REORDERING PACKET BASED DATA IN REAL-TIME DATA ACQUISITION SYSTEMS

Stephen Kilpatrick, Galen Rasche, Chris Cunningham, Myron Moodie, Ben Abbott
Southwest Research Institute®
Automation and Data Systems Division
ben.abbott@swri.org

ABSTRACT

Ubiquitous internet protocol (IP) hardware has reached performance and capability levels that allow its use in data collection and real-time processing applications. Recent development experience with IP-based airborne data acquisition systems has shown that the open, pre-existing IP tools, standards, and capabilities support this form of distribution and sharing of data quite nicely, especially when combined with IP multicast. Unfortunately, the packet based nature of our approach also posed some problems that required special handling to achieve performance requirements. We have developed methods and algorithms for the filtering, selecting, and retiming problems associated with packet-based systems and present our approach in this paper.

KEY WORDS

IP networks, Packets, Data Acquisition, Reordering

BACKGROUND

While previously a bad choice due to performance, data loss, hardware size, lack of ruggedness, etc. IP hardware is now an excellent alternative for implementing real-time data collection and processing systems. In fact, features such as low-cost hardware, tool leveraging from the internet domain, precise clock synchronization (e.g. IEEE 1588), gigabit performance, and robust cables and connectors demand the consideration of IP approaches as possible implementation methods.

We have recently integrated a large airborne instrumentation system and utilized an IP approach. The system is capable of sustaining a continuous gigabit network data collection and processing rate. Data passed via IP from sensors such as strain gauges, video monitors, and position sensors, is 'tapped' off by various IP based clients in the network. We leverage IP multicast to avoid duplicate copies of data flowing through specific branches in the network and thereby allowing for scaling across multiple clients. Thus, a safety analyst might subscribe to a portion of data and show real-time plots with respect to his interest while a performance analyst might be looking at some of the same data (combined with other parameters) to see if the aircraft is performing correctly. Further, a bandwidth constrained telemetry link might be selecting further data that is of interest to watch from a ground station. We have found the open, pre-existing IP tools, standards, and capabilities support this form of distribution and sharing of data quite nicely.

