STANDARD FIELD BUS NETWORKS FOR INDUSTRIAL APPLICATIONS*

Gianluca Cena†, *Luca Durante*† and *Adriano Valenzano*‡

‡ Centro per l'Elaborazione Numerale dei Segnali
† Dip. di Automatica e Informatica
Politecnico di Torino
Corso Duca degli Abruzzi, 24 - 10129 Torino - Italy

Preferred address for correspondence:
Luca Durante
Dip. di Automatica e Informatica
Politecnico di Torino
Corso Duca degli Abruzzi, 24
I - 10129 Torino TO
Italy
E-mail: durante@polito.it

*This work was partially supported by UNINFO and CCIAA of Turin.

Abstract

This paper deals with three emerging proposals for standardizing industrial field bus networks. Even though significant work is being done by some international organizations to develop suitable field bus standards, a wide consensus is still to be reached on what features this kind of subnetworks must exhibit and on the protocol mechanisms which are necessary to support them. The aim of this paper is to present the solutions introduced in FIP, PROFIBUS and SERCOS, network architectures which have already been adopted as national standards by some European countries. These field buses are compared briefly and some formulas are introduced in order to offer the reader a preliminary evaluation of the performance which can be expected from this kind of systems.

Keywords

Communication Protocols, Field-buses, Industrial Communications, Network Standards, Real-time Communications.

Introduction

Interest in industrial computer networks has significantly increased in the last decade due to networks being considered a primary way to simplify the transfer of information so that the degree of integration needed by computer integrated manufacturing (CIM) systems can be achieved.

The demand for communication capabilities in the industrial field has led private and public organizations together with standardization bodies to spend a lot of time, resources and efforts in developing suitable communication standards able to satisfy many requirements typical of process control and manufacturing automation environments. At present, a widely accepted model [1] organizes an industrial communication system as a hierarchy of three types of networks, each one having different goals and also different communication capabilities, protocols and complexity. Type one networks are to be used to interconnect machines which perform tasks such as manufacturing engineering, production management, resource allocation and so on, while type two are designed to be used with cell controllers, milling, inspection and control workstations in manufacturing plants. At the lowest level in the hierarchy, type three networks also called "field buses" are used to connect equipment controllers, sensors, actuators and less intelligent devices. Field buses are certainly less complex than type one and type two networks, however they must tackle a number of aspects not shared with the other kinds of communication systems. In particular field buses must support real-time data exchanges and offer services for periodic and aperiodic data reading and writing. Low-cost interconnection techniques must be used since cheap devices (i. e. sensors and actuators) can also be connected directly to the bus. In addition it should be possible to integrate a type three subnetwork in the manufacturing environment and to deliver messages between subnetworks in the same plant.

In the last few years suitable international standards such as MAP [6] and CNMA [7] have been developed and adopted for type one and type two networks which are mainly based on the ISO/OSI services and protocols. The standardization process of field buses began about ten years ago with the proposal of the International Electrotechnical Commission (IEC) and the Instrument Society of America (ISA): the IEC/ISA SP50 Fieldbus. Actually only the physical and the data link layers are completely defined. Due to the lack of a well defined and stable application layer, manufacturers do not still adhere to IEC/ISA SP50: they usually rely on more consolidated national standards as WorldFIP and ISP that introduce some minor change to the Frech standard FIP and to the German standard PROFIBUS respectively. FIP and PROFIBUS have significantly affected the standardization activity of the IEC/ISA SP50 Fieldbus in the last few years.

The aim of this paper is to introduce three field bus architectures which have been adopted as national standards in Europe and are being examined by international standardization committees so that they can eventually be included as parts of international standards.

For this reason we will consider the above cited Field Instrumentation Protocol (FIP) and PROcess FIeldBUS (PROFIBUS).

SERCOS [2] is another German standard considered in this paper: SERCOS is particularly oriented to numerical control applications and is not so flexible or powerful as FIP and PROFIBUS, however it must be taken into account because it has been submitted to IEC and it has been considered for a possible standardization [3].

The paper is organized as follows: section 1 contains a short presentation of the three field bus architectures while section 2 deals with the main characteristics of the protocols pointing out strengths and weaknesses of each proposal. Finally, in section 3 some performance considerations are introduced and performance figures are obtained assuming a network configuration which allows FIP, PROFIBUS and SERCOS to be compared.

1 Field bus architectures

1.1 Field Instrumentation Protocol (FIP)

FIP is a collection of French national standards [5] promoted by the French national standard organization (AFNOR). Each FIP network is based on a shared-bus topology while the transmission medium adopted can be either a shielded twisted pair or fiber optics. The use of signal repeaters is foreseen so that networks with a tree shaped architecture can also be built. Transmissions adopt the conventional Manchester encoding, with the addition of violation bits. Bit synchronism is granted by a special message preamble, while frame synchronism is achieved by means of a pair of delimiter fields. Tab. 1 summarizes some characteristics of the FIP network.

The FIP medium access technique requires that a single station called the bus arbiter be responsible for managing the right to transmit over the shared bus. Each network segment is controlled by its own arbiter. The arbiter sends request frames to the other stations which, in turn, are only allowed to return reply frames.

Each request/reply transaction builds up an elementary sequence. The FIP specifications include sequences for exchanging variables and messages: services for exchanging variables are in fact a peculiar characteristic of FIP.

FIP stations can produce and/or consume variables. A station can be a producer, a consumer or a producer and a consumer at the same time. An elementary transaction for transferring a variable value consists of a two-phase protocol as shown in Fig. 1. During the first phase the variable identifier, which is known to the variable producer and to all the consumers after network configuration, is broadcast on the network by the bus arbiter. In the second phase the producer replies to the bus arbiter by putting the current value of the variable on the bus, while at the same time all the consumers read that value from the medium and thus the transaction is completed.

A number of elementary sequences constitutes an elementary cycle. Each elementary cycle contains four timing windows devoted to transmitting cyclic and acyclic variables, to exchanging messages and to synchronizing the stations in the network respectively.

A FIP macrocycle is constituted by a sequence of elementary cycles and it defines the sequence of periodic inquiries the bus arbiter must send on the network. The structure of the macrocycle is determined off-line, when the network is configured and cannot be changed dynamically.

Messages are also exchanged by means of elementary transactions driven by the bus arbiter in a way which is very similar to the one used for variables, in fact the message sender (producer) can start its transmission when the bus arbiter broadcasts an identifier used to distinguish the specific message to be sent on the network. Receivers can detect the destination of the message by inspecting an address field which is contained in the transmitted frame.

It is worth noting that, in this way, FIP has to handle two kinds of addresses, that is to say variable addresses which have a global meaning, largely independent of the variable physical locations, and station addresses which are used to deliver the messages. When a network is made up of several segments, variables cannot be shared among segments (i.e. variables are local to a single segment) while messages can be exchanged between different segments. This allows information to be transferred between FIP segments and between FIP and other subnetworks.

