all archival transfer procedures, established methods to identify information corruption exist, and they should be evaluated in how successfully they can help with the subject.

3.1 Listening

Critical listening is probably the most accepted method for checking audio content. When done with 100% coverage, it does usually provide good quality assertion. On the other hand, the effort required presents a serious burden, specifically for high-throughput processes in industrialized environments. For distributed workflows, such overhead may be unacceptable. Specifically for interstitial errors, the method has two additional caveats that should not go unmentioned. First, listening fatigue which is the most critical parameter for this method may lead to error events going unnoticed, and this is specifically problematic with events that may only occur every few dozen hours. Fatigue can be regarded as a special case of reduced Quality Control (QC) coverage. Sample checking is an intentional way of reducing effort by reducing QC coverage. This however comes with a reduced detection rate, and thus will not lead to better efficiency. Moreover, listening can only identify an anomaly as such, but usually it cannot tell if it concerns a newly added error, or if the disturbance is pre-existent.

101 Microsoft, Universal Audio Architecture https://msdn.microsoft.com/en-us/library/windows/hardware/dn640534%28v=vs.85%29.aspx.

3.2 System sizing and optimisation

Over-sizing transmission paths and recording devices to prevent performance-related errors is a widespread approach. This is owed to the fact that Ethernet and computers have become commodity goods.

Audio engineers are cautious when it comes to the choice of their digital audio workstation (DAW). DAWs based on general purpose processors and operating systems have become prevalent since about 1995 when personal computers became powerful enough to crunch two channels of standard resolution audio and record them to disk. The invention of the DAW has since revolutionized audio technology.

Since 1995, single-thread processor performance has increased at an annual rate of 52% until 2004, and at an annual rate of 21% for the last decade. Today, this has resulted in a 128-fold increase[102] (for Integer instructions). Moreover, current commodity processors offer simultaneous multithreading at 4 threads, or more. In the same period, the bandwidth per information channel increased by merely six times (6x) when going from 16 to 24 bits (1.5x), increasing the sample rate to just below 200 kHz (4x). With these figures in mind, it is hard to imagine that modern computers still may cause problems when recording audio.

When looking at the above mentioned phenomenon of excessive Second-level Interrupt Handler rates, these unfortunately do not scale with performance, but may remain at the same values. It has to be observed that excessive DPC queues are regarded as an abnormal operating condition, and the most common source is a software bug, or an interoperability problem.

Figure 5: DPC Latency Analysis.

At the same time, optimisation guides exist both for MAC [103] and PC [104] which all basically aim at reducing the count of concurrent processes that may consume processor time. Some of the measures, such as deactivating serial interfaces or network controllers, may affect usability. For network transfers, similar concepts exist. A 1000 Mbit/s network link has a net capacity of approximately 100 MB/s in both directions. This is equivalent to streaming around 700 channels of 24bit audio at 48 kHz sample rate. Current commercial Audio over IP (AoIP) solutions use that bandwidth at typically 10% (64 channels).[105]

Over all, platform sizing and optimisation are absolutely relevant when it comes to production of digital audio files. The best error detection method will not prevent errors on undersized or flawed equipment; it will only provide evidence that something is not right. On the other hand, an oversized platform does not guarantee results that are free from errors. Only specific error detection methods can provide this information.

102 http://preshing.com/20120208/a-look-back-at-single-threaded-cpu-performance/.
103 http://us.focusrite.com/answerbase/optimising-your-mac-for-audio.
104 https://www.audinate.com/faq/how-can-i-tune-windows-pc-best-audio-performance.
105 https://www.audinate.com/products/manufacturer-products/dante-brooklyn-ii.

3.3 Signal analysis

Signal analysis is based on pattern recognition, raising a notice on the occurrence of the error pattern in the signal. It typically uses detection of "not natural" signal forms, such as clicks, or periods of silence, or spectral anomalies such as exaggerated high-frequency content. As long as the error keeps the pattern, these methods provide good detection rates. Unexpected error patterns will however not be detected. For detection patterns that are too generic, a high rate of false positive error pointers will be generated. Pattern detection in principle is not able to discriminate errors that have been added on transfer from those that already existed in the original. Other drawbacks are that results usually still require some manual interpretation, and that the analysis and validation are resource-costly.

3.4 Flow control

Several digital audio workstations offer error pointers to buffer underruns and overflows. Within the domain of the DAW, this method provides good protection. It does not help with signal alterations caused by software bugs, bit-rot, or parasite processing.

3.5 Process redundancy

A system recently suggested in the FADGI study[106] involves the use of a second recording system, in parallel to the actual production DAW. In short, an additional capture system, typically a standalone recorder, is defined as a reference platform that will not cause any (or at least not the same) errors during transfer. The method requires parallel systems that are sufficiently different. After the completion of the capture, both recordings will be compared, using signal processing and/or cryptographic methods, asserting the production recording being accurate as long as no difference between them can be detected. This method has several advantages above aforementioned approaches, as it is specific to errors introduced during the transfer process. However, it introduces considerable acquisition and operation overhead, which seems to reflect in the low adoption rate so far reported by the study author in personal conversation with me.

