Look Ma, No Hardware!

Juan M. Guadiana

Project Engineer, New Mexico Technology Group
White Sands Missile Range, NM 88002-0398
APPROVED FOR PUBLIC RELEASE, DISTRIBUTION UNLIMITED.
OPSEC Review Conducted on June 23, 2009

Abstract

Google *Soft Decom* and the number of hits will be tenfold over the same search last year. The migration of hardware functionality toward software is relentless. On the telemetry front, Data Bridges that take Pulse Code Modulated (PCM) signals and transform them to ubiquitous network packets make it all too easy. The need for expensive hardware such as the Decommutator (Decom), Frame Synchronizer, Digital Recorder, and Oscillograph Recorder (StripChart) will diminish sharply. Software Decom packages will feel the squeeze too, from homegrown Soft Decom software that is easier to maintain and has no licensing issues.

This paper airs the dirty laundry associated with this hardware and software. Latencies and ugly temporal aberration that really plague an analyst. Also discussed is how a few packet/file formats eliminate the need for most of the hardware in a traditional telemetry data processing facility.

Key Words

Data Bridge, Integrated Decom Display, Soft Decom, Virtual Display

Introduction

There is little doubt, about the coming demise of the hardware decom. Digital (Chapter 10) Recorders, although recent, are next and the StripChart is not far behind. Functionality that can be achieved with the Data Bridge and modern computers offer compelling cost and performance advantages over existing hardware. What is a Data Bridge? A Data Bridge acquires the serial PCM and transforms it into network packets, a computer friendly form. The packet navigates the network to the destination and is converted back to PCM by another Data Bridge. You could also call this a multiplexer-demultiplexer. Writing this packet to the disk drive of a quick computer accomplishes minimally what a Chapter 10 recorder does. The packet is the key to streamlining the architecture.

It was serendipity that all of the pieces were in place at the integrated Telemetry Data System (iTDS), White Sands Missile Range. Data Bridge functionality already existed in the Correlating Source Selectors (CSS), the hardware front end of the iTDS. Desktop computers with vector processing technology easily handled the high speed data in “real time”. The computer’s capacity was so robust that the computer was pressed into service as the Integrated Decom Display (iDD). As the name implies, the iDD decoms the data while rendering the desired display (embedded uncompressed video, in this case) in “Real Time”. This development led to unexpected results and what began as a temporary solution, became the architecture and heart of the iTDS. Related Technical papers are later containing greater detail.

Mo' pain, no gain!

Let us take a look at what was happening. Data normally acquired at several field sites is relayed to a Data Center. Transformed into packets (DS3 normally) for efficient relay to their destination, where packets are applied to a demultiplexer to regenerate the PCM. The PCM is then applied to traditional hardware, like decoms, recorder, Matrix Switch, TTL driver and many coaxial cables.

PCM must be recorded at the acquisition site or after the demultiplexer. Digital recorders convert PCM to Chapter 10 packets, and write them to the recorder’s disk drive. If your software will not read these packets, you must reproduce and decom the PCM. Finally, Chapter 10 files are written to DVD for customers, if the DVD can hold the file. Otherwise, the customer gets the recorder's drive. That often does not work because the drives usually are RAID drives and exchanging RAID drives is a compatibility nightmare. It is better to copy the file over to an external non-raid disk. The problems may be insurmountable if your customer does not use the same recorder manufacturer. Even when same manufacturer’s recorder is used, firmware versions may differ and you just can’t get there. What is wrong with this picture?

Stop the Insanity!

Why regenerate PCM? This is tragic, because the customer never wanted the Chapter 10 files to begin with! The customer really wants a raw binary file. The file that contains each minor frame received with a time tag appended. The file was considered a "Post Test" data product only, and at one time it was, provided days or weeks later. This file, intended for the customer’s data reduction program uses the customer supplied record format. Woah! Doesn’t this record look like the Data Bridge packet? Yes it does.

Again, the packet is read from the Data Bridge, a record is generated with the packet contents and the record is written to the drive. Not constrained to use those nasty Chapter 10 packets, the same data format may be used that your customer requested to begin with. Binary files in realtime, Wow! This must be too good to be true, let's go over this once more. Normally the PCM is recorded on a Chapter 10 recorder. Then reproduce the Chapter 10 signal into a decom or Chapter 10 savvy software that converts the packets to the data format you wanted to begin with. Why not use the customer's record definition to begin with, bypass the Chapter 10 recorder, and lose nothing.