1.2 Process Fieldbus (PROFIBUS)

PROFIBUS is a German national standard promoted by the German standard organization (DIN) [4] which is able to interconnect both low cost devices, such as sensors and actuators, and more powerful machines such as PLCs and NCs. Some basic characteristics of PROFIBUS are summarized in Tab. 1.

The network architecture is based on a reduced protocol profile and is quite similar to miniMAP [6]. The network topology is, therefore, based on a common bus and the access technique adopted is a slightly different version of the token-bus mechanism. Bits are transmitted on a shielded twisted pair using an NRZ encoding scheme combined with an EIA RS-485 signaling method. A simple bit serial asynchronous transmission protocol has been adopted so that commercially available components can be used.

A PROFIBUS network can include two kinds of stations: masters and slaves. Masters take part in the token circulation while slaves play a passive role and can only be polled by the master stations. This hierarchy is typical in field control systems, where sensors and actuators are slave stations while PLCs and NCs are master nodes.

The PROFIBUS medium access control includes the services foreseen by the IEEE 802.4 MAC standard [8], however an additional service has been introduced called cyclic send and request data with reply (CSRD) to deal with cyclical polling of the slave stations. In this way a number of slaves can be specified in a list and then are polled automatically by a master at each token reception without any user intervention.

1.3 SERCOS

SERCOS is another standard proposal [2] developed by the German association of machine tool manufacturers and has been submitted for standardization to IEC [3]. The SERCOS documents specify a field bus protocol to be used in digital control applications based on NCs. A SERCOS network consists of a fiber optic ring connecting one “master” station with one or more “slaves”; each slave, in turn, can be connected to several drives. The main SERCOS characteristics are also shown in Tab. 1.

Direct exchange of information takes place only between the master and the slaves but is not allowed between the slaves. Information consists of data and commands. Data transmissions are strictly cyclic and totally driven by the master. Each cycle starts with a master sync telegram (MST) which is sent to all the slaves: each slave uses this message to achieve synchronism and to detect its own transmission window (time slot) by starting an appropriate timer. Synchronization between the master and the slaves is maintained by means of an NRZI encoding scheme and bit stuffing.

During its time slot the slave returns amplifier telegrams (AT) that contain data for the master as shown in Fig. 2; the master collects all the data coming with the answers of the slaves and, at the end of the cycle, it is able to send a single master data telegram (MDT) containing the sequence of current data values for the slaves. Each slave which receives the MDT is able to identify and extract only the subset of data it needs from the MDT frame. A single slave returns to the master one AT for each controlled drive.

2 Discussion of field bus characteristics

FIP, PROFIBUS and SERCOS have their own characteristics and peculiarities because of the substantial differences that can be found in their architectures. In this section the three standards are briefly discussed, taking into account the main requirements of an industrial field bus such as the real-time support, the degree of fault tolerance and reliability and the set of services offered to the application processes.

2.1 Response time

A crucial requirement in field bus applications is a bounded response time. In other words the user must be guaranteed that the communication system will satisfy a data transfer request within a maximum period of time (possibly very short). In our case this lead to minimizing the maximum delay which elapses between the service request submitted to the network by an initiating user on a station and the reception of the corresponding message by the destination node.

The SERCOS protocol is deterministic by nature so that all the transfer times are deterministic too. Since the same transmission cycle is periodically repeated by the network, delays experienced in exchanging a single datum is always bounded to the cycle time.

The behaviour of a FIP network is similar, but in this case not all the variables are exchanged during the same elementary cycles. In general the cycle time of a variable falls between the durations of the elementary cycle and the macrocycle, so that the maximum response time for transferring any variable value is bounded to the macrocycle length.

In both SERCOS and FIP no confirmation is returned to the sender of a variable value to indicate the success or the failure of the transmission. If an error occurs during a variable exchange, the receiver does not get the right value until the next cycle. FIP users, however, can request new values for the corrupted variables by using the window devoted to aperiodic data transfers in the FIP elementary cycle. The responsibility of such an error recovery action is left totally to the user.

PROFIBUS, just as the well-known token bus technique, grants an upper bound to the time needed to access the channel and hence to the time required to send a high priority message containing data values. PROFIBUS allows a master station to send a (high priority) frame each time it gets the token. The transmission of low priority messages and the polling of slaves can be carried out only if the measured real token rotation time is lower that an established target token rotation time. In this way the maximum delay that high priority frames can experience is upper bounded, and the bound depends heavily on the target token rotation time [11].

Finally it is worth noting that the acyclic data transfer requests in FIP and SERCOS and the delivery requests of low priority frames in PROFIBUS are queued on the transmitter side and are carried out by the network only when the urgent traffic allows it to do so, so that a bounded response time is not granted for this kind of services.

2.2 Efficiency

Field buses are particularly oriented to those applications which require a periodic exchange of small sized data even though the exchange rate can be significantly high. In the following the term efficiency is used to mean the ratio between the amount of useful user data (such as variable values) moved through the network during a time cycle and the number of bits needed to transmit such an information.

Field bus efficiency is largely affected by the mechanisms introduced in the medium access control (MAC) sublayer. Because of the different medium access techniques adopted, each network exhibits better efficiency in some particular condition.

FIP, for instance, is very efficient when each produced variable is consumed by several stations. In this way a single elementary sequence is needed to supply all the consumers of a variable with the current value simultaneously. As mentioned in section 1.1 variables can have different cycle times. This allows the network load to be kept as small as possible when some variables have to be updated at a rate slower than the elementary cycle.

In PROFIBUS, when a master-slave cyclic connection is established, a real data transfer on the network takes place only when new data values become available on the producer side. Consumers (that is, receivers) have a local copy of each datum involved in the transfer stored in a local buffer. This copy is updated whenever a new value is received from the producer. If the value on the producer side remains unchanged the producer (slave) replies to a poll command from the consumer (master) with a very short acknowledgment since no new datum has to be returned in this case. This mechanism is provided to save system bandwidth so that the transmission of redundant information can be avoided and thus messages traveling on the network can be kept as short as possible.

The efficiency of SERCOS relies on the timed medium access. Each slave is assigned a predefined slot of time to return data to the master, so a single master inquiry (that is the MST message) can be used to poll all the slaves. Data transferred from the master to the slaves is packed into a single frame which requires a single transmission thus avoiding the need to replicate the message control bits several times. Obviously better performance is reached when all the drives attached to the slave stations need data at the same refresh rate.

2.3 Data exchange priority

FIP cyclic data exchanges are performed at the highest priority, that is to say, cyclic data transfers are always carried out in a deterministic way. Acyclic transfer requests are assigned to two priority levels, that is to say urgent and normal. Urgent requests are honored first by the bus arbiter that enforces the priority mechanism with a suitable scheduling of the servicing requests.

