

Comparing CAN- and Ethernet-based Communication

By Prof. Dr.-Ing. Konrad Etschberger

Preliminary Remark

A direct comparison between CAN and Ethernet is surely not possible, because Ethernet does not provide a secure communication: An Ethernet receiver only checks if a received frame is correct. If not, it simply discards the frame, there is no information of the transmitter and therefore no re-transmission initiated. To provide this, we need an additional higher protocol layer like TCP/IP to get a response or confirmation message for informing a transmitter that its frame has been discarded. Therefore, when comparing performance data we have to take TCP/IP into consideration. This implies the additional protocol overhead as well as the implied processing time.

On the other hand a CAN protocol chip by means of the implied error signalling feature of the CAN protocol already provides a highly secure transfer of data without any additional acknowledgment by a higher protocol layer.

In the following, some features and aspects of the two technologies are summarized:

Frames per Second (FPS)

FPS is a common method of rating the throughput performance of a network. This is especially relevant for applications where the payload is only few bytes. This is the case for most of so-called monitoring and control systems.

- **Standard Ethernet**

Total Frame Length: MAC Preamble (7 Bytes), Start Frame Delimiter (1 Byte), Destination MAC Address (6 Bytes), Source MAC Address (6 Bytes), IP-Header (20 Bytes), TCP-Header (20 Bytes), Padding Bytes (if Payload is less than 46 Bytes)(6 Bytes), Frame Check Sequence (4 Bytes), Interframe Gap (96 Bits) (12 Bytes): Total Minimum Frame length (1 .. 6 Data Bytes Payload) = 84 Bytes, 7 Data Bytes: 85 Bytes; 8 Data Bytes: 86 Bytes

With Standard Ethernet we have to consider an increased percentage of bus collisions when the bus load is higher than 20 % (at about 50 % bus load there are only bus collisions). Therefore only about 20 % of the bandwidth actually is available.

10 Mbps Ethernet

Transmission of 1- 6 Data Bytes:

Maximum number of frames per second =

(20 % of 10.000.000 bits/s) : $84 \cdot 8 \text{ bits/Frame} = 0,2 \cdot 14.881 = 2.976 \text{ Frames/s}$

Transmission of 8 Data Bytes:

Maximum number of frames per second =

(20 % of 10.000.000 bits/s) : $86 \cdot 8 \text{ bits/Frame} = 0,2 \cdot 14.535 = 2.902 \text{ Frames/s}$

With **100 Mbps Ethernet** the maximum number of frames per second is about 29.000 Frames/s

- **CAN**

Total Frame Length: SOF (1 Bit), Identifier+ RTR(12 Bit), Data Length Code (6 Bits), Data Field (0 ... 64 Bits), CRC (16 Bits), ACK-Field (2 Bits), EOF (7 Bits), Interframe Space (3 Bits); Stuff Bits (3 Bits)¹: Total Frame Length: 58 Bits (1 Data Byte Payload) ... 114 Bits (8 Data Bytes Payload)

1 Mbps CAN

With CAN we can load a system theoretically up to a bus load of 100 % without no fear of collision; this is possible also practically if we have frames which are not very time critical. If there is a higher percentage of frames for which no longer delay are acceptable we also should reduce the maximum bus load. In the following 100 % bus load is assumed.

1 Data Byte

Maximum number of frames per second = $1.000.000 \text{ bits/s} : 58 \text{ bits/Frame} = 17.241 \text{ Frames/s}$

8 Data Bytes

Maximum number of frames per second = $1.000.000 \text{ bits/s} : 114 \text{ bits/Frame} = 8.772 \text{ Frames/s}$

Result

1 Mbps CAN provides a much higher frame rate than 10 Mbps Standard Ethernet/TCP/IP. For transmission of 8 Byte data payload at 100 % bus load the frame rate of CAN is about 8.700 FPS, for 10 Mbps Standard Ethernet and 20 % bus load we can transmit up to only about 2.900 FPS. We then already have to accept bus collisions.

