System-Level MIDI Performance Testing

James Wright
Computer Music Center
IBM T. J. Watson Research Center
Yorktown Heights, NY 10598 USA
<jwright@watson.ibm.com>

Eli Brandt
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213 USA
<eli@cs.cmu.edu>

Abstract

We describe a new approach for testing MIDI performance characteristics, which is both inexpensive and highly accurate. Following a description of the test method, actual results for a number of MIDI interface technologies are presented. These results show that jitter and latency performance of newer (USB-based) interfaces is actually two to three times worse than older MIDI interface technologies. Other potential applications of this technique are briefly discussed. Finally, we consider implications for the future, since the recent PC2001 Guidelines and WHQL (Windows Hardware Quality Labs) logo requirements will forbid the use of the older MIDI interface technologies in new personal computer designs.

Keywords: MIDI; jitter; latency; testing; standards

1. Motivation

For musical applications using MIDI, the performance of the MIDI communication links is often critical. It is well known that inadequate bandwidth limits the complexity and expressiveness of the music that can be conveyed over MIDI. However, the latency and jitter characteristics of a MIDI connection can have even more profound effects. Excessive latency can make a system unusable for interactive performance of any kind, while excessive jitter can compromise or destroy the rhythmic integrity of the musical experience. (We define latency as the average (mean) end-to-end transit time for a single message; jitter is the deviation between the intended and actual time intervals (delta times) between two events over a given transport, or the amount of variation in latency).

New transports such as IEEE-1394, USB, Ethernet and wireless telephony are now being used to transport MIDI. While these new transports offer many benefits, poorly-designed protocols for carrying MIDI over such transports—and complex, non-deterministic operating system drivers—can easily impair the quality-of-service issues critical for MIDI.

This test provides an easy way to assess overall MIDI quality-of-service characteristics on the system level. It is also helpful when diagnosing system configuration problems (e.g. pathological interactions between several MIDI and/or audio drivers).

2. Perceptual Criteria

Moore (1998) argues convincingly that time intervals on the close order of 1.5 milliseconds are both audibly significant and controllable by human performers in common musical situations (e.g. grace notes, flams, strummed chords). A number of perceptual studies have shown that for streams of individual audio events, timing jitter on the close order of one millisecond can be audible, particularly in the context of rhythmically complex and syncopated ensemble music. (Bilmes et al, Iyer, Lunney, Michon, Schloss)

The threshold for tolerable latency depends on the specific application. Passive listening applications such as. streaming Internet audio can tolerate substantial latency. Music composition and interactive performance applications require very low latency and jitter. For such applications, we recommend target system bounds of 10 msec. latency and +/- 1 msec. jitter (lower bounds would be musically valuable). Note that these are *system* level bounds. Since systems generally include at least three components — the MIDI event source, the MIDI transport and the MIDI sound generator — the MIDI transport component should exhibit latency and jitter that is significantly lower than the system bounds (ideally one third or less of these bounds).

3. Test Method

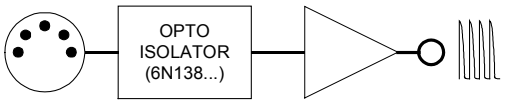


Figure 1.

The test method involves recording MIDI as digital audio and analyzing the recorded waveform to extract timing characteristics. We record the actual signal from the MIDI-IN circuitry as shown in Figure 1, tapping the opto-isolated signal through a resistor to provide minimal buffering. This is effective because MIDI is essentially an asymmetrical pulse train with a clock frequency of about 31KHz, well under the typical 44.1KHz digital audio sample rate.

The basic approach is similar to an earlier method described by Freed (1997), but differs in two respects:

- The MIDI digital pulse stream is not "down-sampled" by using pulse stretching circuitry, but recorded directly.
- A differential technique is used, which compensates for possible timing irregularities in the reference MIDI pulse stream.

The actual MIDI-Wave transcoder (Figure 2) has two MIDI-to-audio circuits and is quite inexpensive (ours was built from a MIDI-Thru box, two resistors and a couple of audio jacks.)

As shown in Figure 3, two distinct MIDI audio streams are recorded during each test: the REF events fed to the device under test, and the TEST events output by that device.) The resulting stereo audio file is then analyzed to determine latency and jitter. Envelope tracking is used to identify pulses in each channel; cross-correlation is used to match each left channel (reference output) pulse with the corresponding right channel (test output) pulse. By differentially comparing the recorded left channel audio (reference source) to the recorded right channel audio (system under test), we measure the timing errors introduced by the system under test, while canceling out any timing irregularities present in the reference pulse stream. The interval between left and right channel pulses corresponds to latency (mean event transit delay), while the variation in that interval over time corresponds to jitter (variation in transit delay).

