Quality of service in industrial applications

Miguel Leon-Chavez, Jean-Pierre Thomesse

Quality of Service in Industrial Applications

Miguel LEON CHAVEZ[1] Jean-Pierre THOMESSE[2]

[1]Benem rita Universidad Aut noma de Puebla
14 Sur y Av. San Claudio, CP 72570, Puebla, M xico
Tel. (52) 222 229 55 00 ext. 7213 Fax (52) 222 229 55 00 ext. 7200
email: mleon@cs.buap.mx

[2]LORIA - INPL - ENSEM, 2, av. de la For t de Haye
F-54516 Vandoeuvre-l s-Nancy, France
Tel. (33) (0)3 83 59 57 46 Fax (33) (0)3 83 44 07 63
email: thomesse@loria.fr

Abstract. This paper presents an analysis of the Quality of Service (QoS) Architectures of the following communication networks: Fieldbuses (IEC 61158, WorldFIP and ControlNet) and Internet. The analysis takes into account, on one hand, the end-user requirements and, on the other hand, the different mechanisms defined by each model to meet and guarantee the end-user requirements. The analysis shows that it is required new QoS architectures in order to provide end-to-end QoS in distributed industrial applications

Keywords: Quality of Service, Fieldbus, Internet.

1 Introduction

Industrial applications are more and more distributed and integrated, not only inside the factory itself but also between remote sites of production, of management, of product development and of product distribution. Various communication networks are then the support of this distribution and become now the backbone of the industrial applications (Almeida, Le n, Fonseca, & Thomesse, 1999).

Inside the factory, the communication networks are hierarchically organized according to levels in what is known as the Computer Integrated Manufacturing (CIM) architecture. Three main levels are usually considered: process/machine control, cell control and floor plant control. For each of these levels there are several solutions, for example, at the lowest level, i.e. the process/machine control, the device networks interconnect the sensors and the controllers. The simultaneous control of several processes or machines is carried out at the cell control level by the fieldbuses. At the higher level, the Local Area Networks based on either Ethernet or IBM Token Ring promote the integration of all the cells within the factory. Outside the factory, the interconnection of the remote sites is carried out with Wide Area Networks which rely upon services provided by telecommunications operators. At this level either dedicated leased lines or Internet/ATM-based services are typically used.

All these communication networks provide different services according to the different needs, and an important point of comparison is the Quality of Service (QoS) Architecture that have been defined in order to meet the different end-user requirements. For example, on one hand, the fieldbuses such as WorldFIP and ControlNet define a periodic or scheduled window to guarantee the transmission of the periodic message; The (IEC 61158, 1999) defines a QoS architecture which is based on both the Data Link Services (DLS) and the QoS attributes which are common between the different types of DLS. And on the other hand, the Internet Engineering Task Force (IETF) has proposed some service models and mechanisms to meet the demand for QoS, one of them is the Subnet Bandwidth Management (SBM) for the IEEE 802-style LANs.

This paper deals with these topics. Which are the end-user requirements? What kind of service provides each communication network? Which are the QoS architectures for each communication network? Is a future common solution possible for current very different QoS architectures?

To treat these aspects, the paper will be organized as follows. In a first part, a top-down approach will be considered to analyze and classify the end-users requirements and to identify the different traffics in relation with the current architectures defined for such applications as well inside a factory, around LANs, as in remote cooperation based on WANs. In a second part, the main solutions will be studied considering the following QoS architectures: three Fieldbus architectures (IEC 61158, WorldFIP and ControlNet) and the Internet QoS architecture. Finally some conclusions are presented.

2 End-user requirements

In industrial applications, the end-users are the sensors, the actuators, the controllers, the transmitters, the programmable controllers, the production machines controllers, the processors, the transport systems, the terminals, the workstations, the people who are in charge of the control, of the maintenance, of the production scheduling, of the technical management and so on. But also the end-users are the software components which have to cooperate, to be coordinated in order to allow efficient monitoring and control of the industrial process operation and consequently of the quality of the products.

The communication requirements of the end-users can be essentially characterized by the types of the data that need to be exchanged, either in what concerns their temporal properties as well as the size of the respective transactions.

