Priority Scheduling in Switched Industrial Ethernet

Qizhi Zhang and Weidong Zhang

Abstract—**Switched Ethernet is increasingly advocated as a solution for industrial communication. One typical requirement in such fields is the need to guarantee deterministic transfer delay for cyclic control data, but traditional Ethernet communication does not provide priority feature for industrial data transfer. To address this problem, the IEEE802.1p queuing feature is introduced into both Ethernet switches and communication nodes to give real-time data, especially cyclic control data the priority to access network resources. On the other hand, it is necessary to provide different priority services for real-time data with different urgencies, so EDF (Earliest Deadline First) queuing is also introduced to real-time data queues to provide this feature. Finally the effectiveness of these priority-scheduling methods is verified by a network simulation using OPNET modeler.**

I. Introduction

NOWADAYS, Ethernet is the most widely used local area network technology. Despite originally not being designed for industrial communication, some of its properties, such as easy integration with Internet, inherent compatibility with the management networks used at higher levels in hierarchical industrial systems, and low price make its use in industrial context very attractive. However, the stochastic bus arbitration mechanism adopted by traditional Ethernet (CSMA/CD) has remained a main obstacle for its extensive use in factory floor in the past decade [1].

Currently, using switch technology to enable Ethernet support real-time communication is rapidly becoming a hotspot. When Ethernet adopts micro-segmentation with full duplex transmission mode, each station in it has separate collision domain. The collision problem in IEEE802.3 network is eliminated and simultaneous sending and receiving of data are possible, so the real-time performance of Ethernet is significantly improved.

When moving to factory floor, one fundamental requirement for Ethernet is to provide deterministic transfer delay for cyclic control data. Wang and Chen *et al.* suggested using Ethernet switches with IEEE802.1p queuing feature to give real-time data the priority to access MAC output [2], [3]. However, non-real-time data can still block real-time data at source nodes due to contending for the same MAC output. Therefore, in this paper we suggest introducing the IEEE802.1p queuing feature into MAC of source nodes likewise to avoid this problem.

Concerning the real-time performance evaluation of switched Ethernet, Song and Koubaa evaluated the delay characteristic of switched Ethernet based on queuing theory [4], [5]. Georges *et al.* pointed in [6] the queuing theory method studies the communication system with input services such as Bernoulli or Poisson Processes, which are not representative to the data sent by industrial devices, which typically send real-time data to the network cyclically. Moreover, for multilevel network, the delay calculation by queuing theory method is often difficult even not be possible [7]. Another method to analyze the delay characteristics of switched Ethernet is Network Calculus theory developed by Cruz. He proposed the (σ,ρ)-model and gave an upper bound on delay introduced by a network element in [8] and deduced the delay for a networked topology in [9]. The advantage of Network Calculus method is that it allows the maximum transfer delay to be easily derived.

The rest of this paper is organized as follows. Section II analyzes the data types and all delays a frame may experience in switched Ethernet. Section III introduces IEEE802.1p queuing feature and priority scheduling in switched Ethernet. The calculation of maximum transfer delay for cyclic data based on Network Calculus method is given in Section IV. Section V goes on to verify the effectiveness of priority scheduling in switched industrial Ethernet by simulation. We offer concluding remarks in Section VI.

II. The Delay Analysis of Switched Ethernet

A. Definition of data types in industrial Ethernet

The main problem concerning Industrial Ethernet is how to form methods to support typical industrial real-time

Manuscript received September 15, 2004. This work was supported by National Science Foundation of China (60274032), and Science and Technology Rising-Star program of Shanghai (04QMH1405).

Qizhi Zhang is with Department of Automation, Shanghai Jiao Tong University, Shanghai 200030, P.R. China (Tel/Fax: +86.21.62826946; e-mail: zhangqz@sjtu.edu.cn).

Weidong Zhang was with the Department of Automation, Shanghai Jiao Tong University, Shanghai 200030, P.R. China. He is now with Univ. Stuttgart as a Humboldt researcher, D-7000 Stuttgart, Germany (e-mail: wdzhang@sjtu.edu.cn).