PROFIBUS supports a two level priority scheme, while no priority mechanism has been included in SERCOS. Each PROFIBUS master station is allowed to send at least one high priority frame when it gets the token, moreover high priority frames always have precedence over the low priority transmissions. The high priority can be used only with the conventional acyclic connections, so that the cyclic polling of the slave stations could be delayed when the arrival rate of urgent frames is significantly high. The standard recommends that only very important events make use of the high priority transmission services.

2.4 Fault tolerance

Operations in a FIP environment rely entirely on the availability of the bus arbiter. If the active bus arbiter becomes faulty, a method based on timeout lets other stations configured as potential bus arbiters to activate a new arbiter election mechanism. FIP station faults do not prevent the whole network from working, even though some produced variables can become unavailable.

SERCOS subnetworks also rely on the master correct operation but, because of the simplified topology adopted, no recovery mechanism has been included in the specifications. A slave failure, in general, does not affect the other master-slave connections if the faulty slave maintains the capability of repeating bits received from the network. The PROFIBUS token passing mechanism is safe against token losses, corruptions and/or duplications. Stations are free to enter or leave the logical ring at each time and the fault-tolerance characteristics are the same as any that can be found in any IEEE 802.4 network.

2.5 Data-link services

The FIP and the PROFIBUS field buses are based on a three layer protocol stack which includes the physical, data-link and application layer of the OSI reference model, thus communication services are specified at the upper interface of each OSI level. The SERCOS specification is not based on a layered model, thus service definitions are not included in the standard document.

Both the FIP and the PROFIBUS data-links offer services to send messages in a connectionless fashion, with or without acknowledgment of the correct reception by the receiver, according to the PROWAY C specification [12]. A transmission error occurring during an acknowledged transmission is detected when the receiver's acknowledgment does not arrive within a predefined period of time; in this case a number of retries are carried out by the transmitting station without releasing the bus mastership.

PROFIBUS also defines a service to ask for a reply containing data from the receiving station. Data values returned in this case must be previously made available to the responding station by invoking a reply update service.

FIP supports a specific set of services aimed at managing the network distributed variable database. In this case exchanged objects are not messages but variables which have a global network identifier. Services are provided to read/write the local value of a consumed/produced variable and to invoke an acyclic update of the variable value.

2.6 Application services

Both the FIP and the PROFIBUS standards offer a subset of services which are very similar to those described in the manufacturing message service (MMS) specification [9],[10]. This is not surprising because one of the aims of the field bus standardization efforts is to fully integrate the instrumentation subnetworks in the framework of the factory network architecture and, in particular, in the networks used at the cell and plant levels (type two networks) that are based on more complex and powerful protocol profiles such as MAP [6]. Such an integration can be achieved if the application processes on the different devices connected to the network share a common view of the underlying communication system or, in other words, can access a common set of objects and services to handle them.
At present MMS is the only high level standard explicitly defined for the manufacturing environment, and its role is now widely accepted. FIP adopts a subset of MMS services, while PROFIBUS defines a slightly modified version known as fieldbus message specification (FMS). The following services are offered to the application processes in both cases:

- variable reading and writing,
- part programs uploading/downloading,
- remote control of program execution,
- management of events such as alarms.

FIP and PROFIBUS adopt different techniques to map field bus functionalities, such as the automated cyclic exchange of a number of variables with short response times, on the application layer services.

The FIP application layer is structured in two distinct functional units called MMS and MPS. The former is responsible for implementing a subset of the MMS messaging services, while the latter allows the user to access the FIP distributed variable data base supported by the underlying data link layer. The application model offered to the user consists of a set of variables, described in terms of type, attributes and value, and identified by character string names.

Application services are provided to read and write local and remote variables. It is also possible to refresh a variable value and to obtain information about the transmission/reception of a variable.

In PROFIBUS, by contrast, variables can be accessed by the user of the application layer only through the conventional MMS services based on message exchanges support. Even though this method seems to be not so efficient or simple as using a set of "ad hoc" protocol mechanisms, these MMS-like services offer, in practice, a method to access shared variables. In fact, as mentioned in section 1.2, PROFIBUS supports cyclical data exchanges by means of a special set of services (CSRD): this allows a master station to initiate an automatic poll operation on a set of slaves. Once the polling has been started it is periodically repeated each time the master becomes the token owner without any other service invocation by the application processes. It is possible to access this mechanism by means of a particular kind of virtual connection called cyclic connection.

Application processes can exchange variables (both simple and structured) either on a conventional virtual channel or on a cyclic connection. In the latter case, data is polled automatically from the slave stations by means of the CSRD mechanism in a way that is totally transparent to the user, thus creating a behaviour of the system similar to FIP where variable values are cyclically refreshed without any explicit user intervention.

3 Field bus performance

In this section some formulas are introduced in order to compare the field bus proposals assuming a specific network configuration. The cyclic exchange of variable values is one of the main characteristics of field buses. In fact, the ability of sampling data from sensors and sending commands to the actuators at a rate satisfactory for a significant number of automatic control applications largely depends on how efficiently cyclic transfers can be performed by the network.

For this reason we have considered the time elapsed between two subsequent exchanges of the same cyclic variable as the performance index to be used in our analysis. In the following this time is referred to as the time cycle T_C and we are interested in computing the minimum value for T_C for the standards considered in this paper.

3.1 SERCOS time-cycle

The SERCOS specifications define three kinds of messages (telegrams) used to implement the data exchange cycle in Fig. 2. Transmission times can be evaluated by considering the lengths of the messages transmitted within a SERCOS cycle. Thus, if M is the number of slaves, $\overline{L^{S \to M}}$ is the average number of bytes sent cyclically by each slave to the master and $\overline{L^{M \to S}}$ is the average number of bytes sent cyclically by the master to each slave, the following expression can be easily derived for the time-cycle T_C:

$$\begin{aligned} T_C \quad > \quad & t_{BIT} \cdot \left[94 + 48M + \lfloor 8M(7 + \overline{L^{S \to M}})1.2 \rfloor + \lfloor 8(3 + 4M + M\overline{L^{M \to S}})1.2 \rfloor \right] + \\ & T_{1min.1} + T_{ATMT} + T_{MTSY} \end{aligned} \tag{1}$$

In equation (1) t_{BIT} is the time needed to transmit one bit on the physical medium, the operator "$\lfloor \cdots \rfloor$" returns the integer part of its argument while $T_{1min.1}$, T_{MTSY} and T_{ATMT} are timing values defined in the SERCOS standard and shown in Fig. 2 and have the following meanings:

$T_{1min.1}$ is the shortest time which can be recognized by a drive,

T_{MTSY} is the maximum time interval needed by the drive in order to be ready to receive the MST from the end of the MDT,

t_{ATMT} is the maximum transition time needed to switch from transmitting an AT to the ready state to receive the MDT.

Equation (1) also takes into account the jitters introduced by the local clock of each station in the ring and the time needed by each station interface to switch from the transmitting to the repeating mode and vice versa. Moreover, the overhead introduced by the bit stuffing mechanism has also been considered.

