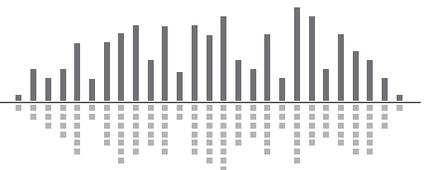


Symetrix White Paper: Latency of a SymNet System



Contents

Introduction	1
Where Does Latency Come From?	2
AD/DA Conversion	2
Digital Input and Output Cards	2
AEC Algorithm:.....	3
Digital Signal Processing:	3
Digital Signal Processing with Look-ahead Latency	4
Dante Digital Audio Buss	5
What Steps Can Be Taken to Compensate for Latency?.....	6
Enable Delay Compensation	7
Manually Compensating for Latency	7
Adding Latency on a Channel by Channel Basis.....	7
Adding Latency to Compensate for Dante Routing.....	8
Reasons to Not Enable Delay Compensation	9

Symetrix White Paper: Latency of a SymNet System

When designing or commissioning a SymNet system, it may be desirable to know the throughput latency of the signal path from the physical inputs, through all DSP modules and routing, to physical outputs. Knowing the throughput latency of audio within a SymNet system may be helpful in several applications including: syncing audio with processed video, installing SymNet for a live sound application and having a known starting point for speaker alignment, time aligning hardware in a broadcast facility, and ensuring multiple outputs across several SymNet devices are time aligned, to name but a few.

The first step is to understand where and how latency accrues, then what steps can be done to compensate for the latency.

Dante routing is performed by associating a receiving (Rx) channel with a transmitting (Tx) channel. This is called a subscription.

Where Does Latency Come From?

Latency accrues from various parts of a SymNet system:

- AD/DA Conversion
- Digital I/O
- AEC algorithm
- Digital Signal Processing
- Dante Digital Audio Buss

AD/DA Conversion:

Converting audio from an analog audio signal to a digital signal, and then converting it back from a digital signal to an analog audio signal creates latency. This is known as AD/DA conversion. Within the SymNet Composer platform all hardware uses the same A/D and D/A convertors, so conversion latency is consistent across all SymNet products.

The A/D conversion (analog to digital) of SymNet hardware is 0.28mS.

The D/A conversion (digital to analog) of SymNet hardware is 0.60mS.

Additionally, there is 0.042mS additional latency due to FPGA/TDM bus buffering.

This means the base latency of audio passing through a SymNet DSP, from analog input to analog output, would be 0.922mS when no DSP processing is applied to the channels.

Since audio is rarely routed through a DSP without any processing applied to the signal path, this 0.922mS conversion latency is the base, or starting point, of which all other latency will added when using analog inputs and outputs.

Digital Input and Output Cards:

The 4 Channel Digital Input and 4 Channel Digital Output cards both use sample rate conversion which results in an additional 139 samples of latency along with two samples of delay due to buffering, for a total of an additional 2.958mS latency for 48KHz signals.

If the digital input or output is running at a different sample rate than 48Khz, then the latency can be determined with the following formula: (Group delay + 0.063mS)

Group delay of the sample rate converter is dependent on input and output frequency:

Latency = $(102.53125/F_{in} + 36.46875/F_{out})$ seconds

For the digital input card F_{in} is the incoming sample rate, $F_{out} = 48Khz$.

For the digital output card $F_{in} = 48KHz$, F_{out} is the outgoing sample rate.

AEC Algorithm

Some of the more common supported sample rates and the associate latencies are listed below:

Sample rate ; Khz	44.1	48	88.2	96
Latency Digital In	3.147mS	2.959mS	1.985mS	1.891mS
Latency Digital Out	3.026mS	2.959mS	2.613mS	2.579mS

One other thing that should be noted is that there will be some additional latency added by the downstream digital device. The downstream device latency may vary for reasons such as sample rate conversion being enabled or disabled, whether the unit re-clocks the audio, and/or whether the unit buffers the digital audio.

AEC Algorithm:

During a teleconference or video conference session, the microphones located in the conference room are routed to the far end participant, while at the same time everything the far end participant says is played out the conference room speakers.

