

Introduction

Many telemetry systems require multiple serial data streams be transported across a ground network. If the data streams must be re-generated as the original serial data streams, this regeneration must be done such that they have the same time correlation to each other. Often times, the time stream itself must also be regenerated.

This App Note discusses methods for collection, transport, and time-correlated regeneration of serial data streams.

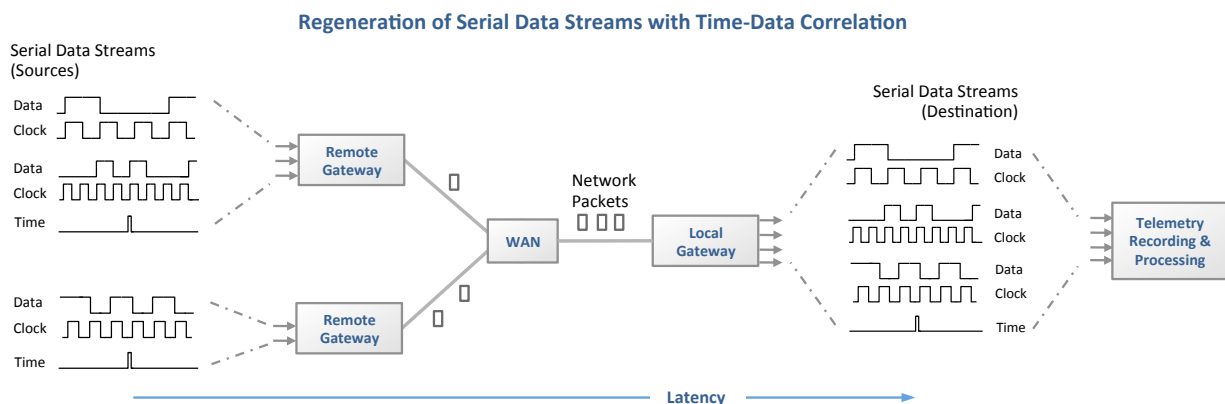
Background

Telemetry data is often produced as serial data streams from the receivers/bit synchronizers. These streams may be generated at a test site, but processed at a separate telemetry processing center. In the long ago past, long distance serial networks were used to make the connection. Wide area networks have replaced these circuits.

With WANs, there is network packetization of the data. The time correlation of multiple telemetry data streams and time data stream must be maintained. A range of techniques is used to accomplish the regeneration of serial data streams with time-data correlation.

Overview Diagram

The diagram below shows two sites where telemetry streams are generated and then sent over a WAN to another site where the data is regenerated and then processed.



Network Transport of Serial Data Streams

Network transport requires packetization of the data, and this packetization necessarily adds some delay (or latency). With packetization, data bits are buffered at the remote network gateway over each packet period. During this packet period, the remote gateway collects data bits for each serial data stream and combines those bits, along with time information, into the next packet sent across the network.

The WAN induces some additional latency along with jitter. Jitter is the variation in the receipt of the packets on the other end. This jitter requires the local gateway to buffer the data to avoid a data underrun condition. In other words, it must hold up output of the serial data streams to ensure it doesn't run out of data before the next packet is received.

Real time systems may use a WAN FEC (Forward Error Correction) technique to allow UDP to be used as the network protocol. TCP can reorder packets and retransmit lost packets which leads to higher network latency and jitter. UDP is more efficient in that packets are transmitted once (aka fire and forget). A WAN FEC allows for data recovery in the event that packets are dropped.

Time-Division Multiplexing

One approach is to time-division multiplex each of the data streams at the remote gateway along with an IRIG signal. Each data-time stream occupies a pro-rata portion of the aggregate data stream. Synchronization markers are added to the TDM'd stream to create frames and this data is packetized and sent across the network to the local gateway.

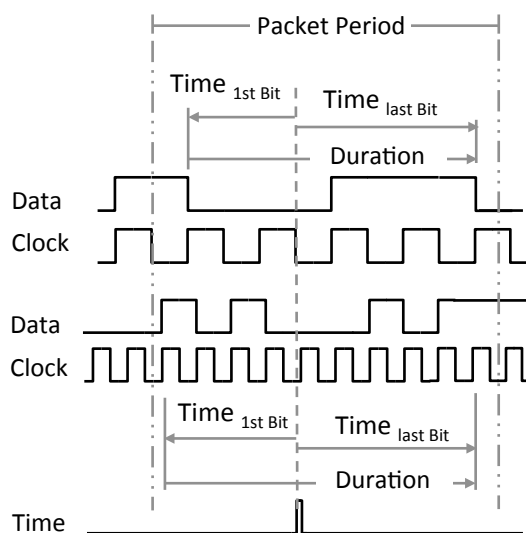
The local gateway receives the packets, performs the master frame synchronization and then extracts out (demultiplex) each of the data streams. The serial data/clock and time signals are regenerated in hardware.