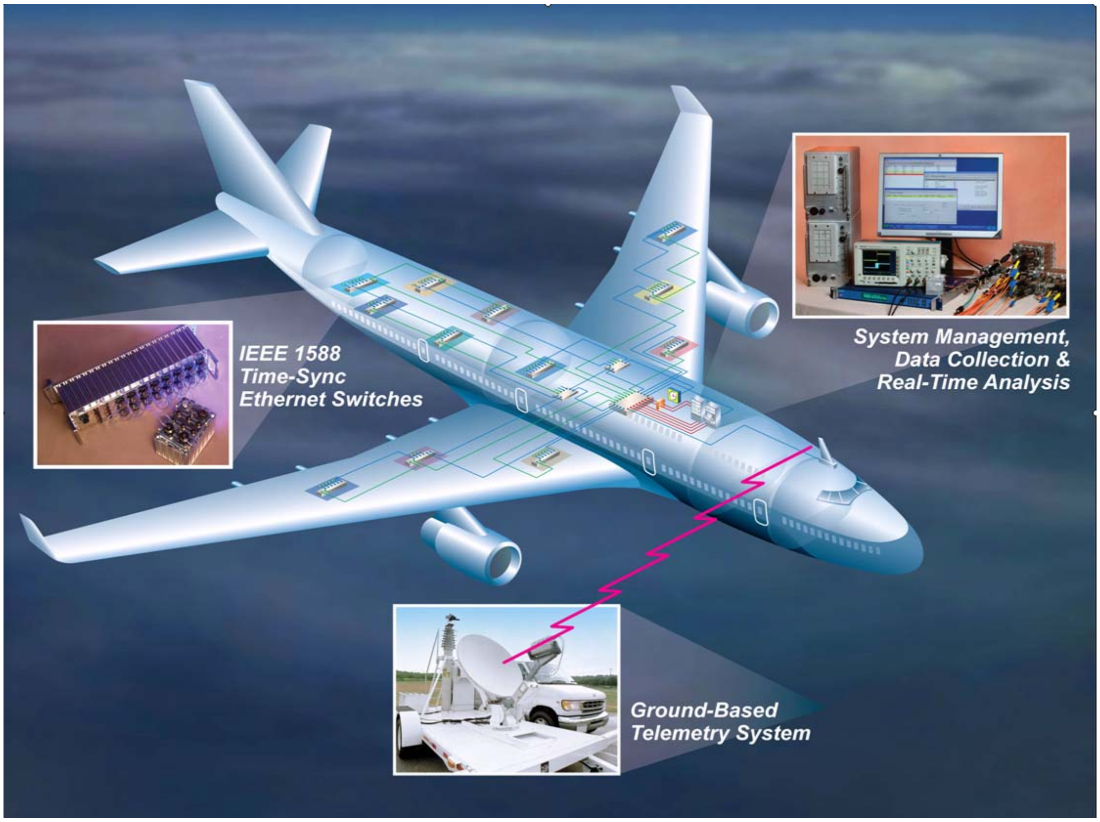

Figure 1. Network-Enhanced Data Acquisition and Telemetry System

NEED FOR RETIMING AND REORDERING

Unfortunately, the packet based nature of our approach posed some problems. In particular, IP protocols perform quite poorly if packets are very small. Thus, a single reading from a strain gauge should not be sent alone. Rather, it should be grouped with other readings. This can be accomplished by either batching multiple readings across time from the same gauge to form a single packet, or combing the data with other sensor readings. It is this forming of packets that causes problems. Consider a live strip chart display showing the strain of a single strain gauge. If data is presented on the strip chart as it is received in packets, the chart will burst forward each time a packet is received. Figure 2 shows this gapping based on the packets coming from a single sensor source (Data Acquisition Unit "A" (DAU-A)). Thus, in order to present the data to the strip chart in a representative manner, the data must be retimed. That is, individual data items from the packets must be pulled into a stream, delayed, and then be presented to the strip chart one at a time based on their original sample rate. Retiming implies a few things:

• Packets must not only include data items, but also must include temporal information concerning when the data items occurred.
• Data items being correlated from two different sources must be synchronized and collated back together in time.
• There must be a global clock standard. We use IEEE 1588.
• Reordering of data across different packet sources involves a sorting problem. If done wrong, this could be quite time consuming at gigaabit network rates.
• Specific data items of interest must be filtered and selected.