Best Intentions

Chapter 10's promise is that the data from these recorders would be interoperable. This has not been the case, but binary files are indeed interoperable. Range users customarily take binary files, as part of their post test data reduction process. These files are processed by almost every computer known to man, and deliver better effective interoperability. With a few file standards, the process becomes transparent. Good bye, Chapter 10.

Not so fast, you say? Chapter 10 does reference the bit stream to time very well when either recording or reproducing. This would really matter if the vehicle's telemeter was synchronized to an onboard GPS frequency reference. The telemetry stream becomes a time code stream as well. If the telemeter did that, there would not be need for any ground station time tag. It turns out this does not matter at the data center. PCM is subject to digital relay effects, the multiplex delays are orders of magnitude greater than the time accuracy Chapter 10 recorders can deliver. Chapter 10 recorders have the most utility at acquisition sites, but the lack of interoperability will give even that application the kiss of death. This may sound blasphemous, but the Chapter 10 window is closing.

Back to the Future

Demultiplexer latencies or temporal characteristics were briefly mentioned. Adverse temporal effects are inherent any time buffers are used in any "real time" implementation. Temporal degradations are the "Dark Side" of a digital architecture. At White Sands, field sites were first to implement digital technology, replacing analog frequency domain Microwave and Fiber multiplexes with the nifty new DS3 technology. This provided a big boost to reliability and bandwidth. The Data Center, however, was not able to make this "jump" to Hyperspace. Collaborations with Tam Tram (PMRF), Gene Law (PMR), industry partners NetAcquire, Apogee Labs, and others helped the iTDS make the jump to a digital architecture. Temporal characteristics of digital signaling forced bit synchronizers and most digital recorders to be removed from the "Real Time" environment. Indeed, Chapter 10 did provide "breathing room" here.

Conservation of Bits

The following figures depict what happens temporally, when using a buffer such as is used in a demultiplexer. Bit rate must be "dithered" to prevent a buffer from over or underflow. This "dithering" causes the latency to take on the Ramp characteristic shown in Figure 1. The data processed is actually moving back and forth in relative time, in response to the changing bit rate. All buffers are subject to this "Conservation of Bits" law, otherwise the buffer will over or under flow, and the bits end up all over the floor. There is no way around this, but there are ways to ease the pain. The Correlating Source Selector's buffers provide rate adaptation very adeptly. Latency is minimized, but at the cost of bit rate stability. There is no free lunch, most PCM equipment is designed to be very stable, clock rate wise, suitable for use with bit synchronizers, transmitters etc. Subjected to excessive temporal variations, these units lose data. Buffers may be made larger, but that increases latencies. Fortunately, when these devices underflow, the clock signal stops and data is not lost. By setting the buffer's bit rate above the highest anticipated data rate, the device function is improves. In this "Starved" mode buffers underflow periodically but never (in theory) overflow.