2.1 Temporal requirements

In industrial applications time-constrained data can be commonly found, i.e. data that have a limited time validity. Hence, all the processing of such data, including both computing and transmission, must be carried out in a bounded time. This time window must be sufficiently small in order to allow efficient monitoring and control of the industrial process operation and consequently of the quality of the products. How short such deadline must be depends on the particular physical process under consideration. Examples of such data are the feedback and control signals of closed loop control subsystems or the temperature control in a chemical reaction.

These data are called *critical data* which can be further characterized as *soft* or *hard*. In the former case, a few violations of the processing deadline can be tolerated. On the other hand, in the latter case, the applications cannot tolerate any delay in the respective processing beyond the deadline, i.e. the end of the time validity. This is the case when missing a single deadline may jeopardize the whole system operation and even endanger equipment, people or environment.

Besides critical data, the applications make also use of non-time-constrained data. In this case, a delay on the respective processing has no effect on the proper operation of the manufacturing process nor on the quality of the products. This is called *non-critical data* and is normally handled by background application tasks.

Another time-related aspect that concerns the data is the regularity and frequency at which it is produced and processed. Typically, the industrial applications know the current status of the industrial process by means of the data produced and transmitted either periodically or randomly by the sensors or higher level devices. In the same way, the applications control the industrial process by means of the control data transmitted to the control devices. In any case, when the transmission of such data is carried out with a constant frequency then it is named *periodic* and it can be critical or not.

On the other hand, critical data resulting from random emergency situations is normally transmitted aperiodically and thus it is named *aperiodic*. Notice, however, that there might be non-critical data requested aperiodically, too, e.g. by a high-level supervision task.

In some cases it is possible to bound the minimum separation time between two consecutive aperiodic transactions of the same source. This particular sort of aperiodic data is normally known as *sporadic*.

2.2 Size of data per transaction

With regard to the size of data to be transferred per transaction, some industrial applications exchange bits, some of them exchange words of data, some others exchange streams of data and long streams of data.

The bits of data are typically associated with the status of very simple devices, usually restricted to on/off status. Applications dealing with such sort of devices require transfers of a few bits of I/O data per node, only. However, some devices, either sensors or controllers, are more elaborated and use more information such as status, diagnostic and control. In this case, the associated applications require the transfer of words of I/O data.

Going up in the device complexity hierarchy there are the higher level devices such as CNC machines, robots, and other equipment. The operation of such devices requires a stream of several words of data associated with status, configuration, diagnostic, calibration and control. The applications that manage these devices require, then, the exchange of streams of I/O data. At even higher levels in the industrial system there is the need to transfer files, to access databases, to gather the data associated to the whole system, to allow remote access to the system, etc. These operations commonly require the exchange of long streams of data per transaction.

It is interesting to note that, in general, as the size of the data per transaction increases, the associated time-constraints, either deadline or frequency, become more relaxed. In fact, transactions carrying bits or words of I/O data are normally associated to sensors and controllers which are directly involved in the inner control loops of a system or subsystem, thus requiring relatively high communication rates. Streams of I/O data, normally associated with higher level devices, are involved in outer control loops, thus requiring relatively lower communication rates. At the manufacturing system higher levels, the applications typically require long streams of data associated to the overall control and supervision of the whole plant and the feedback actions are usually performed by human intervention. Thus, the respective temporal requirements are even more relaxed.

3 QoS Architectures

Several QoS architectures have been proposed for different communication networks, a survey of the QoS architectures is presented in (Aurrecoechea, Cambell, & Hauw, 1998). A survey of the Internet QoS architecture is presented in (Xiao & Ni, 1999). Next sections analyze two QoS architectures: the Fieldbus architectures (IEC 61158, WorldFIP and ControlNet) and the Internet QoS architecture.

3.1 Fieldbus QoS architectures

The *Fieldbuses* are special purpose LANs used to connect all kinds of devices into the factory. These networks are based on a layered architecture which only includes the physical layer, the data link layer (MAC and LLC) and the application layer (Thomesse, 1998).