3.2 FIP time-cycle

To compute the minimum value of T_C for a FIP network it is convenient to assume that the FIP macrocycle is composed of a single elementary cycle. This is not restrictive, since in our case cyclic variables have to be exchanged at the maximum possible rate.
A FIP elementary cycle consists of four kinds of data exchange: cyclic variables, acyclic variables and acyclic messages with and without acknowledgment. In general, all these transmissions affect the value of T_C. By indicating with T_{c_v}, T_{ac_v}, T_{m_ack} and T_{m_nak} the contribution introduced for T_C by each type of data exchange we can write the following expression:

$$T_C = T_{c\_v} + T_{ac\_v} + T_{m\_ack} + T_{m\_nak}$$

If the actual lengths of messages are used to evaluate the transmission times, when the FIP compact message encoding [5] is used, we obtain:

$$\begin{aligned} T_C &= (T_R + 61t_{BIT})(2N_{c\_v} + 4N_{ac\_v} + 4N_{m\_ack} + 3N_{m\_nak}) + \\ & \quad 8t_{BIT} \cdot \left(\sum_{i=1}^{N_{c\_v}} L_{c\_v_i} + \sum_{j=1}^{N_{ac\_v}} L_{ac\_v_j} + \sum_{k=1}^{N_{m\_ack}} L_{m\_ack_k} + \sum_{p=1}^{N_{m\_nak}} L_{m\_nak_p} \right) \end{aligned} \tag{2}$$

where:

N_{c_v} is the number of cyclic variables exchanged,

$L_{c\_v_i}$ is the length in bytes of each cyclic variable,

N_{ac_v} is the number of acyclic variables exchanged,

$L_{ac\_v_i}$ is the length in bytes of each acyclic variable,

N_{m_ack} is the number of messages with acknowledgment,

$L_{m\_ack_i}$ is the length (bytes) of each acknowledged message,

N_{m_nak} is the number of messages without acknowledgment,

$L_{m\_nak_i}$ is the length in bytes of each not acknowledged message,

T_R is the minimum time that must elapse between the transmissions of two subsequent frames on the network.

It is worth noting that equation (2) has been derived for the FIP elementary cycle, because it has been assumed to be the same as the FIP macrocycle.

3.3 Profibus time-cycle

The following expression of the time-cycle T_C can be obtained for a PROFIBUS network:

$$T_C = \sum_{i=1}^{N_M} \left(T_{TC} + \sum_{j=1}^{H_i} T_{MC_{i,j}}^{(high)} + \sum_{k=1}^{L_i} T_{MC_{i,k}}^{(low)} + \sum_{l=1}^{R_i} T_{MC_{i,l}}^{(ret)} \right) \tag{3}$$

where:

N_M is the number of masters in the network,

T_{TC} is the token cycle time,

$T_{MC_{i,x}}^{(y)}$ is the message cycle time for x_{th} message sent by the i_{th} master. y indicates high priority, low priority or retry respectively,

H_i is the number of high priority messages sent by master i in a cycle,

L_i: is the number of low priority messages sent by master i in a cycle,

R_i: is the number of retry messages sent by master i in a cycle.

3.4 Minimum time cycle

In order to evaluate the minimum time cycle for SERCOS, FIP and PROFIBUS a preliminary consideration has to be made, mainly because the three networks are very different in nature. In particular, since SERCOS is more special-purpose and less flexible than FIP and PROFIBUS we have selected a typical SERCOS configuration and computed the minimum T_C value and compared it with a similar system designed using FIP and PROFIBUS. In general, the functionalities of any FIP or PROFIBUS network can not be obtained by means of a SERCOS system, thus this assumption is necessary to make the comparison possible.

Since only one master is present in a SERCOS network a PROFIBUS configuration including a single master station is used while in FIP the master functions can be performed by the bus arbiter. Our sample system is one of typical digital control with the following characteristics:

- the network consists of $M+1$ stations that is M drives (slaves) and one control unit (master);
- only cyclic and acyclic variables are exchanged between the control unit and the drives; cyclically each slave sends a variable to the master and the master sends a variable to each slave;
- the minimum time elapsing between two subsequent messages sent on the network is proportional to the transmission time of a single bit t_{BIT} by a constant α.

SERCOS

Equation (1) gives a lower bound for T_C when $T_{1min.1}$, T_{ATMT} and T_{MTSY} are replaced by their minimal values. In this case we obtain:

$$T_C \quad > \quad t_{BIT} \cdot \left[94 + 48M + 3\alpha + \lfloor 8M(7 + \overline{L^{S \to M}})1.2 \rfloor + \lfloor 8(3 + 4M + M\overline{L^{M \to S}})1.2 \rfloor \right] \quad (4)$$

It is worth noting that the SERCOS standard reserves only a small amount of the system bandwidth for the transmission of acyclic values during each cycle. Our example takes into account possible acyclic data exchanges which, for this reason, should be split up on several network cycles.

FIP

The FIP network contains $M+1$ variable producers: on each macrocycle the M drives send their variables to the master and the master transmits a burst of variables (one for each drive). The master packs all the values for the slaves into a single structured variable in order to obtain a more efficient transmission. In this way each station generates a single data frame and N_{c_v} is equal to $M+1$ in equation (2).

Moreover, no message transfer has to be taken into account and T_R can be considered to be equal to αt_{BIT}. In these conditions equation (2) becomes:

$$T_C = t_{BIT} \cdot \left((\alpha+61)[2(M+1)+4N_{ac\_v}] + 8(\sum_{i=1}^{M+1} L_{c\_v_i} + \sum_{j=1}^{N_{ac\_v}} L_{ac\_v_j}) \right) \tag{5}$$

Since the structured variable transmitted by the master can be significantly longer than the variables sent by the slaves it is convenient to rewrite equation (5) in the following way:

$$T_C = t_{BIT} \cdot \left((\alpha+61)[2(M+1)+4N_{ac\_v}] + 8M(\overline{L^{M\to S}} + \overline{L^{S\to M}}) + 8\sum_{j=1}^{N_{ac\_v}} L_{ac\_v_j} \right) \tag{6}$$

where $\overline{L^{M\to S}}$ and $\overline{L^{S\to M}}$ are the mean length of the master variables and the mean length of the slave variables respectively.

PROFIBUS

In PROFIBUS, M slaves and one master must be considered and in this case it is also assumed that all transmissions are successful (i.e. there are no transmission retries) and all messages are sent at the lower priority. Equation (3) can then be rewritten as:

$$T_C = t_{BIT} \cdot \left[11M(\overline{L^{M\to S}} + \overline{L^{S\to M}}) + 209M + 253 + M\alpha + (M+2)max\{\alpha, 35\} \right] +$$

$$t_{BIT} \cdot \left\{ 11 \sum_{i=1}^{N_{ac\_v}} L_{ac\_v} + N_{ac\_v} \cdot [616 + 2(\alpha + max\{\alpha, 35\})] \right\} \tag{7}$$

where the first term refers to cyclic variables, while the second one refers to acyclic variables. $\overline{L^{M \to S}}$, $\overline{L^{S \to M}}$, N_{ac_v}, L_{ac_v} and αt_{BIT} have the same meaning as in SERCOS and FIP.