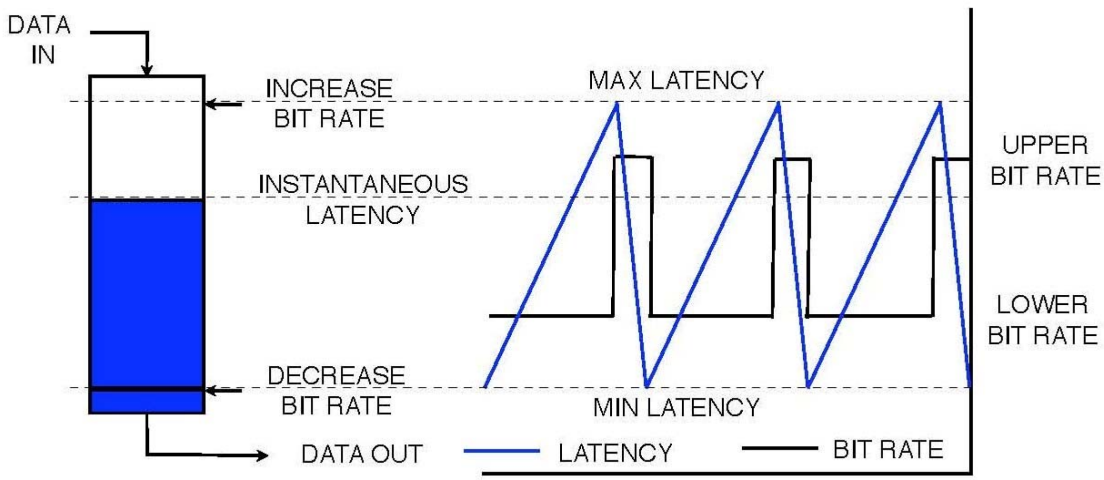

Figure 1 Typical Buffer Characteristics

This problem manifests anytime bit rates are constrained. For example, the bit rate clamps were set to +/- 1% on the CSS equipment to keep the data rate from varying outside the digital recorder bit rate tolerances. Variances during the testing should have been small and were. This was not the case with latencies. In this case, the actual bit

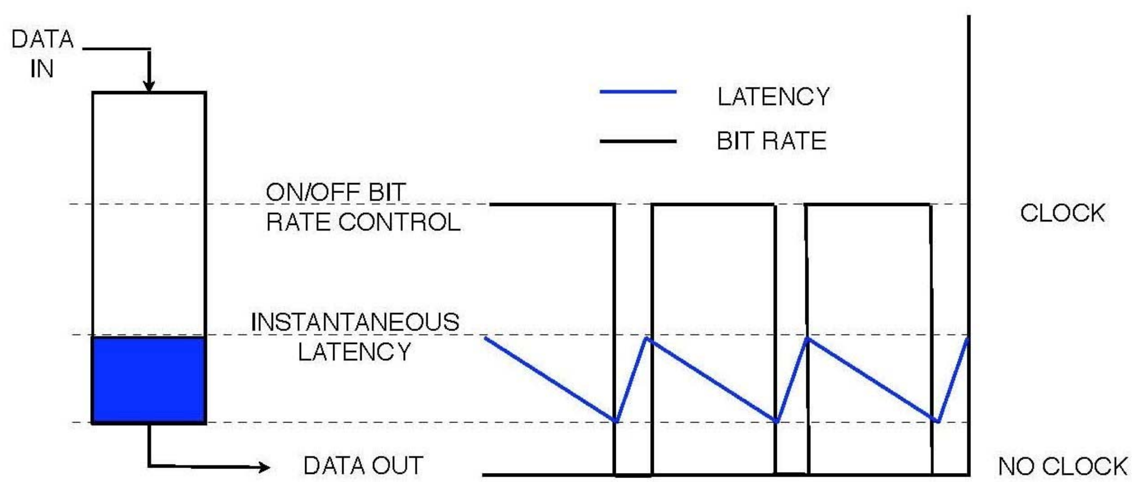

Figure 2 Starving the Buffer

rate was a modest 1% above its assigned rate. Pre-mission checks were over 10minutes in duration and the buffer (First in/First out) accumulated over three seconds worth of data, causing telemetry displays to run over three seconds late. The Safety Officer (SO) in charge of the mission had been allocated a 5 second budget to make flight termination decisions. With only a two second effective budget, the mission fell into jeopardy and would have ended in a missile termination but other data sources (Radar and Optics) had solid acquisition allowing the mission to continue. If data enters a buffer at a rate faster than it leaves, the buffer will fill (latency increases) until the buffer overflows. The "Starved" buffer trick will help rate constrained devices such as the Metrum VHS digital recorder, MARS and other like equipment.

Paradigms

Programmers too, suffer paradigm paralyses. The biggest sin a programmer can commit is to hard code. Many "short cuts" were taken when writing the first mini Soft Decom. Hardcoded for the application it was designed for, it took no longer to write the code (from scratch) for this Decom, than it did to write the database flat file for an existing hardware Decom that drove the Stripcharts. You always learn more when doing something you are not supposed to do. It took a while to climb out of the paradigm, the code for the Decom itself was but a few lines, that takes frame aligned data, defruits, scales, and realigns it for its destination. A program that is only a page or two in length is going to be easy to maintain.

