# On the Transport of a Constant-Bit-Rate (Time Sensitive) Application Across a Packet Network That Imposes Variable Delay

**Geoffrey M. Garner**
**Samsung (Consultant)**
**For 802.1 AVB TG**
**February 8. 2006**

The purpose of this paper is to clarify the relation between the jitter and wander performance for a constant bit rate (CBR), time-sensitive application that traverses a packet network, and the jitter and wander performance of clock signals that may be available at the network nodes and are synchronized with each other (but not necessarily with anything else). This is in response to discussion of this topic in the February 1, 2006 AVB call and to Reference [1].

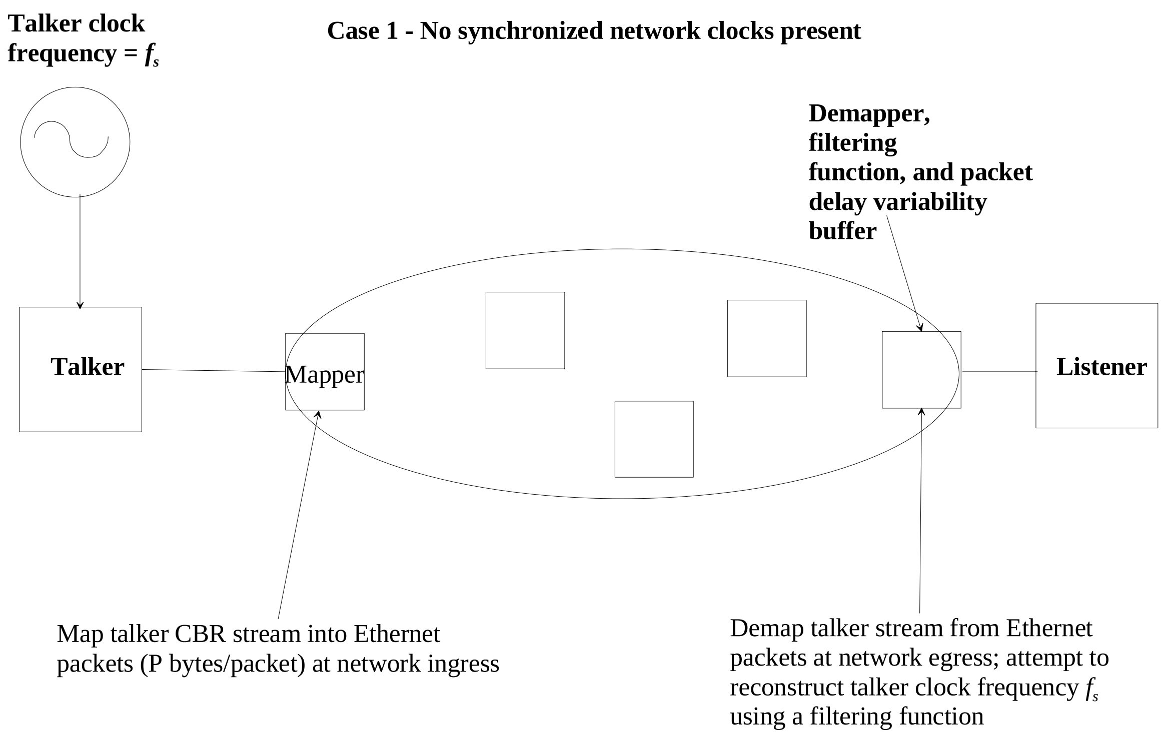


**Figure 1. Illustration of transport of CBR application over a packet network when synchronized network clocks are not available.**

Figure 1 illustrates transport of a CBR application from a talker to a listener over a packet network for which the delay across the network is not constant, and for which synchronized network clocks are not present. The talker frequency is $f_s$ bytes/s and is assumed to be nominally constant. Any jitter and wander due to variation of the talker frequency is assumed to be well within the CBR application interface jitter and wander requirements, i.e., the requirements that must be met at the input to the listener. These requirements may be expressed in terms of MTIE, TDEV, and peak-to-peak jitter over a specified time interval (though other measures of jitter and wander are possible).

At the mapper, packets associated with the CBR application are presented to the network at rate $f_s/P$, where $P$ is the number of application bytes in one packet. The packets traverse the network to the demapper. Since the network imposes packet delay variability, the packets arrive at the demapper with instantaneous rate that varies with time. The application bytes are demapped from the packets at the demapper and must be presented to the listener at a rate whose average is $f_s$; in addition, any jitter and wander due to variation from this rate must be within the application jitter and wander requirements (MTIE, TDEV, peak-to-peak jitter measured over a specified time interval, etc.). There are two reasons these requirements must be met. First, the demapper contains a finite size buffer to absorb the packet delay variability; if there is a difference between the rate of removal of bytes from this buffer and the rate at which bytes are delivered to the buffer that is sustained for a sufficiently long time, the buffer will overflow or underflow and result in an impairment in the application. Second, the listener interface has existing jitter and wander tolerance requirements, and in general cannot tolerate jitter and wander beyond these requirements (because the listener was designed to meet these requirements.