3.5 Results

Equations (4), (6) and (7) have been used to draw the diagram families shown in figures 3, 4, 5 and 6. Each diagram shows the behaviour of the ratio T_C/T_{BIT} versus either the average length (L) of messages transmitted over the network or the number (M) of slave stations present in the system. The use of a normalized time cycle makes it possible to compare the three kinds of protocols since the transmission speeds specified in the standard documents are significantly different in the three cases (see Tab. 1).

Figs. 3 and 4 show how the message length affects the value of T_C in a network consisting of one master and eight slaves. In particular, the model of the system considered in Fig. 3 is such that one cyclic variable is written and one cyclic variable is read by the master for each slave in the network. In Fig. 4 also acyclic variable exchanges are taken into account, since the master satisfies two acyclic variable access requests per cycle coming from any two slaves.

In both the cases in Figs. 3 and 4, FIP and SERCOS perform better than PROFIBUS and this is more evident as the lenght of the messages sent on the network increases. Moreover, when the acyclic traffic is also considered the behaviour of SERCOS is slightly more satisfactory than that of FIP. This is due to the fact that only a limited amount of acyclic exchanges has been considered, which fits in well with the small part of system bandwidth reserved to this kind of traffic in SERCOS.

In Figs. 5 and 6 T_C is plotted versus the number of slaves M when an average packet length of six bytes is assumed and also in this case the absence or presence of acyclic exchanges has been considered. Once again the FIP and SERCOS networks seem to be more efficient and this becomes more evident when the number of stations is increased.

The reader must be warned that a fair comparison of the three protocols must take into account also other aspects which have not been analyzed in this paper. For instance, network throughput is another important performance index that can lead to results very different from those derived for T_C. Moreover, the ability to integrate a field bus subnetwork in the whole manufacturing communication system should also be considered: from this point of view PROFIBUS and FIP are certainly more "open" and oriented to internetworking than SERCOS.

4 Conclusions

The evolution of integrated manufacturing systems has speeded up the introduction of digital communication networks at all hierarchical levels in modern factories. In the last few years several communication standards which are intended to support industrial applications have been approved by the international committees. A number of pilot plants have been established in the United States, Europe and Japan based on these standards in order to achieve complete integration and the interconnection of open etherogeneous devices.

While considerable attention, in the past, has been paid to "high level" networks oriented to complex and heavy tasks and able to interconnect mainframes, minicomputers, cell controllers and so on, little has been done to develop international standards for field bus networks that are to be used with more simple and less intelligent devices such as drives, sensors and actuators.

The demand for low-cost communication systems which can satisfy real-time requirements has led some European national standardization bodies to promote their own field bus standards and then to submit them to the international organizations for approval.

This paper has dealt with three emerging proposals, namely FIP, PROFIBUS and SERCOS. The main characteristics and communication mechanisms introduced for each network have been discussed with the aim of presenting a brief account of the current state of the art concerning the evolution of field buses and their features with respect to their use in a factory environment.

Since the FIP, the PROFIBUS and the SERCOS networks are currently being implemented in Europe, although they are not widely available, some preliminary considerations about what performance level can be expected from such systems have been introduced. Particular attention has been paid to the real-time supports for exchanging cyclic data that are a distinctive characteristic of the field bus protocols.

References

[1] McLean C., Mitchell M. and Barkmeyer E. "A Computer Architecture for Small-Batch Manufacturing", IEEE Spectrum, n. 5, May 1983.

[2] SERCOS Interface Inc. "SERCOS Interface: Digital Interface for Communication between Controls and Drives for Numerically Controlled Machines", Frankfurt, 1990.

[3] International Electrotechnical Commission "Electrical Equipment of Industrial Machines - Serial Data Link for Real-time Communication between Controls and Drives", Draft International Standard, Revision 7, June 1993.

[4] German Institute of Normalization "PROFIBUS standard part 1 and 2", DIN 19 245, April 1991.

[5] French Association for Standardization "FIP Bus for Exchange of Information Between Transmitters, Actuators and Programmable Controllers", NF C46 601-607, 1990.

[6] General Motors "Manufacturing Protocol Specification Version 3.0", 1988.

[7] Computer Networks for Manufacturing Automation "CNMA Implementation Guide version 4.0", 1989.

[8] IEEE "Token Bus Access Method", Std. No. 802.4, 1985.

[9] ISO "Manufacturing Message Specification - Service Definition", DIS 9506-1 TC 184 / SC 5 / WG 2, 1988.

[10] ISO "Manufacturing Message Specification - Protocol Specification", DIS 9506-2 TC 184 / SC 5 / WG 2, 1988.

[11] P. Montuschi, L. Ciminiera, A. Valenzano "Time characteristics of IEEE 802.4 token bus," in IEE Proceedings Part E, Vol. 139, No. 1, Jan. 1992, pp. 81-87.

[12] International Electrotechnical Commission "Proway C Specifications", IEC Std. 955, 1989.

Standard	Encoding	Access method	Media type	Topology	Transm. speeds.	Maximum length	Max. Nodes Segm./Tot.
FIP	Manchester with violation bits	Centralized Polling	Twisted pair	Bus	31.25 kbit/s	1500 m	32/256
					1 Mbit/s	500 m	
					2.5 Mbit/s		
			Fiber optics	Star	5 Mbit/s		
PROFIBUS	NRZ Asynchronous character oriented	Token passing with Master/Slave polling	Twisted pair (RS-485)	Bus	9.6 Kbit/s	1200 m	32/127
					19.2 Kbit/s		
					93.75 Kbit/s		
					187.5 Kbit/s	600 m	
					500 Kbit/s	200 m	
SERCOS	NRZI with bit stuffing	Slotted ring	Fiber optics	Ring	2 Mbit/s		255/255