This approach is very hardware/firmware centric. Any TDM multiplexing scheme introduces some sliding skew in the data as data is "held up" waiting for its slot to come around. It works best when the various serial data rates are integer multiples of each other, and more importantly, sourced from the same clock. Otherwise, there's some amount of jitter in the time-data correlation.

Time-Based Regeneration

Another approach is to capture the time information necessary to regenerate the data, along with the data in each packet. For example, the time information might include the starting and ending time for the packet, along with the time offsets for the first and last bit of each stream. Another approach captures the absolute time for the first bit of each stream in the packet along with the duration for the number of bits in that packet. The time information for each serial data stream is independently captured.

Time-Tagging of the Bits in a Packet



The local gateway uses hardware-level time-release capabilities and is configured to output the serial data streams with a time offset that accounts for the packetization and network delays. The data streams can be accurately regenerated along with any timing streams.

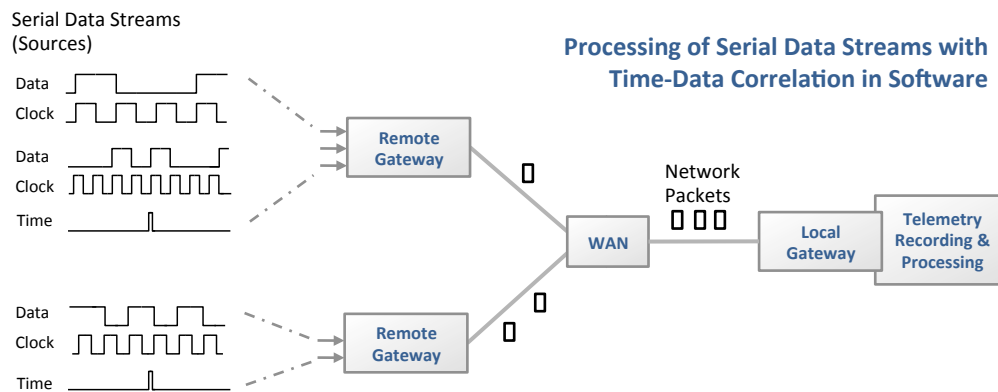
The network bandwidth needed to transmit the information for each data-time stream is slightly greater than that needed for a TDM approach since the packets contain each data stream's time information. The key benefit is that data from multiple sites can be regenerated with time correlation between them.

When properly implemented, this method can achieve very accurate time-data correlation, on the order of 20 microseconds.

Software-Based Time-Data Correlation

The serial data streams being regenerated are subsequently processed by systems/software that performs functions such as frame synchronization, time-tagging, decommutation, EU conversion, etc. In this process, the time information associated with each telemetry measurand is calculated and associated with the data.

Modern systems can perform the necessary time-data correlation *without* having to regenerate the serial data streams. This eliminates hardware and cost in the telemetry processing center. These modern software-based systems can effectively re-associate the time information in the processing.



For example, one of the first steps in processing is to time-tag the telemetry frames. The time information in the network packets from the remote gateway associates time with specific bits. Knowing where the time hacks associated with a specific bit that precedes the sync pattern and with a bit that follows the sync pattern, the frame synchronizer can establish the time of the sync pattern by interpolating between these two bit times. With each telemetry frame time-tagged, the decommutation software can process and accurately time-tag each telemetry measurand.

Since the data flows through the software processing as it arrives, these software-based approaches can reduce system latency by eliminating some of the buffering at the local gateway. Software-based approaches also afford greater flexibility.

The potential downside is interoperability with other systems. Serial data lines provide an easy (lowest common denominator) way to connect systems. Software-based systems may require data translation. Most software-based systems can be configured to regenerate physical outputs as serial data streams.

Helpful Links:

Related Products:

softFEP Network Front End Processor

Can we help? AMERGINT's expertise is available to assist in your systems engineering and design

Randy Culver
randy@amergint.com

Office: 719-522-2802
Cell: 719-930-6304