Consider this; Professional Software Decom packages are large. 1), the nature of a generic decom, is that it must contain many methods to handle most permutations of the data stream. 2), the Decom is tasked to process every parameter (hundreds or thousands of parameters) in that stream and, at times, several streams. This makes for very large complicated software packages. What if, Soft Decoms need not do everything? Suppose each display were to employ its own Decom? The Decom need only handle as few parameters as are needed on that display, only the methods that apply. Any time a display is added, a Decom is added. Truly self-scaling, distributed processing with no single point failure mode, this is the iDD. A massive task is divided up into smaller much simpler tasks. Remember top down design and stepwise refinement? That paradigm is not going away.

Reusable Code, Really

Writing generic, one size fits all software decom packages are large tasks and expensive. They do bring a lot to the table, data mining for example, yet often most of the package is not well utilized. The code must contain rarely used methods or techniques that normally lay dormant. Though dormant, they do add to the bulk and complexity. Maintaining this type of software is no small feat and is costly. Why not strip away all of that unused code? Auto code generating applications have been written that build a small software Decom tailored to the application. The itty bitty Soft Decom is easy to maintain, but auto code writing applications themselves are not. So, write your own itty-bitty Soft Decom from scratch and do not add any code or any hooks you will not be using immediately.

Now, let's violate this new paradigm and generalize parts of the small decom where it makes sense. Generalized parts enable a plug in functionality, so you add only what is needed. A community source code library of plug-ins could be maintained by a Range Commander Council subcommittee.

Binary File/Packet Standards

Range customers have used binary files as their post test data product for quite some time. These formats can be all but identical to network packet formats. Shown in Figure 3, the first format defines records as minor frames with an appended time tag. Most telemeters use a constant word length but are not required to. The first format example does not. The second format is slightly different in that atypical length minor frame words (i.e. 7, 9, 10, 12 bit) are placed in words of normal length (i.e. 8, 16, 32 or 64 bit). The third file format contains a header for each record and a fixed length Documentary block. The Documentary block may carry Display Definition Tables, other documentary data or an application as well. Segmented variants handle lengthy frames, while maintaining reasonable packet lengths.