AEC (Acoustic Echo Canceling) is used to remove acoustic echo for the far end participant, caused when the far end audio plays out the conference room speakers and is picked up by the conference room microphones and routed back to the far end participant. Acoustic echo is distracting and makes conversing difficult, if not impossible.

The SymNet AEC algorithm adds 11mS of latency to the microphone audio coming out of the SymNet AEC Analog Ins module “AEC outs”. Normally this latency is not noticeable, because the audio from the AEC algorithm is sent directly to the far end participant.

However, if this AEC output is also routed to local conference room speakers, then the 11mS may not be desirable. For this reason, the SymNet AEC Analog Ins module also include “direct outs” which can be used to bypass the AEC algorithm and the 11mS of latency it adds to any signal routed through it. Typically, the “direct outs” are used for local reinforcement of the conference room mics, or for any media sources connected to the AEC card analog inputs. The AEC direct inputs incur only the 0.30mS of A/D plus buffering latency.

Digital Signal Processing:

All DSP designs have some inherent latency, and the DSP modules used directly affect the overall throughput latency of a SymNet signal path.

First, it should be noted that processing latency on a dedicated DSP accrues differently than latency which accrues on a PC processing audio in a DAW, such as Pro Tools or Cubase. A PC typically uses block-based processing where it gathers up 32 or 64 samples of audio and then processes all at once. The end result of PC block-based processing is that the more processing needed, for instance adding more VST plugins to a channel, the more processing latency can accrue because the OS has to reallocate memory and resources to accommodate all the various operations happening simultaneously on the PC.

On a dedicated DSP the entire signal path, all DSP modules and routing, is processed in one single sample. (Note: the exception to this is DSP modules that use a look-ahead time. More on this later.)

To repeat, a single sample is used to process the “entire” DSP signal path. A single sample at 48kHz is equal to 20.833 microseconds, which rounded for simplicity is 0.021mS. Latency is not accrued on a module per module basis.

Think of it this way. An analog input feeds a gain module, then an EQ, then a selector, and finally speaker manager (with delay disabled) in signal path across all channels. At the beginning of the sample period, the DSP takes the input signal, does the gain, feeds to output to the EQ, does the EQ, feeds the output to the selector, does the selector, feeds the output to the speaker manager, does the speaker manager, and feeds the final result it out the analog output. All this is done within the same 48 kHz sample period of 0.021mS.

SymNet DSP hardware uses a 400MHz SHARC chip as the dedicated DSP for processing the audio signal path.

It should be noted that in addition to the single sample used to process the entire DSP signal path, there is an additional sample of latency “in” and “out” of the SHARC chip used for DSP buffering, adding 0.042mS of latency to process the signal path.

What this means, is that as long as a signal path is not using a DSP module with a look ahead time, nor using any delay modules, then the entire signal path from analog input to analog output within a SymNet DSP is processed in 0.985mS. This is the combined AD/DA conversion and FPGA/TDM bus buffering latency of 0.922mS, plus the DSP processing and buffering latency of 0.063mS. That is less than one millisecond of total latency for processing the entire signal path. Now that is fast!

Digital Signal Processing with Look-ahead Latency:

As mentioned, some modules do cause latency beyond that of a single sample period. These DSP modules use a look-ahead time, and the latency of each is additive to the particular path they reside on. A look-ahead time is a generic term used when an algorithm delays the output of a process in order to react more quickly to the input of the algorithm. To better understand this concept, consider analog AGC vs. the digital AGC in SymNet. An AGC takes any audio signal routed to the AGC input and attempts to output that same audio at a target RMS level. AGC can be helpful for maintaining a consistent volume for any source where audio levels can vary greatly, such as microphones used by different participants or iPod docking station.

In the past, Symetrix manufactured the 422 and 421M AGC Leveler, both analog automatic gain control devices. Sometimes a user would notice that after a period of no input, there would be an audible period where the AGC output would have to ramp up or down the audio to the target output level. This was because the analog circuitry could only respond to the analog input signal in real-time, when it truly needed to know what level the audio that was about to hit the input would be at. In other words, the AGC needed to see into the future so the analog circuitry could respond and be ready for what was about to pass through the device. Obviously seeing into the future is impossible.