Nowadays, there is one approach in order to define the Fieldbus QoS architecture, i.e. the IEC 61158 QoS architecture (IEC 61158, 1999), which is based on both the Data Link Services (DLS) and the QoS attributes which are common between the different types of DLS. The IEC 61158 can be seen as a collection of Fieldbus Standards because it defines eight types of DLS, numbered from 1 to 8, each corresponding to a Fieldbus Standard as follows: TS 61158, ControlNet, Profibus, P-Net, Foundation Fieldbus, SwiftNet, WorldFIP and Interbus, respectively.

Basically, the DLS provide transparent and reliable data transfer among the DLS-users. In addition to the basic classes of DLS (connection-mode and connectionless-mode) the IEC 61158 defines the two following classes of DLS: a DL(SAP)-address, queue and buffer management service and a time and transaction scheduling service.

In this approach, a DLS-user may select, directly or indirectly, the parameters of the QoS attributes (i.e. priority, maximum confirm delay, authentication, scheduling-policy and timeliness) in order to determine the quality of the DLS.

The IEC 61158 QoS architecture is well adapted to the type 1 of DLS, i.e. the TS 61158 Fieldbus, simply because the DL-entities can control the QoS attributes. Nevertheless, it is not always the case in the other types of DLS. A possible solution is, of course, mapping the QoS attributes and then to define the DL-entities to control them, as in SwiftNet case.

On the other hand, ControlNet and WorldFIP have defined their own QoS architecture, next subsections will analyze them.

3.1.1 IEC 61158 Part 3 QoS architecture

The IEC 61158 Part 3 defines a QoS architecture based on the Data Link Services (DLS), which provide both transparent and reliable data transfer between DLS-users. The two following basic classes of DLS are defined: connection-mode data transfer service and connectionless-mode data transfer service. In the first class, before data transmission, it is required to establish a logical channel, called connection; after data transmission, the connection is released. In the second class, no any connection is required before data transmission.

In addition, two other classes of DLS are proposed as follows: a DL(SAP)-address, queue and buffer management service and a time and transaction scheduling service. The former defines the interactions between the DLS-user and the DLS-provider that take place at a local DLSAP (DL Service Access Point). Information is passed between the DLS-user and the DLS-provider by DLS primitives that convey parameters. The time service provides DLS-users with a means to indirectly synchronize and schedule their activities with a shared sense of time. The scheduling service allows completing a deferred primitive, distributing the current value of a buffer and starting an exchange service.

It can be noted that the four classes of DLS will be provided according to the types of DLS, i.e. the DLS-user is limited to those classes of service supported by the selected DL protocol implementation, i.e. the Fieldbus network.

The IEC 61158 defines the following QoS attributes: DLL priority, DLL maximum confirm delay, DLPDU authentication, DL-scheduling-policy and DL-timeliness. The first four QoS attributes apply conceptually to both connection-mode and connectionless operation. The fifth attribute applies only to connection-mode operation. Next paragraphs describe the attributes and their parameters.

The DLL (DL Layer) priority attribute specify an associated DLL priority used in scheduling DLL data transfer services and also determines the maximum amount of DLS-user-data that can be conveyed in a single DLPDU (DL Protocol Data Unit). Three DLL priorities with their corresponding ranges of conveyable DLS-user-data (per DLPDU) are, from highest priority to lowest priority: *Urgent* (up to 64 octets), *Normal* (up to 128 octets) and *Time-Available* (up to 256 octets). Urgent and Normal are considered *time-critical* priority levels; Time-Available is considered a *non-time-critical* priority level. The default QoS value is Time-Available but it can be set by DL-management.

The DLL maximum confirm delay attribute specifies upper bounds on the maximum duration permitted for the completion of each related instance of a sequence of connection-oriented and connectionless DLS primitives. Each parameter either has the value *Unlimited* or specifies an upper bound, in a given time unit, for example 1 ms, from 1 ms to 60 s, and so on. The default QoS value is Unlimited but it can be set by DL-management.