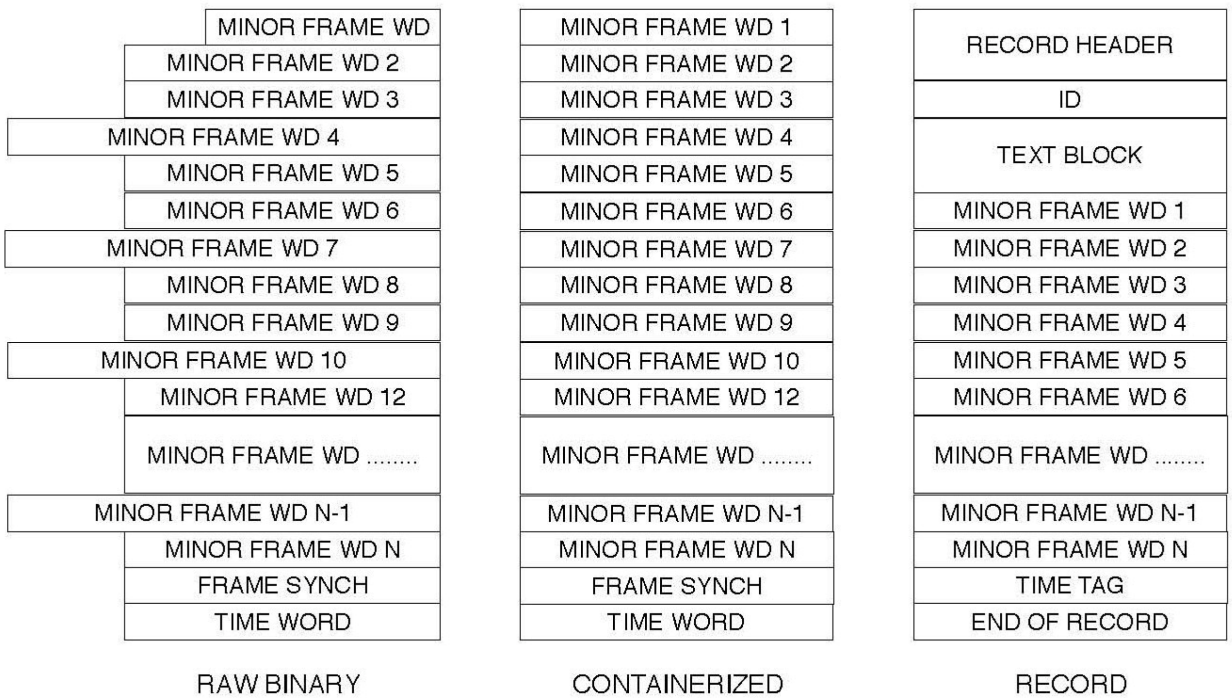

Figure 3. Proposed File/Packet Models

Packetizing need occur only once, right at the acquisition site. Data may or may not be frame aligned but time is embedded. If time is not embedded in the data stream, how is it used? For example, a mission requires five acquisition sites, results in five data and five IRIG B time code streams. Chapter 10 recorders are not going to process five time code streams. After best source processing, one composite stream remains, but that stream's temporal history is not consistent with any of the IRIG B time code streams. Time must be embedded in the packets at the source.

The last format proposed enables the minimal architecture migration at White Sands. The integrated Decom/Displays (iDD) are diskless, booting from servers in the control room. The iDD application reads the Documentary Block from the data stream then configures the iDD automatically to render the display. Overall security is improved, only the server's disk drives require protection as the iDD is unclassified once turned off.

TIME OUT!

As DS3 networks are replaced by IP networks the temporal degradations will really be aggravated. The temporal characteristics of IP Networks will make you yearn for DS 3. Packets will arrive in a nondeterministic manner, potentially out of order. Larger buffers will be needed, and the "Conservation of Bits" buffer law will strike again, exacting larger latencies. Source Selectors and decoms will struggle with out of order packets, acquisition site time tags will greatly simply the problem. Acquisition site recordings are preferred today because they generally offer the best time tags. By tagging at the source, the time tags at the Data Center are every bit as good, because they are in fact the acquisition site time tag**.** Correlating the time tags from multiple streams, should yield better composite time tag. Acquisition site recording, now becomes a nice to have, redundant recordings needed only when the Data Center recordings or networks fail.

Once packet/file standards are used, the need for Chapter 10 evaporates. Range customers can immediately use their data. No waiting in line at the TM facility to get Chapter 10 files converted to a useful format. Chapter 10 hardware is expensive hardware and expensive to maintain. Packetizing and time tagging at the source just keeps paying off. Just like the Ground Hog Day movie, but at the Lotto.

End of End of Paper

The warning stripe at the end of a roll of chart paper has become prophetic. Sadly, the end is near for the Stripchart. Safety Officer’s are the most demanding of all StripChart users. During missions, their focus on the real time display part of the StripChart . As events disappear from the real time display, they would review the paper recordings, during and after the mission. The thermal paper recording is not real time. Despite this, and other drawbacks of the velocity driven thermal recorder, the real time virtual display has not quite replaced this technology. Safety Officer's were invited observe at an iDD prototype at the iTDS. The chart ran left to right (not vertically) and the entire flight test could be displayed on the 30-inch high resolution monitor a long with an attitude model. There was no need to look anywhere else. As in every iDD morphism, decommutation is performed onboard the same computer rendering the display.

Conclusion

The writing is certainly on the wall for some magnificently designed hardware and software that has served everyone well. The ink is not dry yet, users will hang on to what they have become accustomed to, so long as they can. There is a new opportunity to ignite everyone’s creativity again, and the intellectual property is yours! For any questions or additional information, please feel free to contact me anytime at Juan.M.Guadiana@US.Army.MIL.

Related Papers

Software Decommutation and Integration,	ITC Proceedings, October 2005
Best Source Selection on Encrypted Data,	ITC Proceedings, October 2006
The SoftDecom Engine,	ITC Proceedings, October 2006

Acknowledgments

My colleagues at White Sands Missile Range; like Jesus Benitez, who can "jump" tall paradigms in a single bound and at the New Mexico Technology Group (NewTec), like Gene Dirk, who asks "Why not?"

Our industry partners, the little companies that could!