However, this is what a look-ahead time provides a digital AGC, a brief look into the future of what is about to come through the algorithm. This means that there is never an audible ramp up or down of the output of the AGC to reach the target output level, because the digital AGC delays the output so that it can always react to what is coming down the AGC pipeline. The look-ahead time in SymNet Composer is 24 samples, and is common to all modules that use a look-ahead.

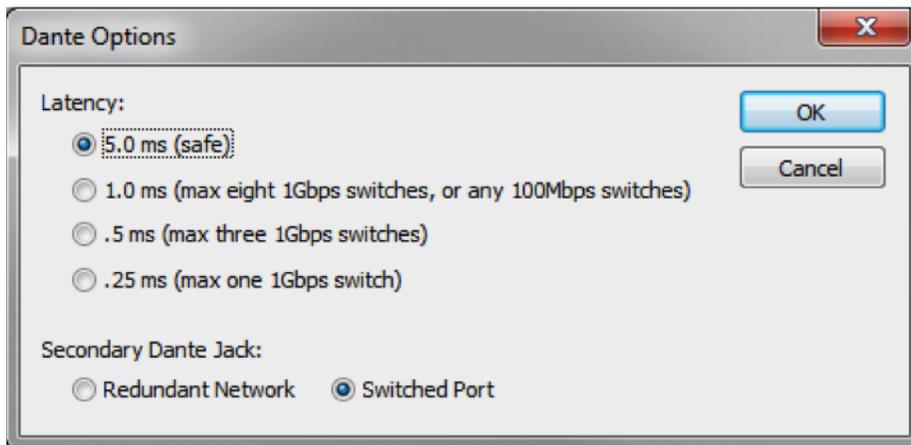
The DSP modules that use a look-ahead are:

- AGC (Automatic Gain Control)
- Compressor
- Limiter
- De-Esser
- Expander
- Gate
- Multi-Dynamics

24 samples at 48kHz equals 0.5mS, or one half of a millisecond. This means in a SymNet signal path, from analog input to analog output, every instance of the above listed modules will add 0.5mS of latency to the DSP chain they are in.

Dante Digital Audio Buss:

SymNet Composer hardware uses Dante to transfer digital audio between SymNet devices over off-the-shelf Ethernet hardware. When Dante is routed from one SymNet device to another, some delay, also known as latency will accrue due to the transfer of the digital audio between the transmitting device and the receiving the device or devices. Latency settings for a SymNet system are based on the network switch topology in which Dante digital audio must travel from one SymNet device to another. The latency setting is selected by going to the Dante Flow Manager and hitting the “Configure Dante...” button.



The speed of the network switches and the longest path Dante will have to travel on the network determines the latency settings that are applied for all SymNet units.

For instance, if there is a single 100Mbps network switch used for the Dante network, then a 1.0mS latency setting would be lowest latency setting supported, and all SymNet Dante enabled devices in the SymNet system would incur 1.0mS of latency when audio is transferred via Dante between the two units. If this single network switch was a gigabit network switch, then the lowest latency setting of 0.25mS would be supported.

Additionally, the longest network path determines the latency setting for the SymNet system. For instance, take a system where five SymNet Dante enabled devices are on one network switch, and then one other SymNet Dante enabled devices are on another switch that is 4 network switch hops away. This means there is a total of five switch hops in the longest Dante network path. As such, the 1.0mS latency setting would need to be selected to support the longest run of 5 switch hops.