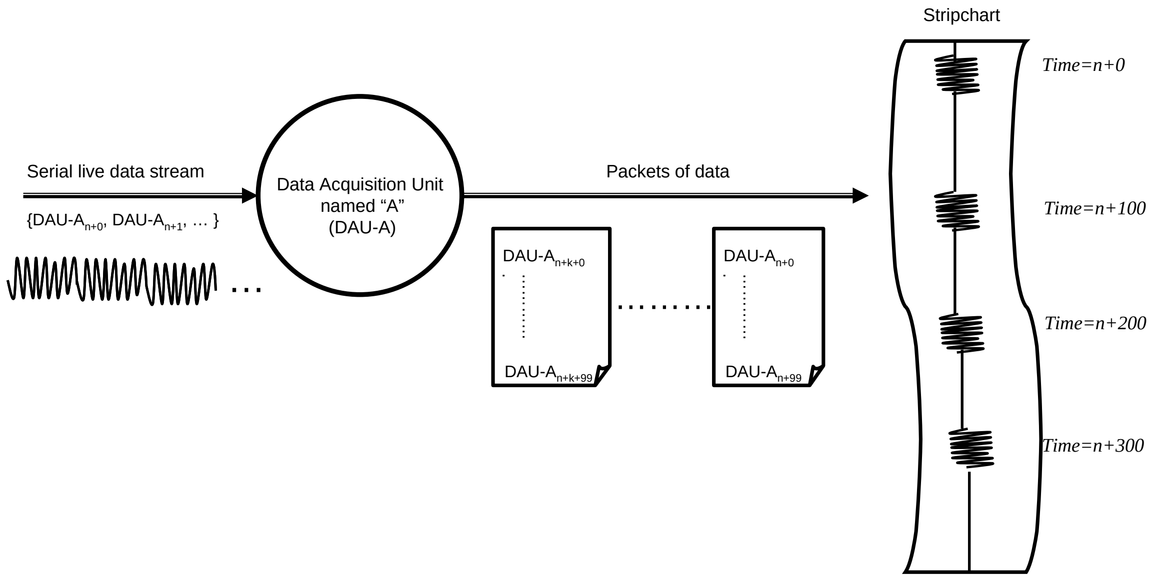

Figure 2. IP network packets will generate time gaps if elements are not retimed.

HOW WE RETIME

Since data is from one source, converting it back from packets into a stream simply requires receiving the packets of data, queuing data elements based on their original time, and then at some constant delay time after the original data time of each element, forwarding on the data elements for real-time display and analysis. This delay means that data cannot really be live, rather, it must incur the maximum time that is used to form larger packets. For this reason, some very low rate data acquisition units may send small packets even though they are less efficient. In order to know time of individual data elements, we include the time of the first element in the IP packet as header information. Then, for known data acquisition unit rates, we can calculate the time of each of the individual data elements. Figure 3 shows a form of the queue we use to retime the data. This is basically a priority queue with time being the priority [1]. Time is used to release items from the queue for downstream processing (e.g strip charts).

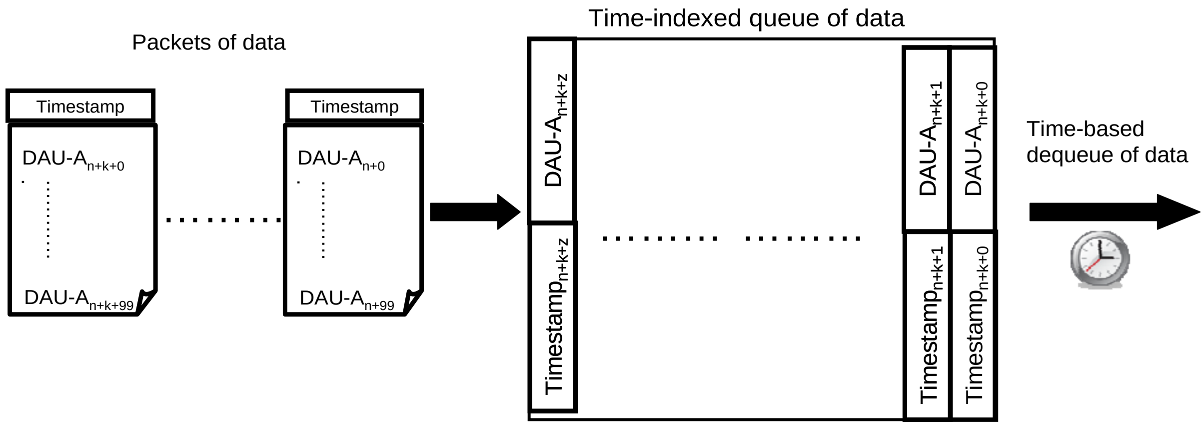

Figure 3. Elements IP network packets are inserted into a time based queue for retiming.

HOW WE REORDER

Additionally, we have developed methods and algorithms for the filtering, selecting, and retiming problems associated with our packet based approach. Our IP packets are sent with IP multicast thereby allowing multiple clients to subscribe to their flows. Our filtering and selection techniques use a one pass rule based technique to quickly pull items. Our retiming technique uses time-based automatic garbage collection along with a time-based hashing function to support the insertion sorting concept required by a priority queue. This sorting is required since data elements from various sources can overlap in time due to the packet creation and fill delay. Figure 4 shows an example of our reordering approach with three data sources.