The DLPDU authentication attribute determines a lower bound on the amount of DL-addressing information used in the DLPDUs that provide the associated DLL data transfer services. Three levels are specifiable: *Ordinary*, each DLPDU includes the minimum permitted amount of addressing information. *Source*, each DLPDU includes a source DL-address where possible. *Maximal*, each DL-address includes the maximal amount of addressing information possible. The default QoS value is Ordinary but it can be set by DL-management.

The DL-scheduling-policy attribute allows deferring, canceling and resuming any DLS-user intercommunication. Two choices are possible: *Implicit*, any required communications with peer DLS-user(s) from a DLSAP-address, or from a DLCEP (DL Connection End Point), will occur as soon as possible. *Explicit*, any required data communications with peer DLS-user(s) from a DLSAP-address, or from a DLCEP, will occur only when the deferral is explicitly cancelled by an involved DLS-user. The default QoS value is always Implicit it cannot be set by DL-management.

The DL-timeliness attribute only applies to connection-mode service. Then, each DLCEP establishment request, and each response, can specify DL-timeliness criteria which are applied to information sent from, or received into, retentive buffers at that DLCEP. Four types of DL-timeliness can be supported: *Residence*, *update*, *synchronized* and *transparent*.

Residence timeliness is an assessment based on the length of time that a DLS-user datum has been resident in a buffer, which is the time interval between the moment when the buffer is written and the moment when the buffer is read.

Update timeliness is an assessment based on the time interval between the moment of occurrence of a multi-DLE synchronizing event and the moment when the buffer is written.

Synchronized timeliness is an assessment based on the time intervals and timing relationships between: the moment of occurrence of a multi-DLE synchronizing event (S_T), and the moment when the buffer is written, and the moment when the buffer is read; So: DL-timeliness = $0 \leq (W_T - S_T) \leq (R_T - S_T) < \Delta T$

Transparent timeliness occurs when timeliness is selected on a DLCEP but none of the above assessments are performed. In such a case the DLC preserves any prior buffer timeliness, but does not itself invalidate that timeliness. When no prior buffer timeliness exists, the default timeliness value is *true*.

No timeliness occurs when timeliness is not selected on a DLCEP. In such a case the DL-timeliness attribute of DLS-user data always is *false*.

3.1.2 ControlNet QoS architecture

ControlNet uses a Concurrent Time Domain Multiple Access (CTDMA) method to access the medium and the Producer-Consumer model, i.e. each transmitted message has been assigned an identifier which localizes the data to be produced or consumed.

In CTDMA, the network time is divided into time windows, called Network Update Intervals (NUI), which are further divided into three windows or portions: scheduled, unscheduled and network maintenance. During the scheduled window, each configured node transmits one and only one time-critical message in a given time. During unscheduled window, the nodes transmit one or more non time-critical messages according to the network load, the transmission order is rotated in each NUI (round robin). The network maintenance window is used as "guardband" to allow the constant duration of the NUI, as it is shown in figure 1.

It is interesting to note that these windows provide two levels of QoS, i.e. given that the NUI are cyclically repeated, the scheduled window provides a cyclic guaranteed service since the messages are transmitted at a deterministic and repeatable rate; the unscheduled window provides a best-effort service since the messages are transmitted as time permits.

It should be noted that ControlNet uses a distributed and static resource reservation mechanism in order to provide the guaranteed service of QoS, i.e. all the nodes must be configured off-line, at network configuration time, in order to have one slot into the scheduled window, furthermore each node must know the time critical message to be transmitted since the scheduled window always allows each configured node one and only one chance to transmit in a given interval.

However, if the number of nodes increases or if the time critical messages in each node change it is necessary to stop the application in order to determine whether the new set of requirements can be met and guaranteed. If yes, the user requirements are configured and the application is restarted. However, many applications cannot be stopped without great losses.

3.1.3 WorldFIP QoS architecture

World Factory Instrumentation Protocol (Galara & Thomesse, 1984) (EN50170-3) defines a Bus Arbitrator (BA) which Ògives permission to speakÓ to each information producer. The application layer sends to the BA, at the data-link layer, a set of service request primitives that describe the execution of both the Basic Cycles and the Macro Cycles (the chaining of one or more basic cycles). A basic cycle is composed of at least one window, the periodic window, and at most of four windows: a periodic window, an aperiodic variables window, an aperiodic messages window and possibly a synchronization window to adjust the constant duration of the basic cycle, as it is shown in figure 2.