Since left and right audio channels in a stereo audio stream are phase coherent, timing skew between the two channels is less than one sample interval (i.e. less than 23 microseconds at a 44.1K sample rate). This is well below the target measurement accuracy of 100 to 200 microseconds.

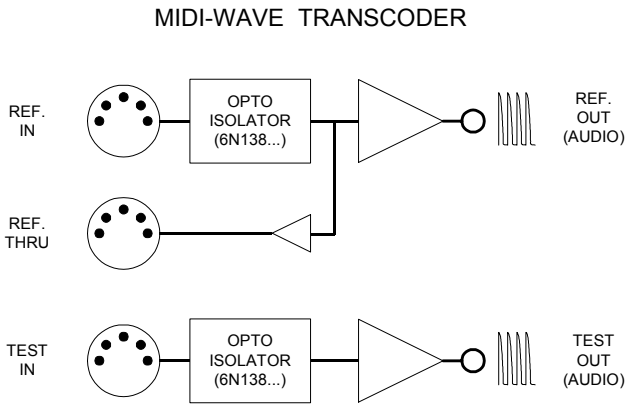


Figure 2.

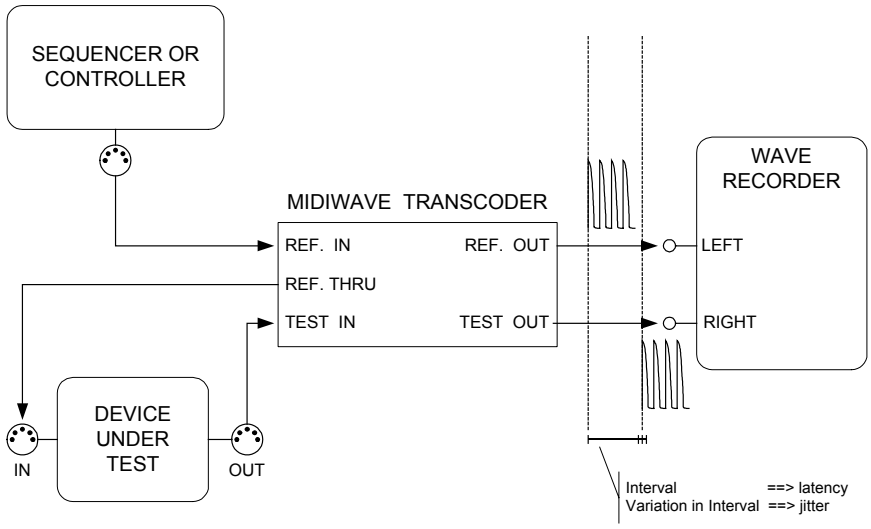


Figure 3. Complete Test Setup

4. Test Setup

Five different types of MIDI interfaces were tested to determine overall MIDI system performance on three different Windows 98SE systems. Round-trip performance (MIDI IN to host system to MIDI Out) was assessed in terms of latency (mean event transit delay) and jitter (variation in transit delay). The interfaces tested were: Creative Labs SBLive! (PCI), Roland SCP-55 (PCMCIA), Roland SMPU-64 (USB), Roland UA-100 (USB) and Crystal Semiconductor CS401 (motherboard/PCI).

Tests were conducted on three systems:

- "TPad" — IBM Thinkpad 770ED (277 MHz Pentium II, 160M RAM, 8G EIDE hard drive)
- "600x" — IBM Thinkpad 600x (500 MHz Pentium III, 192M RAM, 12G EIDE hard drive)
- "MPro" — IBM Intellistation M Pro 6889-14U, dual Pentium II/400MHz, 320M RAM, EIDE and Fast/Wide SCSI.

Each test system was running Windows 98 Second Edition with all available hotfixes. A separate system was used both to generate the reference MIDI event stream and to capture the resulting stereo audio stream. The Steinberg Cubase VST/24 sequencer (v3.7 R1) was used to provide a high-performance MIDI Thru connection on each test system. This software was benchmarked against the Win32 API midiConnect() MIDI Thru facility, and was found to provide signifi-cantly better performance (lower latency, better jitter) for routing incoming MIDI events to a MIDI Out port.

Three test runs were run on each combination of PC system and MIDI interface, with test results combined for subsequent analysis. Schedulers, anti-virus software, network drivers and similar software were disabled; a clean reboot was done when changing the MIDI interface under test. Each test run consisted of 288 pairs of REF and TEST events (yielding 864 event pairs for each tested

interface/system combination). Nominal REF event spacing was 4 msec (about 25% of MIDI 1.0 DIN capacity) to facilitate analysis. The REF event stream is composed of alternating PitchBend and Control Change events in order to produce a consistent stream of 3-byte events with no running status.

