Looking Inside Real-Time Ethernet

By **Volker Goller**

Industrial Ethernet—real-time Ethernet—has experienced a huge upswing over the last few years. Although classic fieldbuses are still running in large numbers, they have passed their prime. The popular real-time Ethernet protocols extended the Ethernet standard to meet the requirements for real-time capabilities. TSN now provides a new route to real-time Ethernet.

Real Time and Communications

In the context of factory automation and drive technology, real time means safely and reliably reaching cycle times in the range of less than ten milliseconds down to microseconds. For these real-time requirements to be satisfied, Ethernet also had to gain real-time capabilities.

Ethernet Is a Lot Faster Than a Fieldbus— So What?

For the real-time requirements of automation to be satisfied, both transmission bandwidth and transmission latency need to be guaranteed. Even if these bandwidths are usually decidedly small (a few dozen bytes per device), this transmission channel must be available in every I/O cycle with the required latency.

However, the guarantee of latency and bandwidth is not provided with classic Ethernet. On the contrary, an Ethernet network may discard frames at any time if this is necessary for operation. What does this mean?

Ethernet is a so-called bridged network. The frames (Ethernet frames) are sent from one point to another: From the endpoint to the switch (bridge), from there possibly to other bridges, and finally to the other endpoint. This architecture is largely self-configuring. The bridges first completely receive frames before forwarding them. And this is where multiple problems arise:

▶ If at peak times there are more frames to store than the buffer memory in the bridge can hold, then the newly incoming frames are discarded.

▶ Because the frames differ in length, they are delayed as a function of their lengths. This leads to fluctuating latencies (jitter).

▶ Because the port through which the switch is supposed to send a frame may already be occupied by other frames up to the full frame size, additional delays come into play. The sending of a large Ethernet frame (1522 bytes) takes about 124 µs at 100 Mbps.

We can argue that Ethernet normally works well and is somehow fair. However, by doing so, we use two words that make no sense in connection with hard real time. It is not enough if a real-time condition is only normally met. It always has to be met.

Anyone who lives next to a chemical plant or a refinery can appreciate this. And industrial communications aren't fair either: The most important thing, the control/closed-loop control application, always has priority.
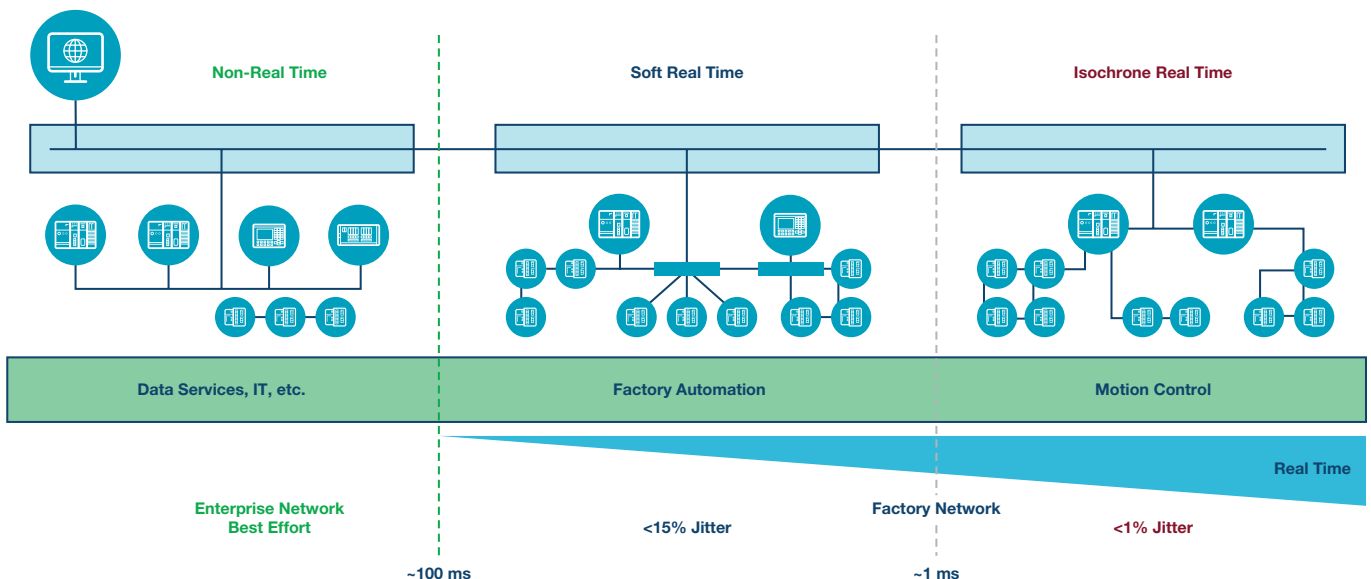

Figure 1. Real-time communications in automation.