Each variable is identified by a unique identifier and each node can be a producer and/or consumer of one or more variables. During the periodic window, BA reads a list of periodic variables and injects each variable identifier onto the network, each variable has only one producer. Consumers needing to utilize the variable, alerted by the identifier, store and use the value broadcasted by the producer.

Whenever an aperiodic variable or message are produced, the producer utilizes the response, of a received periodic variable identifier, to request its transmission. The bus arbitrator stores the identifier, that is carried by the request, into the appropriate queue. After completing the current periodic window, the requested aperiodic variables or the requested messages are then handled in the same way that the periodic variables within the proper window, where the relevant producers reply current values.

It is interesting to note that these windows provide two levels of QoS, i.e. the periodic window provides a cyclic guaranteed service. In fact, in (Song, 1991) a performance evaluation of FIP is presented, from this work, it is possible to compute the message transfer delay to build the list of periodic variables and to compute the size of the windows in a basic cycle. On the other hand, the aperiodic variable and message windows provide the best-effort services since if the duration of the synchronization window is zero new requested aperiodic variables or new requested messages will be transmitted in the next basic cycle.

It can be noted that WorldFIP uses a centralized and static resource reservation mechanism, i.e. there is one node, the BA, having different lists of identifiers that define at any moment the communication requirements of all the nodes. These requirements are known a priori and will not change during application operation. Hence, it is possible to assign an identifier to each variable.

Here again, if the number of nodes increases or if the periodic messages in each node change it is necessary to stop the application in order to determine whether the new set of requirements can be met and guaranteed. If yes, the user requirements are configured and the application is restarted.

3.2 Internet QoS Architecture

Internet or the TCP/IP model is basically composed of the services implemented by the application, transport and internet protocols. Nowadays, this model only provides best-effort services and this is why the Internet Engineering Task Force (IETF) has proposed some service models and mechanisms to meet the demand for QoS.

One service model will be discussed in this section: Integrated Services (IntServ). There are other service models such as Differentiated Services (DiffServ), Multiprotocol Label Switching (MPLS), Traffic Engineering and Constraint-based Routing but are not discussed here, a survey of them can be found in (Xiao & Ni, 1999).

For the IEEE 802-style LANs, a Subnet Bandwidth Management (SBM) has been proposed (Ghanwani et al., 2000; Yavatkar et al., 2000; Seaman et al., 2000) but its main requirement is that all traffic must pass through at least one switch using the IEEE 802.1p, IEEE 802.1Q, or IEEE 802.1D. In this way, Integrated Services can be supported by shared and switched LANs using the IEEE 802 MAC protocols such as 802.3, 802.5 and 802.12. For the 802.3, an explicit user-priority field on top of the basic MAC frame format is then required. These Internet service models are analyzed in the following subsections.

3.2.1 Integrated Services model

The IntServ model (Braden, Clark, & Shanker, 1994) assumes that the resources of the communication system (e.g. bandwidth and buffers) must be explicitly managed in order to meet the user requirements. Given that the resources are finite, resource reservation and admission control are key building blocks of this service. Two other building blocks are required: packet scheduler and classifier.

In order to state the resource requirements, an application must specify the desired QoS using a list of parameters, called flowspec, which is carried by the reservation protocol, passed to the admission control for testing acceptability, and finally used to set the parameters of the packet scheduling mechanism.

The resource reservation protocol (Zhang, Deering, Estrin, Shenker, & Zappala,1993) creates and maintains flow-specific state in the endpoint hosts and in routers along the path of a flow. The Admission control block implements the decision algorithm that a router or host uses to determine whether a new flow can be granted the requested QoS without impacting earlier guarantees. The Packet scheduler block manages the forwarding of different packet streams using a set of queues. The Classifier block maps each incoming packet into some class; all packets in the same class get the same treatment from the packet scheduler.