Synchronized network clocks are not present in Case 1. One class of schemes for recovering (i.e., reproducing) the CBR frequency $f_s$ at the demapper are adaptive clock recovery schemes, i.e., using a low pass filter to control the rate of removal of bits from the demapper buffer to keep the average buffer fill constant. It was shown in [2] that such schemes, in general, cannot meet the MTIE requirements for audio and video applications when multiple, competing CBR streams are present. The reason for this is that the streams "beat" against each other, which results in isolated phase steps of large amplitude that cause the respective MTIE masks to be exceeded. The results in [2], among other things, led to the conclusion that synchronized network clocks are needed in an AVB network to enable jitter and wander requirements for time sensitive applications to be met.

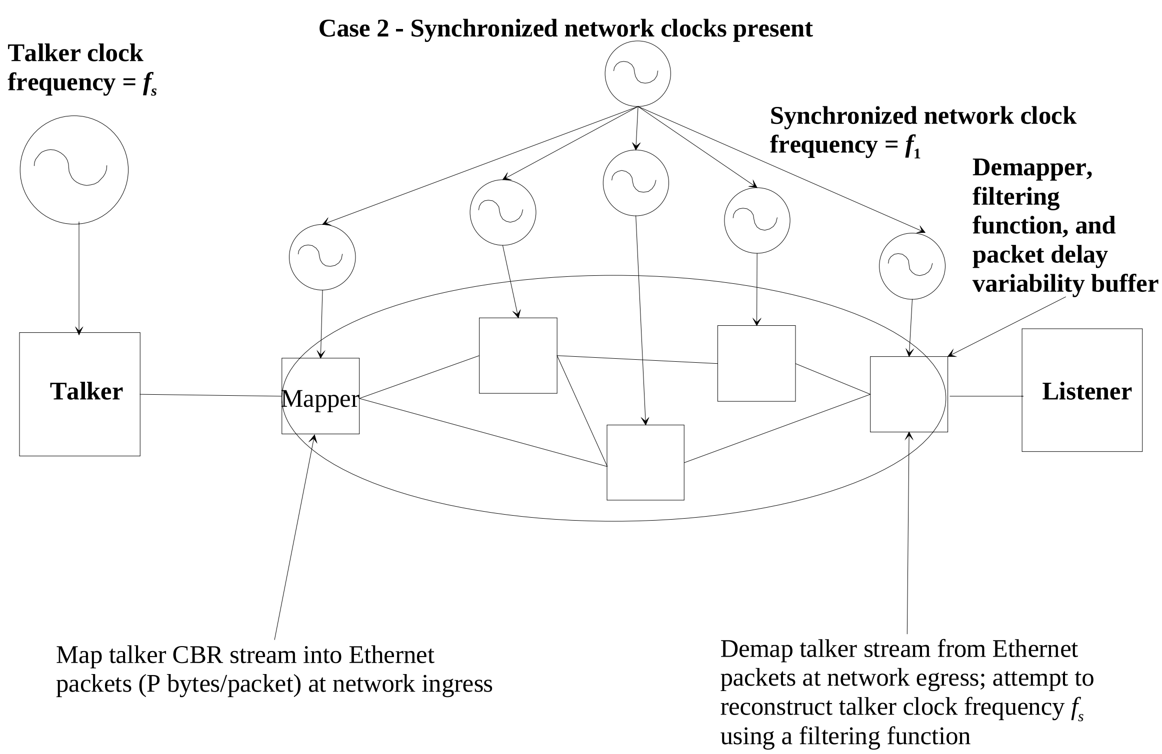


**Figure 2. Illustration of transport of CBR application over a packet network when synchronized network clocks are available.**

Figure 2 illustrates transport of the same time-sensitive, CBR application as in case 1, except that now a synchronized timing signal, i.e., a synchronized clock signal, is available at the ingress and egress nodes. In fact, the signal is available at all network nodes; however, the CBR application illustrated here only requires that the signal be present at the mapper and demapper. The network clocks are synchronized with respect to each other and have frequency $f_1$; however, there is no requirement that they also be synchronized to UTC, TAI, or any other timing source outside the network. In this example, the network may be free-running relative to the rest of the world, or may be synchronized to external sources; the only requirement is that the clocks within the network be synchronized relative to each other (and have frequency $f_1$). In particular, the talker clock is not synchronized relative to the network clocks; in general, $f_s \neq f_1$. This will be the situation in many applications, e.g., if digital video is delivered from a video content provider over one or more service provider networks to the residence, the clock at the video source in the studio will, in general, be different from the service provider networks' timing and/or the AVB timing. If digital audio originating from a CD player or a musical instrument is transported over the AVB network, the audio source clock will, in general, be different from the AVB timing.