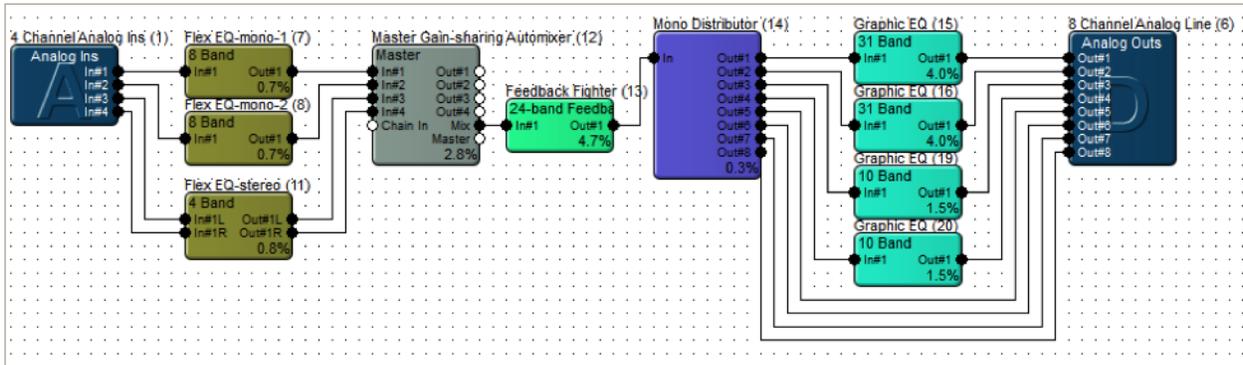
It is important to note that Dante latency is an absolute setting, it is not additive. In other words, latency does not build up as Dante travels through a series of network switches. In the example above, all units, those with only one switch hop separating them and those with 5 switch hops separating them, receive the Dante audio at the same latency setting of 1.0mS which is the latency selected in the Dante Options.

What Steps Can Be Taken to Compensate for Latency?

What Steps Can Be Taken to Compensate for Latency?

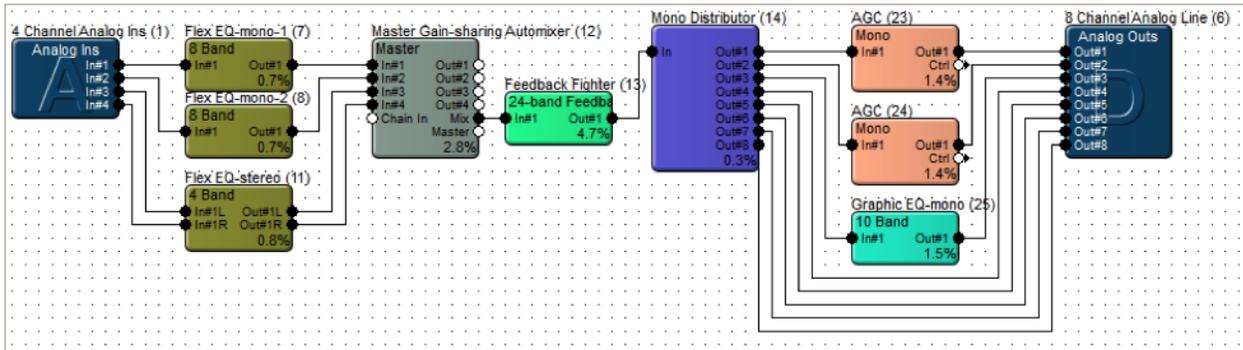
There are several methods for compensating for latency in a SymNet system. First, calculate the latency of the signal path, paying special attention to where latency accrues, and determining whether some form of compensation is needed to insure the outputs receive the same sample and the exact same time.

Example 1:



The above signal path does not contain any modules that use a look-ahead time. As such the entire signal path is processed in one sample cycle and buffered in two sample cycle or a combine latency of 0.063mS. This means, from analog input to analog output, the latency of all channels in this signal path is 0.985mS when conversion latency is added to processing latency. All outputs would receive the same sample at the exact same time.

Example 2:



The above signal path contains AGC modules, which use a look-ahead time, on outputs 1 and 2. As such the entire signal path is processed in one sample cycle and buffered in two or 0.063mS, excluding outputs 1 and 2 which have an additional latency of 0.5mS. This means when conversion latency is added to processing latency, from analog input to analog output, the latency of channels 1 and 2 is 1.485mS while the latency of outputs 3 through 8 is 0.985mS.

If speaker alignment was absolutely necessary across all 8 outputs, then it is evident that the output of channel 1 and 2 would not be in perfect synch with outputs 3 through 8. SymNet Composer has a tool for aligning the samples as they travel through a SymNet DSP called "Delay Compensation".