It can be noted that this service model is situated at the transport protocol of the Internet architecture. The model makes use of the services which are furnished by the network protocol (routing and packet scheduling) to reserve resources along the path of the packets.

The IntServ model defines the following levels of QoS:

Guaranteed: It provides firm (mathematically provable) bounds on end-to-end queuing delays by combining the parameters from the various network elements in a path, in addition to ensure the bandwidth availability according to a traffic specification.

Controlled Load: This is equivalent to the best effort service under unloaded conditions. Hence, it is better than best effort level.

3.2.2 SBM architecture

Basically, the Subnet Bandwidth Management is a signaling protocol (Yavatkar et al., 2000) that allows Integrated Services over shared and switched LAN networks (Ghanwani et al., 2000) and enables Integrated Services mapping on these networks (Seaman et al., 2000). The main components of the SBM architecture are as follows:

Bandwidth Allocator (BA), which maintains state information about allocation of resources on the subnet and performs admission control. BA can be centralized or distributed.

Requester Module (RM), which resides in every end-station (host or router) and not in any switch. This module is responsible for mapping the higher-layer QoS parameters into the data link layer priority levels.

SBM protocol, which is a RM-to-BA or BA-to-BA signaling protocol for reserving resources, querying a BA about available resources, and changing or deleting reservations. This protocol also allows the communication between higher-layer QoS protocols and the RM. The primitives could be implemented as an API for an application to invoke functions of the BM via the RM.

4 Conclusion

This paper has presented an analysis of the Quality of Service in industrial applications. The analysis takes into account, on one hand, the end-user requirements and, on the other hand the Fieldbus QoS architectures and the Internet QoS architecture.

Fieldbuses are used in applications where most of the traffic is composed of periodic messages, which are known a priori, at network configuration time, and then the communication network can guarantee bounded message transfer delays and timing scheduling using different mechanisms. Nevertheless, there are also aperiodic messages, which are known at application run time, and for which the communication network must provide guaranteed levels of QoS.

The IEC 61158 QoS architecture is well adapted to the DLS type 1, i.e. the TS 61158 Fieldbus, simply because it defines the four classes of DLS as well as all the QoS parameters for the defined attributes. Nevertheless, Nevertheless, it is not the case in the other types of DLS such as ControlNet and WorldFIP.

In the Fieldbus networks, the DLS-users usually are the application layer entities or, in certain cases, the entities of an interface between the data link and the application layers. So, the QoS concept is propagated to the industrial applications from bottom-to-up. Moreover, the DLS-provider is distributed, i.e. the DL entities may be either publishers or subscribers at the different nodes, therefore end-to-end QoS is provided.

It is interesting to note that the IEC QoS parameters take into account some user time-related requirements such as those analyzed in Section 1, i.e. bounded message transfer delay, differentiation between time-critical and no time-critical message and temporal coherence. Nevertheless, the periodicity is not taken into account. In other words, the bounded message transfer delay requirement is taken into account by the DLL maximum confirm delay attribute which is applied to connection-mode data transfer service. The differentiation between time-critical and no time-critical message requirement is taken into account by the DLL priority attribute and the DL scheduling-policy attribute. Temporal coherence requirement is taken into account by the DL timeliness attribute which is only applied to the connection-mode data transfer service. Note that the DL scheduling policy only provides DL user with a means to implicitly or explicitly control any required communication but it does not provide any means to control any periodic communication. Therefore, periodicity requirement is not taken into account.

On the other hand, the IntServ model of the Internet QoS architecture argues that the guaranteed services cannot be given without resource reservation and then a resource reservation protocol is required, such as RSVP. IntServ could be supported by the LANs using the Subnet Bandwidth Management (SBM) approach, if all the traffic passes through at least one priority switch using the IEEE 802.1p, 802.1Q, or 802.1D.

It can be noted that the top-to-bottom Internet QoS architecture and the bottom-to-top IEC 61158 QoS architecture could converge at the data link layer if the priority switch of the SBM approach and the DLL priority attribute of the IEC approach are used. However, as the Section 3.1 has analyzed, the DLL priority attribute is not defined by all the Fieldbus networks. Therefore, new QoS solutions are required and these must be dynamic solutions.