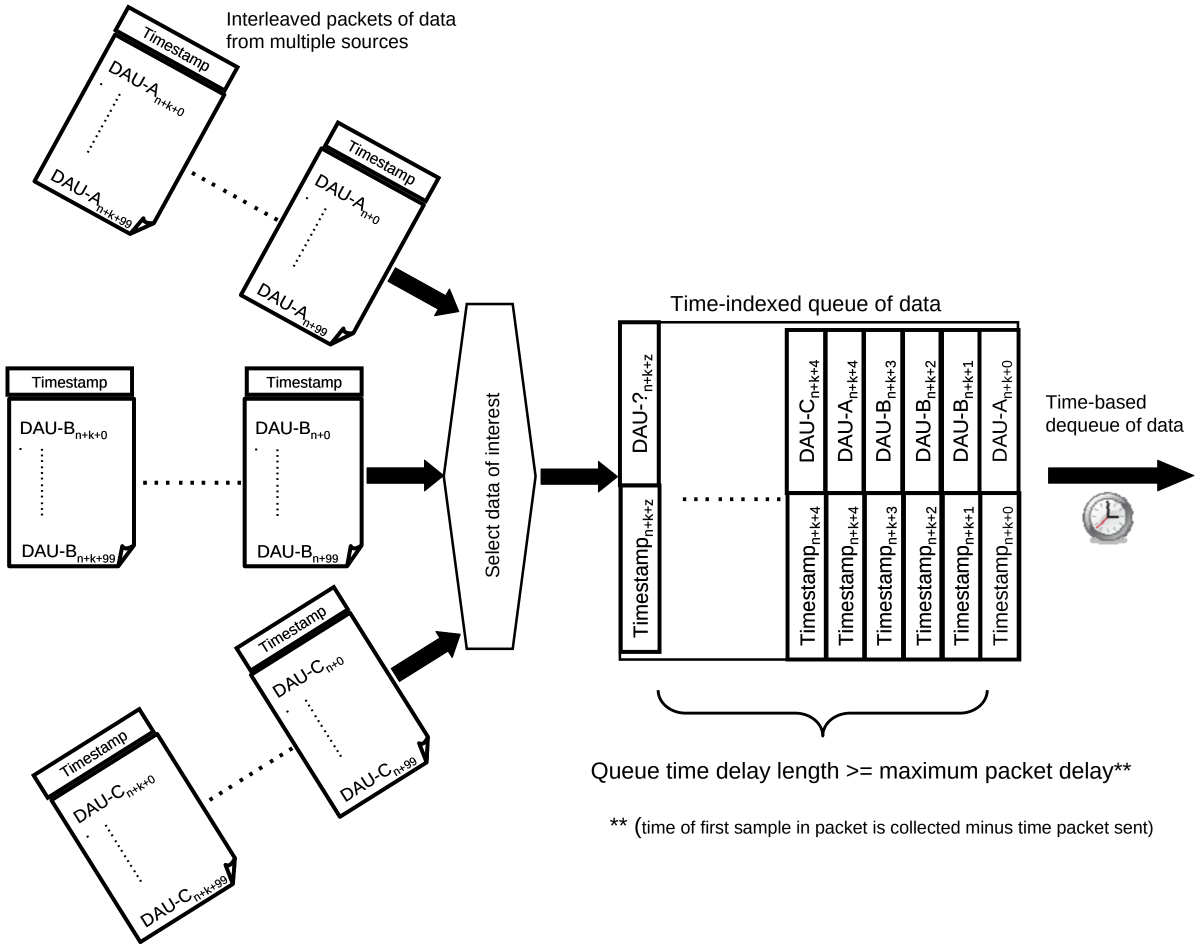

Figure 4. Reordering of data elements since packets can supply them out of order.

CONCLUSION

In conclusion we see that the current state of the art of IP networks provides an excellent framework for the development of real-time data acquisition and telemetry applications. The approach does introduce some problems that must be dealt with, such as packet ordering and timing, but there are methods to deal with these problems.

ACKNOWLEDGEMENTS

This paper presents issues associated with the implementation of real-time data acquisition systems using IP-based networks, and a high-level overview of approaches to address two key issues. The issues and general approaches described herein are the result of collaboration among people in various organizations beyond the authors listed.

REFERENCES

[1] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. Introduction to Algorithms, Second Edition. MIT Press and McGraw-Hill, 2001. ISBN 0-262-03293-7. Section 6.5: Priority queues, pp.138–142.