Enable Delay Compensation

Enable Delay Compensation:

In SymNet Composer, in the Site Preferences found under the Tools menu, there is an option that can be selected entitled “Enable Delay Compensation”. This feature time aligns samples on a DSP by DSP basis. If a single sample is routed to multiple outputs within a DSP, and one or more of the routes acquire latency by being processed with a module that uses a look-ahead time, then delay compensation would be added to the other routes so that the distributed sample plays at the exact same time across all analog or digital outputs.

It should be noted that delay compensation is also performed across inputs to Dante transmit modules. So if multiple signal paths hit Dante Transmitters, delay compensation will time align samples of all audio leaving the unit via Dante.

Basically, the idea is that when delay compensation is enabled, all audio leaving the unit via physical outputs is time-aligned and all audio leaving the unit via Dante is time-aligned. Each treated as a separate system.

In the above Example 2, if Delay Compensation is enabled, all outputs would acquire the same latency as the signal path containing the AGC module. In other words, the entire DSP design, from analog input to analog output, would have a throughput latency of 1.485mS.

It is important to note that Delay Compensation works only within each DSP individually. It does not time align samples spread across the SymNet system using Dante. Compensating for Dante transmit latency must be done manually within the SymNet site file.

Manually Compensating for Latency:

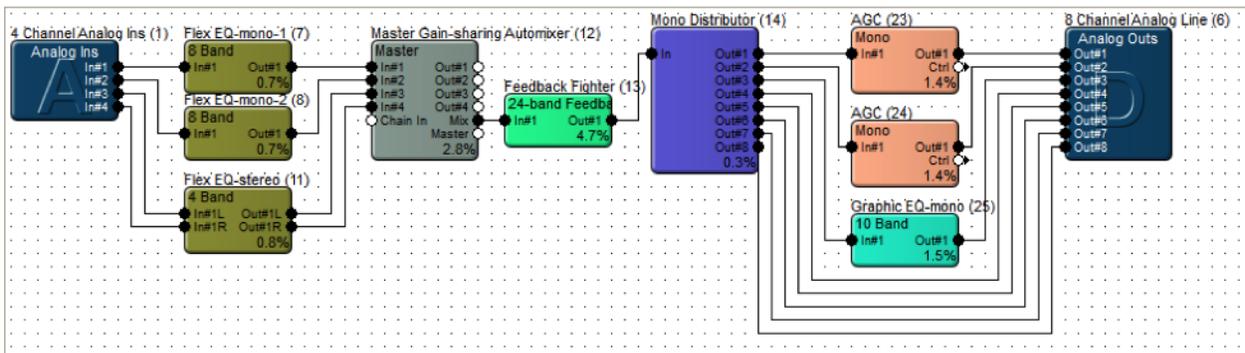
There are two cases in which manually compensating for accrued latency is desired or necessary.

1. Adding latency on a channel by channel basis
2. Adding latency to compensate for Dante routing

Adding Latency on a Channel by Channel Basis:

For instance, take Example 2:

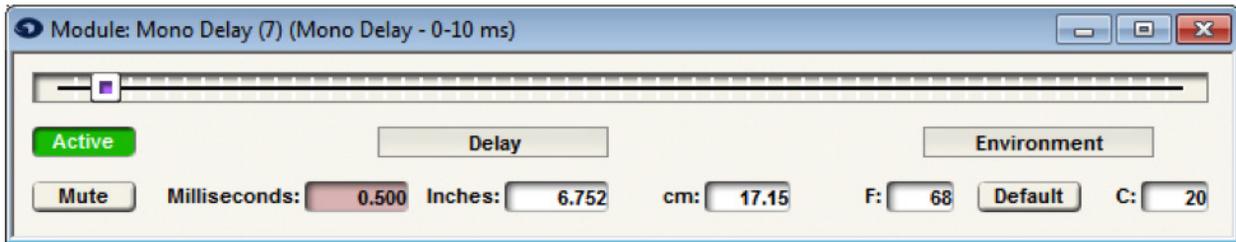
Suppose that outputs 1, 2 and 3 are a stereo 2.1 system and need to be time aligned, where output 3 is the subwoofer. Outputs 4-8 are feeding sources that do not need to be time aligned.