The synchronized clock signals at the AVB network nodes may be provided using one of the synchronization schemes that have been discussed in the AVB TF, e.g., the scheme described in the White Paper [3] and/or IEEE 1588 [4], with the various phase and/or frequency compensation schemes discussed in [5] and [6]. In the current paper, we are not concerned with the particular scheme used to supply synchronization signals to the network nodes; rather, we are concerned with the use of the synchronization signals to transport the CBR application with acceptable jitter and wander accumulation.

There are a number of techniques, with numerous implementations, for using the network clock signal $f_1$ at the network ingress and egress to aid in reconstructing the CBR service clock $f_s$. One well-known method is the following (see IEC-61883 [7]): (1) When the arriving application bytes have just filled an Ethernet packet, write the current time indicated by the network clock, in units of cycles of $f_1$, in the packet. (2) Add to this time a fixed value that is greater than or equal to the maximum delay across the network. (3) When the packet arrives at the network egress, read the time stamp in the packet. (4) When the network clock at the egress time reaches the time stamp value, output a pulse. The resulting sequence of pulses will be a clock whose average rate is $f_s/(8P)$, but which has occasional phase steps whose size is one network clock period ($1/f_1$) due to the granularity of the network clock.[1] Finally, input the pulse stream to a phase-locked loop that multiplies the frequency by $8P$ to produce the CBR application clock rate $f_s$ and that filters the phase steps to keep the jitter and wander within the application requirements.

The fact that the network and CBR application clocks are not synchronized to each other does not affect the ability to transport the application timing. The only requirement is that the network clocks be syntonized, i.e., that the network supplied synchronization signals at the mapper and demapper have the same frequency.[2] Any difference between the network clocks at the mapper and demapper will be transferred to the recovered application timing. This may be seen from the fact that the network clock at the ingress is used to measure when one packet's worth of application bits arrives, and the network clock

[1] More specifically, the value of each time stamp represents the most recent network clock value at the ingress. The difference between the quantized time stamp value and the value the time stamp would have if the quantization granularity were zero accumulates with each packet; when it reaches a full network clock cycle the result is a phase jump of one network clock cycle in the sequence of output pulses at the egress. The long-term average rate of the output pulse stream is $f_s/(8P)$, with some pulses slightly closer together and some slightly further apart (and the time difference between the two different sized inter-pulse intervals is $1/f_1$).

[2] If the network clocks at the ingress and egress are synchronized, i.e., indicate the same time as well as have the same frequency, then they will meet this requirement. If the clocks are syntonized but not synchronized, they can differ by a fixed offset, i.e., their difference is constant in time. In this case, the fixed value added to the ingress clock time to create the time stamp must bound from above the sum of the maximum delay across the network and the maximum fixed offset specified for the network clocks (i.e., the network clock time synchronization requirement).

at the egress is used to reconstruct when the data in this packet should be presented to the application layer. If the two network clocks differ in frequency, the difference will show up as jitter and/or wander in the reconstructed application clock.

Note that the packet delay variability imposed by the network will likely be large compared to any jitter and wander in the reconstructed application clock at the egress. The buffer at the network egress will absorb the packet delay variability, i.e., it will hold the application bytes while they wait to be clocked out to the listener at the reconstructed clock rate. The size of this buffer is determined by the packet delay variability and, to a lesser extent, by the jitter and wander on the reconstructed application clock.

In summary, the application jitter and wander accumulation over the network depends on the jitter and wander of the integral of the frequency difference between the network synchronization signals at the mapper and demapper. The control of application jitter and wander accumulation does not require that the application and network clocks be synchronized with each other, nor that the network clocks be synchronized to anything outside the network.

**References**

[1] Max Azerov, *On Relation Between the Application MTIE and Synchronization Accuracy*, paper prepared for 802.1 AVB group, February 7, 2006.

[2] Geoffrey M. Garner, *Delay Variation Simulation Results for Transport of Time Sensitive Traffic over Conventional Ethernet*, Samsung presentation to IEEE 802.3 ResE SG, San Francisco, CA, July 18, 2005.

[3] David V. James, *Residential Ethernet (RE)(a Working Paper)*, content prepared by David V. James and other contributors and maintained by David V. James, November 16, 2005.

[4] IEEE Std. 1588, *IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems,* November 8, 2002.

[5] Geoffrey M. Garner, *Summary of Potential Choices for Synchronization Transport in Residential Ethernet*, Samsung presentation to IEEE 802.3 ResE SG, San Francisco, CA, July 19, 2005.

[6] Sihai Wang, *Improvements to Boundary Clock Based Time Synchronization through Cascaded Switches*, Samsung presentation to IEEE 802.1 ResB SG, Vancouver, BC, November, 2005.

[7] IEC 61883, *Consumer Audio/Video Equipment – Digital Interface*, Part 1: General Specifications, Part 2: SD-DVCR Transmission, Part 3: HD-DVCR Transmission, Part 4: MPEG-2 TS Transmission, Part 6: Audio and Music Data Transmission Protocol, IEC, 2002.