Figure 2. ISO seven-layer model.

Real-Time Extensions Illustrated Using PROFINET and EtherCAT as Examples

Because no solution to the problem was available in the IEEE, which is responsible for Ethernet standardization, the industry developed its own solutions—once again demonstrating its innovativeness. The solutions all have their strengths and weaknesses and ultimately address different markets.

PROFINET: Universally Applicable

With PROFINET, two complementary solutions are offered. PROFINET RT is a factory automation solution with a cycle time of up to 1 ms. RT is directly based on standard Ethernet. The possibilities of Ethernet (for example, quality of service (QoS, prioritization)) are utilized to give real-time traffic priority. That helps, but QoS does not completely solve the resource and latency problem. That's the reason for restriction to soft real time. The good compatibility with other protocols (such as HTTP, SNMP, and TCP/IP) used in the network is a clear strength of the technology.
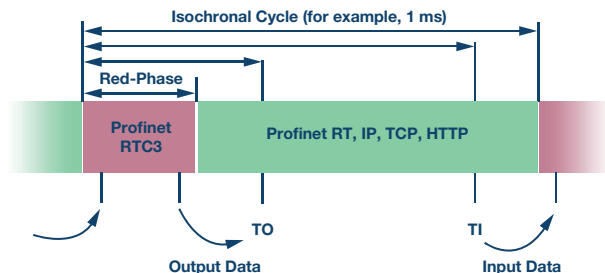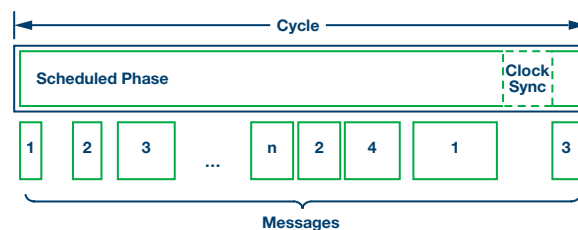

Figure 4. PROFINET IRT.

For hard real time, PROFINET offers the isochronous real-time (IRT) extension. Here, part of the Ethernet bandwidth is reserved exclusively for IRT traffic through an extension to standard Ethernet hardware. This is made possible by precise synchronization of the clocks in the IRT nodes. As a result, a channel (the red phase), can be blocked for normal traffic in every cycle. Only IRT frames in the red phase reach the network. In addition, the network participants send the IRT frames exactly at precalculated times, enabling maximization of efficiency within the red phase. The IRT frames move almost without slip through the network. One advantage of this is that it limits the length of the red phase, in which all other traffic must wait, to the bare minimum. The red phase can occupy up to 50% of the bandwidth of the Ethernet channel.

Figure 3. Overview of protocols.

As already mentioned, a full length (1552 byte) Ethernet frame needs about 124 µs on the wire. If PROFINET IRT occupies max. 50% of 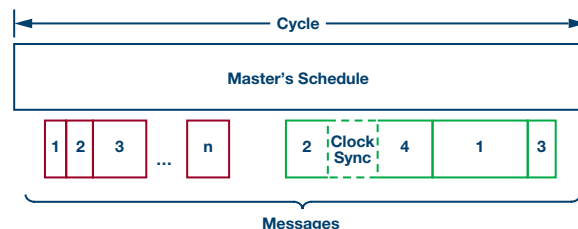the bandwidth, the fastest cycle time is 2 × 124 µs = 248 µs, or 250 µs when rounded up. Only in this way can other protocols (like HTTP) coexist in unchanged form with it.

Even faster cycle times of down to 31.25 µs are possible due to PROFINET 2.3 for IRT's optimizations, including fast forwarding, dynamic frame packaging, and fragmentation.