0-7803-9098-9/05/$25.00 ©2005 AACC

traffic without modifying underlying protocols, and while still supporting existing higher-level protocols for non-real-time traffic such as web-based maintenance. Fig. 1. depicts an industrial Ethernet protocol suite model. Priority scheduling is introduced into switches as well as source nodes to get as much real-time performance as possible, at the same time as the use of standard protocols like TCP/IP and Ethernet is preserved.

Recognize that industrial Ethernet has to support not just the need for both real-time control and instrumentation, but also the need to extract information about the plant and its equipment without disturbance to the real-time world. The network traffic can be classified into three types:

1) Cyclic variables are always generated cyclically. In many real-time applications, cyclic variables represent the major computational demands in control system. They typically arise from sensory data acquisition and closed-loop control tasks.
2) Event variables are generated when occurring. They represent events in industrial communication, such as alarms sent from field devices to the central control system.
3) Messages are generated when required, such as download of set-ups, upload of diagnostics and web-based applications.

B. *The delay analysis of switched Ethernet*

The transfer delay of a frame experiences from leaving from source node to being completely received by destination node can be classified into four groups:

1) The delay at source node includes: (a) the protocol process time for encapsulating raw data; (b) the queuing time a frame must wait in buffer when MAC output is busy; (c) the sending time, which depends on the frame length and the MAC output rate. We combine the last two items and call them the buffer response time of a frame at source node.
2) The delay at Ethernet switch includes: (a) The destination table look-up time and switch fabric set-up time, which are the basic delay of Ethernet switch; (b) The queuing time a frame must wait in buffer when the MAC output is busy; (c) The forward time, which depends on the forward mode that Ethernet switch adopts. When store and forward mode is adopted, the forward time is in direct proportion to the frame length. We also combine the last two items and call them the buffer response time of a frame at Ethernet switch.
3) The delay at destination node includes: (a) the receiving time, which depends on the frame length; when the distance between the switch and destination node is not far, the receiving time can be omitted because the sending and receiving almost take place simultaneously; (b) the protocol suite process time for dencapsulating the frame to obtain the raw data.
4) The delay on transfer lines depends on the total line length between source node and destination node. When twisted-pair is adopted, the transfer delay is about $1\mu s$ on a $200m$ line.

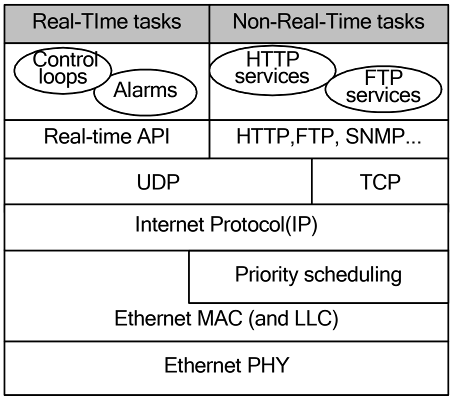

Fig. 1. An industrial Ethernet protocol suite model.

We further divide the above delays into two parts. The first part is those delays that depend on the frame length, the MAC output rate, the basic delay of Ethernet switch, the protocol suite performance of Ethernet nodes, and the length of transfer lines. The second part is the buffer response time a frame experiences in Ethernet switch and source node, which is the most significant part in the whole delay when the network is busy, and is also the main reason for non-deterministic in switched Ethernet.

Traditional switched Ethernet does not provide effective priority transfer mechanism for industrial communication. Because non-real-time data generally have long frame, when a real-time data arrives, if there happens to be a mass of non-real-time data waiting for service, then the real-time data will be delayed significantly. The buffer response time of a frame at source node or Ethernet switch is easily derived as

$$d_{response} = \frac{l_i + \sum_{w=1}^{i-1} l_w}{c}, \qquad (1)$$

where $l_i = m_i + \mu$, $l_w = m_w + \mu$, m_i and m_w are frame lengths of the considered frame and queued frames, respectively, μ is the minimum Ethernet inter-frame gap, and c is the MAC output rate. Because the event variables and messages are sporadic, the number of queued frames w is uncertain when the frame i reaches the MAC buffer. Therefore, the maximum buffer response time of cyclic variables is not bounded.