Table 1: Physical characteristics

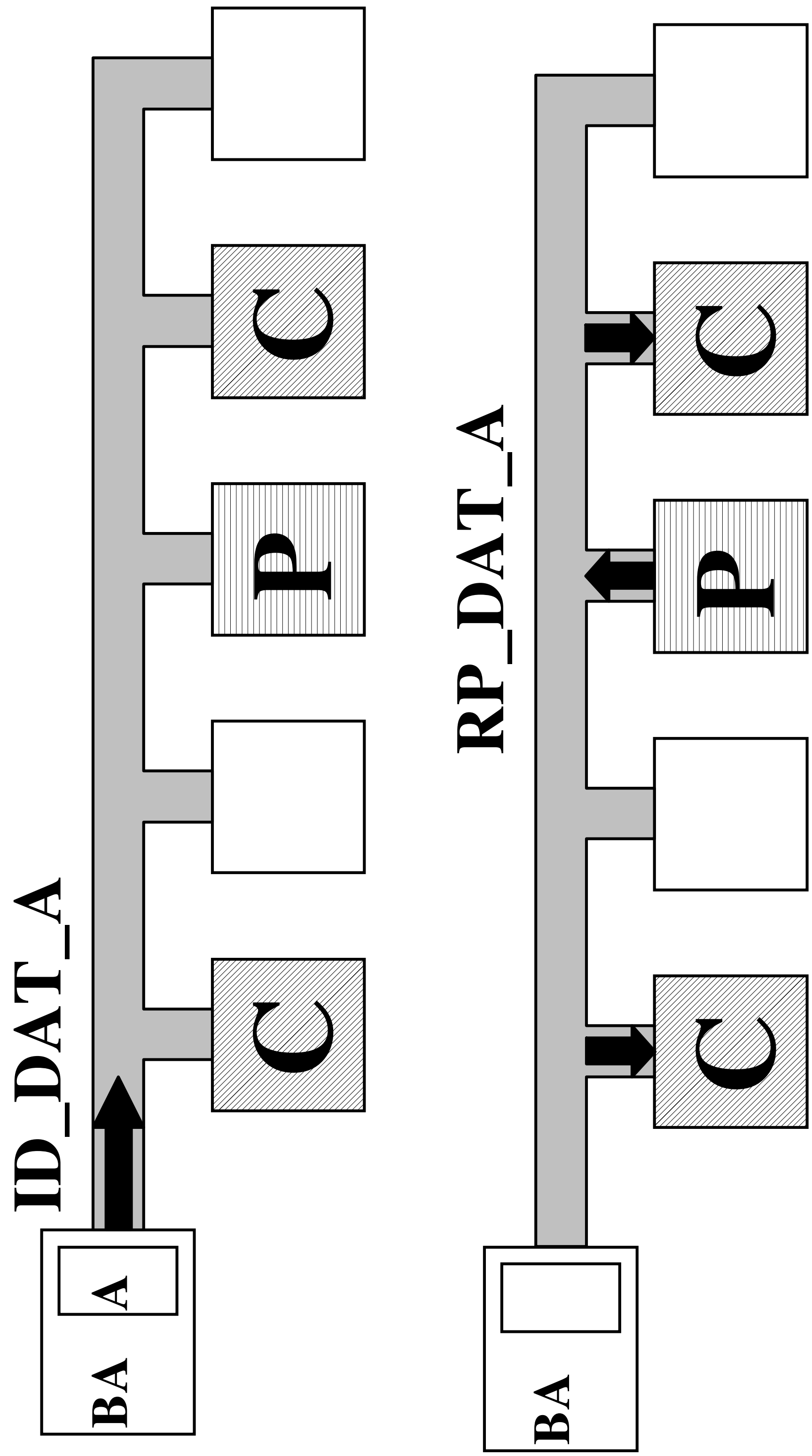

Figure 1:

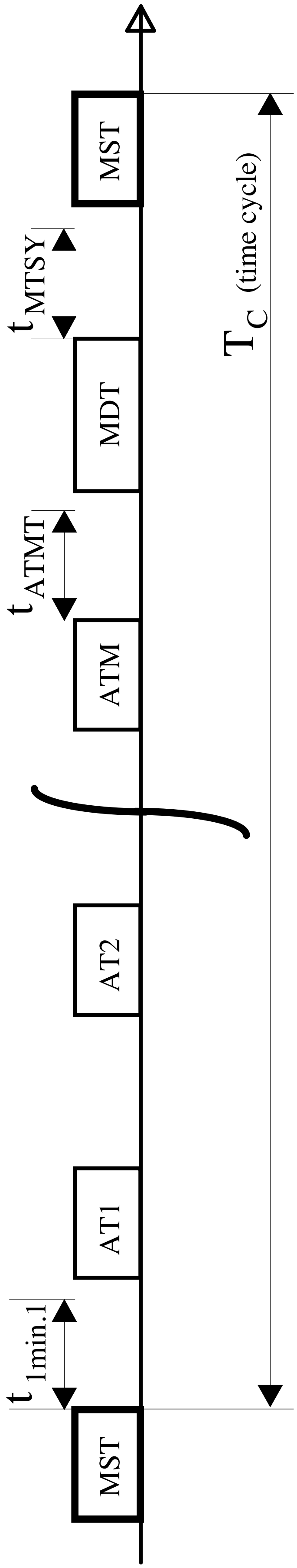

Figure 2:

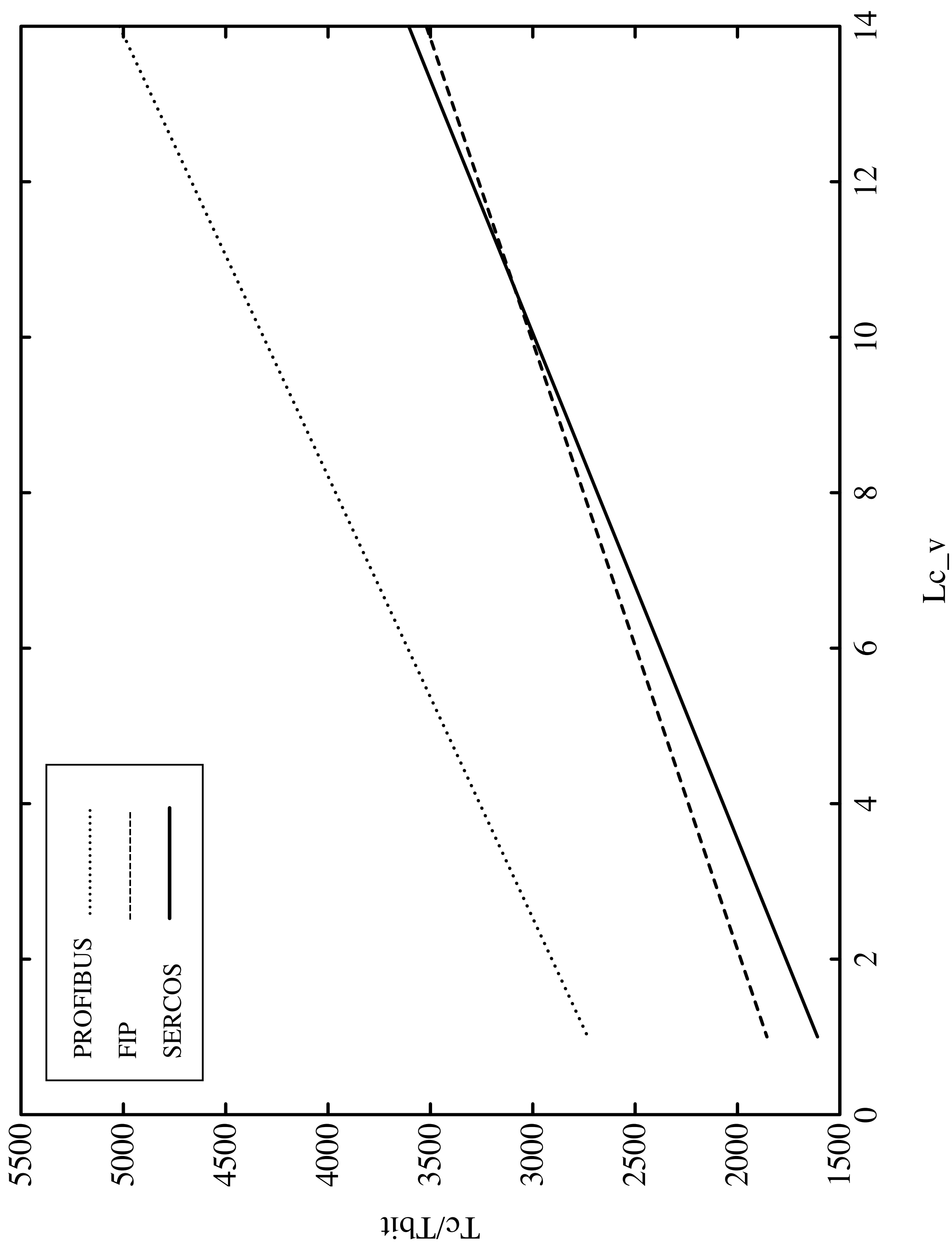
PROFIBUS
FIP
SERCOS
Lc_v
Tc/Tbit
0
2
4
6
8
10
12
14
1500
2000
2500
3000
3500
4000
4500
5000
5500