In spite of a 100 times higher bit rate, 100 Mbps Standard Ethernet (20 % bus load) only provides about 3 times the maximum CAN frame rate.

The reason of the relative low maximum frame rate of Ethernet is the very low protocol efficiency of Ethernet/TCP/IP² at only few bytes payload together with the limited allowable bus load compared to CAN. E.g. for transmission of a payload of 1 Byte the protocol efficiency of Ethernet is only 1,2 %, for CAN this is 14,5 %, for transmission of a payload of 8 Byte the protocol efficiency of Ethernet is 9,3 %, for CAN 57,6 %.

When do we need a frame above of 10.000 FPS at all?

Imagine a system consisting of 32 nodes. Then, if all nodes produce the same amount of frames per second, each node should produce at a rate of about 300 frames per second or produce a message every 3 ms. Therefore 10 000 frames per second already is a rather high message rate. Most applications will have much lower messages rates.

Further Aspects for comparing CAN and Ethernet/TCP/IP

There are several important advantages of CAN compared to Ethernet/TCP/IP which makes CAN especially interesting for many application – of course when its message rate is sufficient:

¹ According to a corresponding study only up to 3 stuff bits are included when statistical data is transmitted

² For Ethernet-based communication we need TCP to provide secure data transmission (Ethernet does not provide this). With CAN secure transmission is already provide by the CAN protocol.

- **Collision-free bus arbitration**

Standard Ethernet can suffer bus arbitration collisions due to the underlying CSMA/CD bus arbitration scheme. Therefore, to limit the amount of bus collisions we can use Standard Ethernet only to a limited bus load, or we have to use additional protocols applied above of the Ethernet protocol (like EthernetPowerLink or ProfiNet). Additional protocols (e.g. time deterministic protocols) of course cause additional software overhead and specific hardware feature.

- **Multi- and Broadcasting**

Even if it is also possible with Ethernet to send broadcast messages, these messages are not reliable. This means that it is not guaranteed that all nodes of a network get a specific broadcasting frame. For some applications, like robotic control this can be a serious problem. With CAN, any message is a broadcast frame and it is assured by the protocol that the transmitted data is consist all over the network due to the inherent error signalling mechanism.

- **Response time limited by protocol software processing time**

In Spite of the high overall frame rate with 100 Mps Ethernet, the maximum frame rate accepted by a node is limited by the time necessary for processing of a frame (TCP/IP protocol processing time). If we also consider, that a TCP/IP message needs to be confirmed (there is a request and a response frame) and we neglect the frame transmission time, the time for processing the IP/TCP frame is in the magnitude of almost 1 ms for a common microcontroller. This means that the maximum frame rate of a node is limited to about 1000 frames per second. Or, expressed in terms of response time, the response time even for a 100 Mbps network is limited to about 1 ms (if we do not consider specific protocols like PowerLink).

- **Data Integrity**

Due to the different data securing mechanism (especially bit monitoring and CRC) and the short message length of CAN frames, CAN provides an unexcelled level of data integrity. The remaining error probability is given by:

$$\text{Remaining error probability} = \text{frame error rate} * 10^{-13}$$

This means that transmission errors will be detected with a very high probability by the integrated error detection mechanism of the CAN protocol. This capability is essential for any application which involves high risks.

Higher Communication Layers and Application Profiles

Ethernet/TCP/IP only provides services necessary for the confirmed transmission of frames between nodes. Beside of this, for a standardized application further communication protocols and features are necessary. The CAN-based CANopen and DeviceNet standards provides these features, especially standards for describing a device or for initializing and monitoring of a running network. Both standards also provide a variety of standardized device and application profiles.

Therefore Ethernet Powerlink also provides additional higher communication layers and profiles based on CANopen. The same approach is made by Ethernet/IP which uses the CIP (Control and Information Protocol) as a common communication profile for CAN-based DeviceNet and Ethernet/IP.

Both solutions therefore provide easy integration and migration of CAN- and Ethernet-based systems.