EtherCAT: The Ethernet Fieldbus

In the development of EtherCAT, there were other requirements in the beginning. EtherCAT is a fieldbus based on the physical Ethernet—that is, layer 1. Even layer 2 is optimized for fieldbus applications and high throughput. EtherCAT doesn't have the classic Ethernet bridge. It uses a summation frame telegram, which makes data transmission especially efficient. Instead of normal Ethernet, in which a separate frame is sent by each device involved in communication between devices, EtherCAT sends one frame per cycle. However, this frame contains all data for the addressed devices. While the EtherCAT frame is being forwarded by a device, the data for that particular device are inserted into and taken out of the frame live. Through this, very short cycle times of even less than 31.25 µs, in the extreme case, can be achieved.

EtherCAT also has time synchronization. A lot of effort was put into enabling the not always ideal Ethernet interfaces on a PC to be used as masters for EtherCAT.

Ethernet traffic such as web or TCP/IP traffic can only be transported in small portions in piggyback style over EtherCAT; direct coexistence on the wire is not possible.

What About the Others?

POWERLINK takes the same basic approach that EtherCAT does; it assumes complete control over the Ethernet and transports IP applications by piggyback to the nodes. But that's the only thing they have in common. POWERLINK does not employ a summation frame protocol. Nevertheless, it performs similarly well in practical applications.

Like IRT, SERCOS has a reserved bandwidth, but uses a summation frame protocol within it. SERCOS allows other protocols to coexist.

It's Time for TSN

IEEE approached the topic of real time within the scope of the audio/video bridging (AVB) protocol. In the improvement of the protocol, the more challenging real-time communications of industry were also considered. The original name for the set of standards, AVB2, was hereupon changed to TSN (for time-sensitive networking). With these standards, it is now possible to use a unified, deterministic version of Ethernet.

This actually makes many things easier. For example, the well-known industrial networks are nearly all defined for 100 Mbps. Today, however, gigabit Ethernet and 10 Mbps Ethernet have become the focus of attention in special applications. The TSN standards cover all speeds. With TSN, the wheel does not have to be reinvented: if it weren't for TSN, all of the existing standards would have to be redefined for gigabit—which would result in costs for hardware development and in market fragmentation.

How Does TSN Help? Real Time with TSN

TSN extends layer 2 of Ethernet to include a series of mechanisms needed for real-time operation:

▶ 802.1AS/802.1AS-Rev provides for extremely precise synchronization of the clocks in the network.

▶ The time-aware shaper (TAS) option enables Ethernet to be operated with hard scheduling. With it, one or more queues of the QoS model can be blocked/released at specific times.

▶ The preemption (interspersing express traffic) option enables long frames to be broken up into smaller parts so that delays are minimized for higher priority frames. It can be used to optimize the guard band for TAS or replace TAS at speeds of above 100 Mbps.

▶ The frame replication and elimination for reliability option can be used for the definition of redundant paths through the network; for example, in rings.