Figure 3:

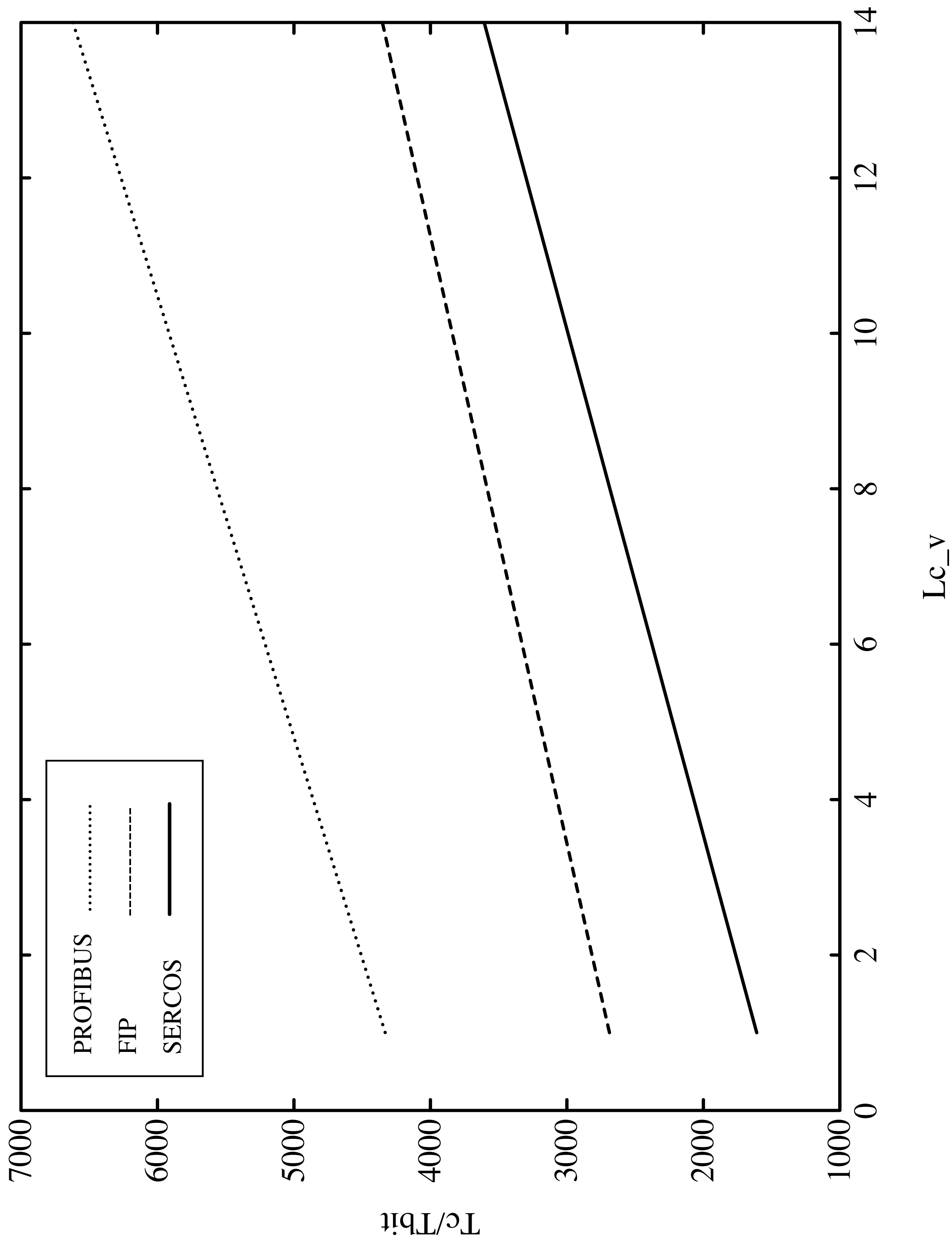
PROFIBUS
FIP
SERCOS
Lc_v
Tc/Tbit
0
2
4
6
8
10
12
14
1000
2000
3000
4000
5000
6000
7000

Figure 4:

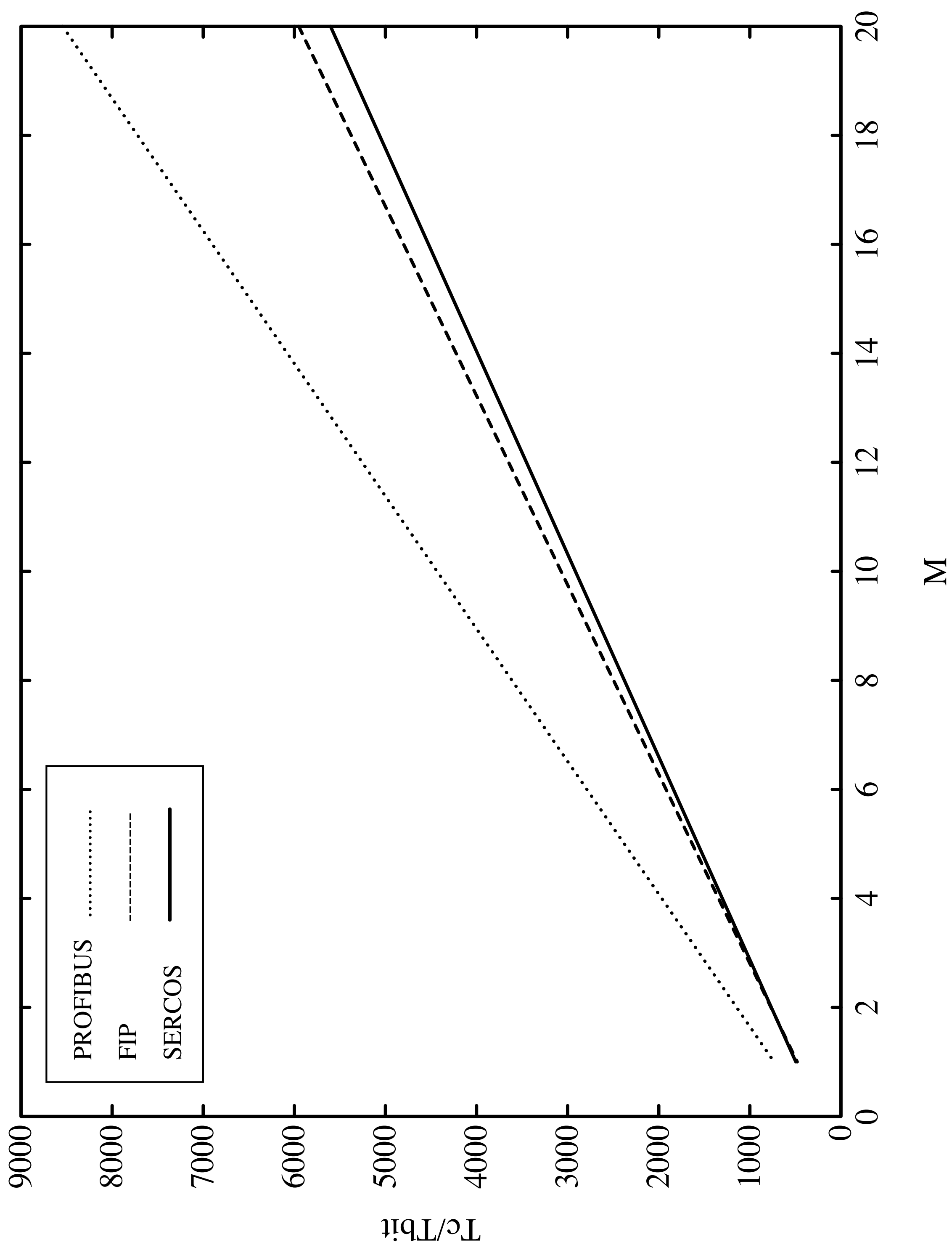
PROFIBUS
FIP
SERCOS
M
Tc/Tbit
0
2
4
6
8
10
12
14
16
18
20
0
1000
2000
3000
4000
5000
6000
7000
8000
9000

Figure 5:

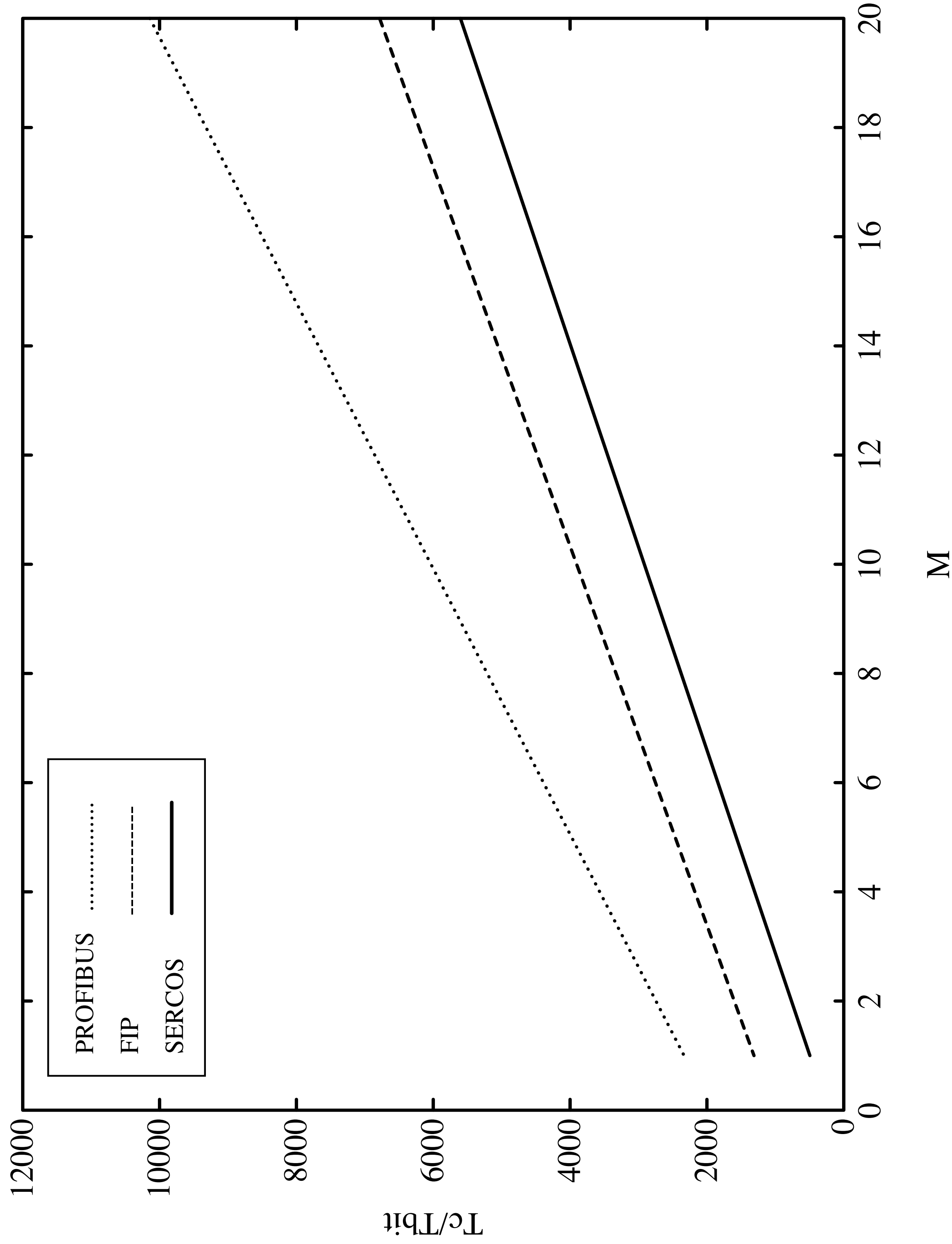
PROFIBUS
FIP
SERCOS
Tc/Tbit
M
0
2000
4000
6000
8000
10000
12000
0
2
4
6
8
10
12
14
16
18
20

Figure 6:

List of figures