Enabling delay compensation would delay all outputs to a throughput latency of 1.485mS, because of the added 0.5mS latency that the AGC adds to outputs 1 and 2. If adding 0.5mS latency to outputs 4-8 was undesirable, while at the same time outputs

Adding Latency to Compensate for Dante Routing

1-3 need to be time aligned, then manually adding 0.5mS of latency to output 3 would be the way to achieve the desired results. Leave “Enable Delay Compensation” unchecked in the Site Preference, and then place a 0-10mS Mono Delay module to the signal path of output 3. The Mono Delay would be set to 0.500mS.



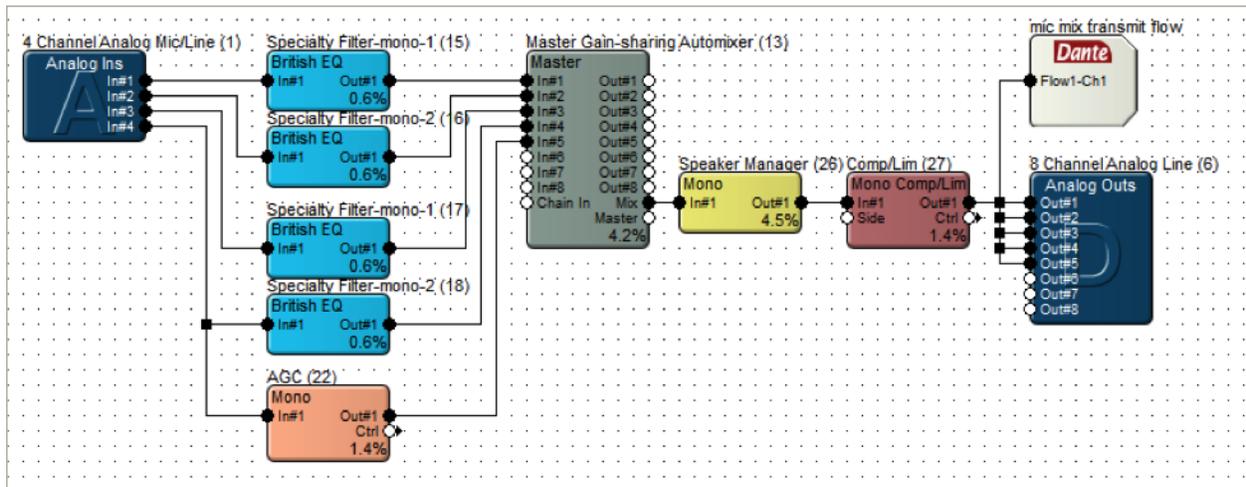
Adding Latency to Compensate for Dante Routing:

Regardless of if “Delay Compensation” is enabled or not, SymNet does not automatically compensate for latency accrued by transmitting audio across Dante. That being said, since the Dante latency is an absolute value that is set for all Dante flows transmitted on the network, manually compensating for Dante related latency is simple and straight forward.

First, open the Dante Flow Manager and hit the “Configure Dante...” button to open the Dante Options for the SymNet system. Here you will note which latency setting the SymNet system is currently using; 0.25mS, 0.50mS, 1.0mS, or 5.0mS.

Next, find a transmitted Dante channel that is also sent to the local DSP analog or digital output, similar to Example 3a below.