4. A new approach

Ideally, protection against information corruption in archival transfers would be non-intrusive, reliable, and affordable. As the error patterns are random and intermittent, constant monitoring is required to catch the events reliably. The domain of interest starts with the Analog to Digital Converter (ADC), and ends with the written file.

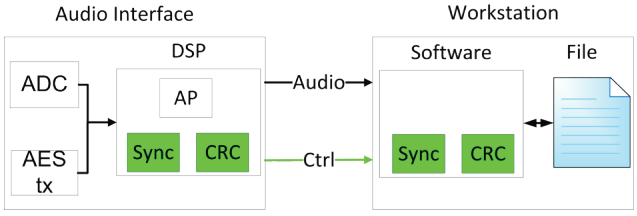

Figure 6: NOA BitProof™ block diagram.

One possibility to add Quality Assurance (QA) to the transfer chain is to use Error Detection Codes that transmit as metadata of the audio stream in real time. As already mentioned, standardised protocols for that purpose do not exist. As an implementation cuts through several functional units and protocol layers (i.e., audio interface, streaming transport, block transport, operating system APIs, application layer, file system, and block storage), vendor-independent

106 http://digitizationguidelines.gov/audio-visual/documents/Interstitial_Error_Appen_2012-09-11.pdf.

standardisation of such a process is complex, probably outstripping actual implementation by far. For NOA, as a specialized vendor in a niche market, a proprietary implementation is quite obvious. Since NOA is a manufacturer of audio transfer hardware, as well as software, we have access to the required components.

The required building blocks for implementation are available technology. As most current audio interfaces, NOA's products feature a Digital Signal Processor (DSP), and there is audio recording software, running on a general-purpose operating system. Both components are suited to crunch Error Detection Codes (EDC) on the signals passing through them, and do the same for the information written to disk, i.e., cyclic redundancy code (CRC). Additionally, NOA's interfaces provide a remote control interface between converter and the recording software that is otherwise used to control the ADC, log QA data from the ADC, and to control attached re-players. This remote control interface may be used to transmit the EDC information simultaneously to the audio signal. As the AES-3 audio interface does not allow to transmit custom data (at least not as required for the purpose), this helps to overcome an otherwise substantial impediment. At the same time, this approach allows using any audio transport path with the same safeguard, including AoIP solutions like AES-67, Dante, or AVB.

The actual working principle includes the following steps:

1. Upon record start, the host software requests a positional reference in the audio stream from the audio interface ('Sync').
2. The audio interface periodically transmits EDC to the host software.
3. The host software obtains the same information from the recorded file and compares it with the reference information from the audio interface.
4. If a mismatch is detected, an error message is provided to the operator. If the EDC codes on sender and receiver match, normal operating conditions are secured, and no further information needs to be provided.

The error patterns relevant for detection include missing information, altered information, corrupted information, and positional disorder. For these errors, any error detection code with positional weighting is suited, including the most basic CRC polynomials. Parity codes are equally useful for all error patterns, with the exception of positional changes (swapped buffers, see section 2.2), as parity does not provide positional information.

The most relevant benefits of this method are that it is immediately available, it is reliable, and it does not incur any operational overhead. In fact, the user will not recognize that any safeguarding is in place unless an error occurs.

5. Conclusions

Securing information integrity is among the most important targets in preservation systems. The proposed method addresses notorious error sources of conventional audio digitisation environments and significantly increases Quality Assurance standards. Error Detection Codes securing the entire transfer chain from the audio interface to the recorded file cause no overhead for the operator. Therefore, the same quality level is provided for small institutions, as for major archives, libraries, or broadcasters. It fits into occasional on-demand digitisation approaches in the same way as for systematic format migrations. Specifically, service providers may want to include this kind of safeguarding, creating added value for the services to their archival customers. Smaller archives that run their own on-demand digitisation activities profit from a consistent Quality Assurance, even if the equipment is only in use from time to time.

In an environment using Fixity to secure authenticity in the creation of Archive Information Packages (AIP),[107] the trusted domain can be extended towards the actual audio interface. Therefore, such Quality Assurance methods have specific relevance in Open Archive Information Management (OAIS) environments.

The method aims strictly at Quality Assurance, not at correcting errors. It helps with controlling the quality limitations imposed by the runtime environment of a preservation system; however it does not solve these limitations. For instance, a computer or a network losing information every minute will still be in the way of an efficient process. Typical commodity equipment that is working reliably, however, can be utilised with no problems, and the requirement to use specific components, like ECC memory, or very expensive audio interfaces is relaxed considerably.

107 OAIS Reference Model, CCSDS 650.0-M-2 http://public.ccsds.org/publications/archive/650x0m2.pdf.