References

Almeida, L., Leon, M., Fonseca, J.A., & Thomesse, J.P. (1999). Real-time Communications in Factory. Proc. *World Multiconference on Systemics, Cybernetics and Informatics* (pp. 119-125). Orlando, USA.

Aurrecoechea, C., Cambell, A., & Hauw, L. (1998). A survey of QoS architectures. *Multimedia Systems*, 6, 138-151.

Braden, R., Clark, D., & Shanker, S. (1994). Integrated Services in the Internet Architecture: an Overview. *Internet RFC 1633*, June.

EN 50170-3. (1995) WorldFIP. General Purpose Field Communication System, *CENELEC EN 50170-3*.

Galara, D., & Thomesse, J.P. (1984). FIP : Proposition dÕun syst me de transmission s rie multiplex e pour les changes dÕinformation entre capteurs, des actionneurs et des automates r flexes. *Minist re de lÕIndustrie et de la Recherche*, France.

Ghanwani, A., Pace, W., Srinivasan, V., Smith, A., & Seaman, M. (2000). A Framework for Integrated Services over shared and switched IEEE 802 LAN Technologies. *Internet RFC 2816*, May.

IEC 61158 (1999). International Electrotechnical Commission. Digital Data Communications for Measurement and Control —Fieldbus for use in Industrial Control Systems. *IEC 61158*.

ISO 13236. (1996). International Standardization Organization. Quality of Service — Framework. *ISO/IEC DIS 13236*.

Seaman, M., Smith, A., Crawley, E., & Wroclawski, J. (2000). Integrated Service Mappings on IEEE 802 Networks. *Internet RFC 2815*, May.

Song, Y-Q. (1991). Etude de Performance de FIP, Aide au Dimensionnement d'Applications. *Ph.D. Th se* , INPL, France.

Thomesse, J.P. (1998). A Review of the FieldBuses. *Annual Reviews in Control*, 22, 35-45.

Xiao, X., & Ni, L.M. (1999). Internet QoS: A Big Picture. *IEEE Network*, 13(2), 8-19.

Yavatkar, R., Hoffman, D., Bernet, Y., Baker, F., & Speer, M. (2000). SBM (Subnet Bandwidth Manager): A Protocol for RSVP-based Admission Control over IEEE 802-style networks. *Internet RFC 2814*, May.

Zhang, L., Deering, S., Estrin, D., Shenker, S., & Zappala, D. (1993). RSVP: A new Resource ReSerVation Protocol. *IEEE Network*, 7(5), 8-11.

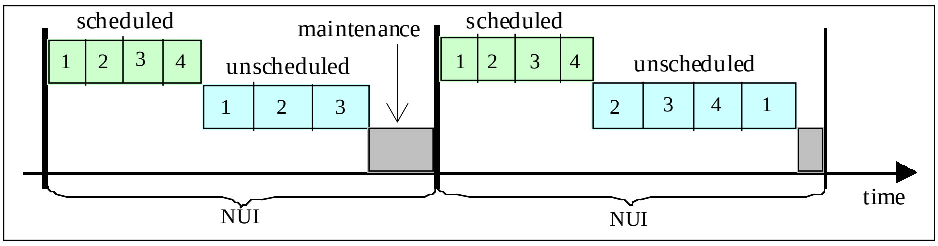

Fig. 1. ControlNet protocol.

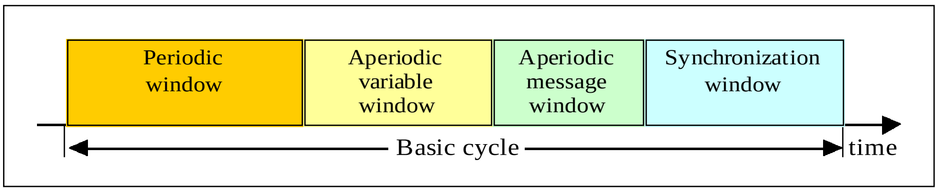

Fig. 2. Basic cycle of the WorldFIP protocol.