5. Results

Figure 4 gives the results for each combination. Three performance clusters were apparent on each of the systems tested (Figure 5) .

The best performers were clearly the non-USB "legacy" MIDI Interfaces (SCP-55, CS-401 and SBLive). The Roland SCP-55 was the best performer by a slight margin. Within this cluster, latency was 2.5–3.3 msec, peak jitter was 3.1–3.6 msec, and the standard deviation for transit delay was 0.6–0.8 msec.

The Roland SMPU-64 was a mid-range performer. Latency was about 2.6 times greater and jitter was about 2.1 times greater, relative to non-USB devices.

The Roland UA-100 exhibited the worst performance. Latency was about 3.7 times greater and jitter was about 2.6 times greater, compared to the non-USB devices (results for UA-100 / 770ED are even worse).

	ThinkPad 770ED (PII, 277 MHz)				Intellistation M Pro (400MHz)			TP600x (500MHz)	
	SCP55 (TPad)	CS401 (TPad)	SMPU64 (TPad)	UA100 (TPad)	SBLive (MPro)	SMPU64 (MPro)	UA100 (MPro)	SMPU64 (600x)	UA100 (600x)
Latency (Mean Delay):	2.5	3.2	7.7	12.4	2.8	7.1	10.8	7.8	8.9
Peak Jitter (Max-Min)	3.1	3.7	8.5	10.5	3.6	6.0	9.1	7.8	7.5
Std. Dev:	0.78	0.66	1.4	2.4	0.67	0.93	1.8	1.2	1.4
Min Delay:	1.1	1.3	4.3	7.4	1.1	4.5	6.8	4.7	5.5
Max Delay:	4.2	4.9	12.8	17.9	4.7	10.4	15.9	12.6	13.0
Median Delay:	2.6	3.2	7.6	11.8	2.8	6.9	10.9	7.7	8.7

Figure 4. (All units in milliseconds)

It is interesting that the SMPU-64 exhibits significantly better MIDI performance than the UA-100, given that both devices use similar USB-IF MIDI protocols. (UA-100 performance is roughly 64% worse; jitter performance on the Thinkpad system was 80% worse.) This performance degradation appears to be due to the presence of active USB audio streams. The UA-100 also supports three stereo USB audio streams (16 bit, 44.1KHz), which are apparently always active even when the corresponding audio ports are closed. This represents a constant additional USB load of roughly 570 bytes/frame, equal to roughly 43% of typical USB bandwidth (Garney 1996), assuming 1308 non-overhead bytes/frame given typical USB overheads; actual overhead factors can vary significantly). Since USB audio transfers are isochronous and USB MIDI transfers are bulk (asynchronous), it appears that the additional load imposed by three stereo audio streams significantly impacts the ability of a USB system (drivers and transport) to deliver MIDI in a timely and consistent manner.

It is likely that enhancements to the host USB drivers could mitigate this effect to some extent. However, no drivers can prevent isochronous traffic (audio) from taking priority over asynchronous traffic (MIDI). Furthermore, no tests have yet been performed in the presence of additional active USB loads. The 64% performance degradation apparently caused by the presence of three USB audio streams suggests that additional USB traffic (e.g. DSL modem, Ethernet NIC, scanner, external storage device) will cause additional MIDI performance degradation.

A number of USB interface manufacturers (e.g. Mark of the Unicorn, Steinberg, EMagic) have developed proprietary USB MIDI protocols which claim to reduce jitter substantially through the addition of timestamps to the USB MIDI data stream. (The Roland devices comply with the standard USB-IF protocol, which does not use timestamps). Latency for these devices is not specified. (We hope to test such devices in future.)

6. Other Applications

The MIDI-Wave test approach is also useful for diagnosing system configuration problems. During the course of testing, systems with mismatched audio and/or MIDI drivers exhibited poor performance or even pathological behavior. Typically, MIDI performance was

impaired (in one case, an 80 msec burst of incoming MIDI data was retransmitted in fits and starts over a period of 400 msec.) In some cases, serious audio glitching occurred when dense bursts of incoming MIDI apparently prevented the audio driver from servicing interrupts. Note that the system as a whole was not overburdened: once properly reconfigured, performance was good.