▶ Use of software-defined networking means that frames are no longer forwarded to the destination by means of the hardware MAC address of the destination node, but are rather forwarded through a combination of special MAC addresses (locally administered multicast MAC) and VLAN IDs. How these frames are routed through the network is no longer automatically determined, but rather configured by software. This combination of multicast MAC and VLAN ID is called the stream ID and all TSN frames with the same stream ID are called the TSN stream. A TSN stream always has just one sender, but it can have several recipients.
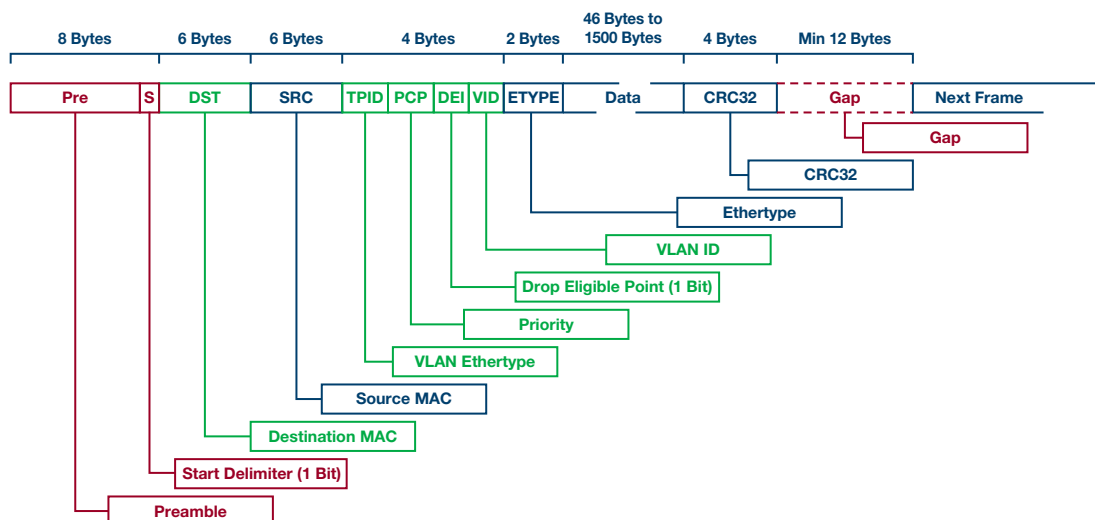

Figure 5. An Ethernet frame, with parts relevant to TSN data stream identification shown in green.

The TSN streams can now be set up in consideration of the existing resources in such a way that no frame has to be discarded anymore. The bridges now use their resources for loss-free forwarding of the TSN streams.

The best effort traffic (standard Ethernet, IP, web) takes place completely normally with the remaining resources (memory/bandwidth).

And What Happens Above Layer 2?

Behind every internet Ethernet protocol lies an organization that advances the standardization and popularization of the respective protocol. Each of these organizations has formulated a TSN strategy. Consequently, we will see nearly all existing protocols again with TSN—in one form or another. Staying with our examples:

For PROFINET, the path to TSN is a relatively short one because there is already a wealth of experience with time-aware shaping available (it is already done very similarly to IRT) and the coexistence of industry and IT protocols has always been supported. Much remains the same for the user, so a familiar environment can yield new performance.

EtherCAT and, similarly, SERCOS will make TSN accessible above the field level. For example, the EtherCAT automation protocol (EAP) is very suitable for networking classic EtherCAT segments via TSN at a low overhead.

However, there are also new players in the field.

There is a group that is defining a completely new industrial Ethernet protocol. OPC UA is being used as the application layer. TSN is seen as a means for making this protocol real-time capable. However, work still has to be done here. A new transport layer for OPC UA, the so-called OPC UA PUB/SUB protocol, is needed for the transport.
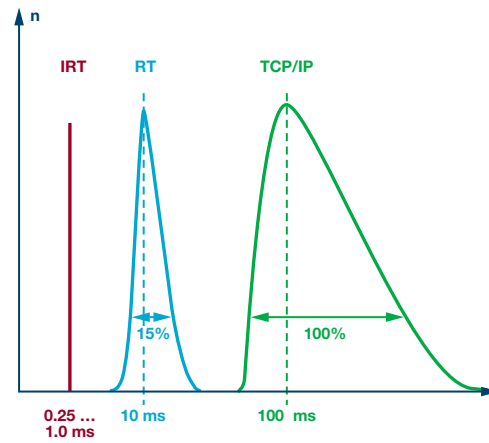

Figure 6. Magnitude of latency/jitter for hard real time (IRT), soft real time (RT), and IT protocols (TCP/IP).

Does a Lot Help a Lot? Not in Real Time

Today we are using 100 Mbps Ethernet in industrial automation and soon gigabit Ethernet will be available. However, higher speed does not automatically mean guaranteed latency and guaranteed transmission. Hence, for hard real time, special mechanisms are always necessary. With TSN, they are now standardized.

Volker Goller

Volker E. Goller [volker.goller@analog.com] is a systems applications engineer with Analog Devices and has over 30 years of experience with a diverse set of industrial application ranging from complex motion control and embedded sensors to time-sensitive networking technology. A software developer by trade, Volker has developed a wide variety of communication protocols and stacks for wireless and wired applications while actively engaging in fielding new communication standards through his involvement in leading industry organizations.