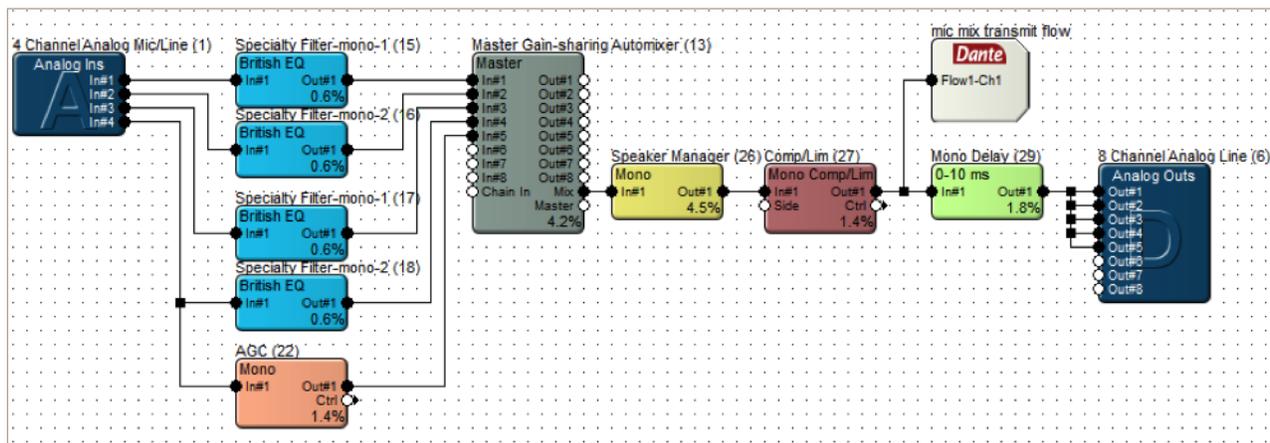
Example 3a:



Since the Dante Latency is a known value from the Dante Flow Manager, a 0-10mS Mono Delay module can be added to the output signal post the Dante Transmit Flow. This would create equal latency between the local physical outputs and the audio after it is transmitted over Dante to another DSP. Set the mono delay to the same setting as the Dante latency. Now, as long as no processing using a “look ahead” is in line after the Dante flow is received in the downstream SymNet devices, the local output will be time aligned with the outputs on the receiving SymNet devices. See Example 3b for a wiring example.

Reasons to Not Enable Delay Compensation

Example 3b:



Note: If additional latency is accrued in the SymNet device receiving the Dante audio, then additional delay can be added to compensate on the local output. Remember, each module that uses a look ahead adds an additional 0.5mS of latency.

Reasons to Not Enable Delay Compensation:

There are several reasons why manually compensating for accrued latency is a superior method for time aligning outputs, rather than simply enabling delay compensation.

First, enabling delay compensation taxes the DSP whether it is used or not. By enabling delay compensation, each physical and Dante output essentially has to have a delay inserted “under the hood” just in case it needs time alignment. As such, for every 8 physical outputs or 8 Dante transmit channels, the DSP must reserve roughly 1% of its DSP. If a unit had 8 physical outputs and 64 Dante outputs, the system would reserve 9% of the DSP for delay compensation when it is enabled. In a system where DSP processing will be used extensively and resources may be tight, giving up 9% of the DSP horse power prior to dropping a single DSP module into the design would be wasteful and potentially interfere with creating the desired signal path.

The second reason for not enabling delay compensation is that time alignment of all of the outputs within every signal path in SymNet may not be needed, so turning on delay compensation simply delays some outputs unnecessarily.

Most often time alignment between outputs will not matter since the audio may be going to completely separate acoustic spaces, to a recording device, etc. As such, manually compensating for accrued latency on the outputs that require time alignment is superior to globally enabling delay compensation. Manually time aligning outputs using the methods outlined in this document ensures that the DSP programmer has access to 100% of the SymNet DSP resources, while at the same time ensuring that delay is only added to outputs that need to be time aligned with other outputs.

Symetrix White Paper: Latency of a SymNet System

© 2013 Symetrix, Inc. All rights reserved. Printed in the United States of America. The information in this document is subject to change without notice. Symetrix, Inc. shall not be liable for technical or editorial errors or omissions contained herein; nor is it liable for incidental or consequential damages resulting from the furnishing, performance, or use of this material. Mention of third-party products is for informational purposes only and constitutes neither an endorsement nor a recommendation. Symetrix assumes no responsibility with regard to the performance or use of these products. Under copyright laws, no part of this brochure may be reproduced or transmitted in any form or by any means, electronic or mechanical, without permission in writing from Symetrix, Inc. If, however, your only means of access is electronic, permission to print one copy is hereby granted. The following are either Trademarks or Registered Trademarks of Symetrix, Inc.: Symetrix, SymNet, SymNet Designer, SymLink and CobraLink. Windows is a Registered Trademark of Microsoft, Inc. Other product names mentioned herein may be trademarks and/or registered trademarks of other companies and are property of their respective owners.