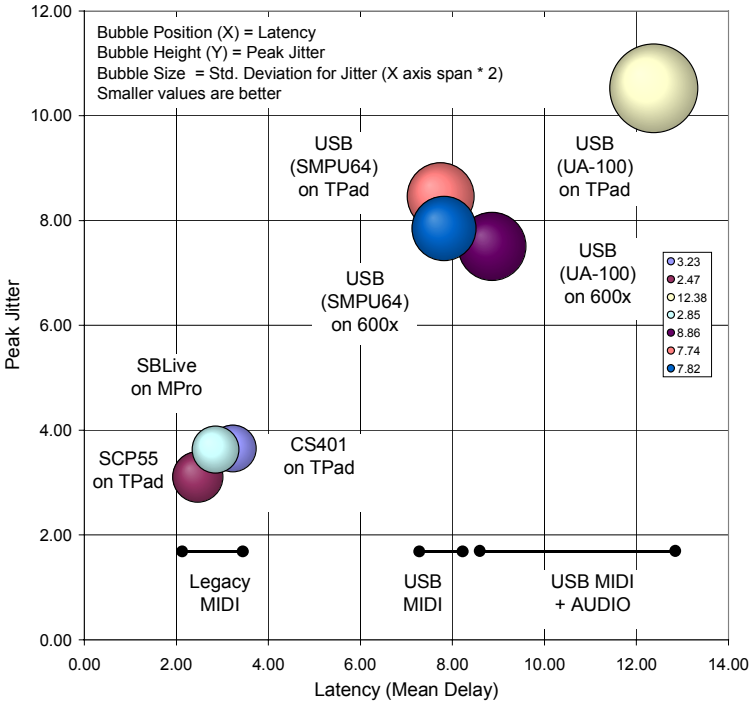
These problems were clearly shown by the captured audio data. Visual inspection of the stereo waveforms identified even fairly subtle driver interaction problems — and just as important, usually indicated which driver was at fault.

7. Implications for the Future

The Intel PC2001 Guidelines and Microsoft WHQL (Windows Hardware Quality Labs) logo requirements together dictate what types of hardware can and cannot be attached to a Windows PC. These specifications specifically forbid the use of MPU-401 style MIDI interfaces (e.g. any interface located at I/O address 0x330 or 0x300) and also preclude direct driver access to parallel and serial ports (introducing unpredictable system latencies into the use of such ports, and any MIDI interfaces connected to them). Current Macintosh PCs already have no options other than USB MIDI Interfaces and the forthcoming Firewire / IEEE-1394 interfaces.

The PCI Bus, CardBus and technologies such as USB and IEEE-1394 will continue to provide many ways to support MIDI interface capabilities. However, the increasing sophistication of commodity operating systems, ironically, makes it much harder to provide robust MIDI input and output with low latency and good temporal fidelity. Sustained efforts are needed to ensure that future systems support musical activities with performance adequate to the needs of professional, academic and art-music musicians and composers. We hope that our new performance testing technique will be a helpful tool towards that end.

Figure 5. Latency, Peak Jitter and Std. Deviation
(Round-trip Windows 98 SE MIDI System Performance)



8. References:

Brandt, E. and Dannenberg, R. 1998. "Low-latency Music Software Using Off-The-Shelf Operating Systems", *Proceedings of the ICMC*, (Ann Arbor)

Cota-Robles, E. and Held, J. 1999. "A Comparison of Windows Driver Model Latency Performance on Windows NT and Windows 98", Intel Architecture Labs <developer.intel.com/ial/sm/>

Freed, A. 1997. "Operating Systems Latency Measurement and Analysis for Sound Synthesis and Processing Applications", *Proceedings of the ICMC*, (Thessaloniki)

Garney, John 1996. "An Analysis of Throughput Characteristics of Universal Serial Bus", Media and Interconnect Technology, Intel Architecture Labs

Iyer, Bilmes et al. 1997. "A Novel Representation for Rhythmic Structure" *Proceedings of the International Computer Music Conference*.

Lunney, H. M. W. 1974. "Time as heard in speech and music." *Nature* (249):592.

Microsoft, Intel 2000. "PC2001 System Design Guide", <www.pcdesguide.org>

Microsoft, Intel et al 1998. Universal Serial Bus Specification, Revision 1.1

Microsoft 2000. Windows Hardware Quality Labs tests <www.microsoft.com/hwtest>

Moore, F.R. 1988. "The Dysfunctions of MIDI." *Computer Music Journal* 12(1):19-28.

Michon, J. A. 1964. "Studies on subjective duration 1. Differential sensitivity on the perception of repeated temporal intervals." *Acta Psychologica* (22): 441-450.

Schloss, A. 1985. "On The Automatic Transcription of Percussive Music From Acoustic Signal to High-Level Analysis." Ph. D. thesis, Stanford University, CCRMA.