III. Priority Scheduling in MAC Buffer

The IEEE802.1p queuing feature is firstly introduced in this section, and then the priority scheduling based on IEEE8021.1p in switched industrial Ethernet is described.

A. The IEEE802.1p queuing feature

IEEE802.1 queuing feature endows buffer scheduler with the capability to discriminate different types of data frames. Without priority scheduling, all arrival frames must wait in the same queue if the service node is busy. As a contrast, with priority scheduling, the scheduler can put arrival frames into different queues based on their priorities, and gives a better response time for frames with higher priority.

To use IEEE802.1p queuing feature, IEEE802.1Q Ethernet format must be adopted (Fig. 2.). A new tag with a length of four bytes is inserted between source address and length/type fields of original Ethernet MAC format. The first two bytes are always set to 0x8100, which is called 802.1Q tag type; In the last two bytes, the former three bits define the user priority, the following one bit is CTI (Canonical Format Indicator), and the final 12 bits define the VLAN (Virtual Local Area Network) identifier [10]. IEEE802.1p can define at most eight priority levels. We adopted three priority levels in later simulation. The user priority of cyclic variables, event variables and messages are set to seven, six and five, respectively.

B. The priority scheduling based on IEEE802.1p

Consider that non-real-time frames may block real-time frames at MAC output of Ethernet switches as well as source nodes, so IEEE802.1p queuing feature is introduced into MAC of both Ethernet switches and source nodes. At MAC of source nodes, the arrival frame is firstly encapsulated into IEEE802.1Q Ethernet format before queuing, and the priority field is set based on the frame type, which can be informed by type of service (TOS) field in IP package. The service strategy at MAC output in all source nodes as well as switches is non-preemptive priority service. i.e. cyclic variables are served firstly, event variables are served only if there are no awaiting cyclic variables, and messages are served only if Cyclic variables and Event variables queues are both empty (Fig. 3.)

Consider also different real-time variables usually have different real-time requirements. For example, a cyclic variable with small cycle usually requires less transfer delay compared with those with larger cycles. Therefore, we propose inserting a deadline at the beginning of data segment in cyclic and event variables (Fig. 2.). Real-time API in Fig. 1. can accomplish this task. Then EDF (Earliest Deadline First) queuing feature can be used in both cyclic variables and event variables queues. Those real-time frames with earlier deadlines can be served more quickly, which is a highly desired feature in many industrial applications.

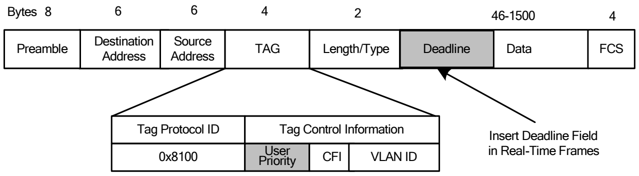

Fig. 2. IEEE802.1Q Ethernet frame adopted by IEEE802.1p.

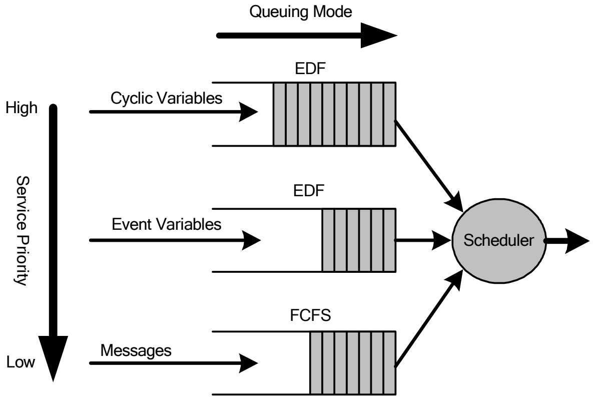

Fig. 3. Priority scheduling based on IEEE802.1p.

C. The Buffer response time of cyclic variables after priority scheduling

Denote m_j, T_j and μ as the frame length and cycle of cyclic variables, minimum Ethernet interframe gap (96bits), respectively. Then the input characteristic of cyclic variables can be described using parameters (σ_j, ρ_j), where j is the priority (less j, higher priority), $\sigma_j = m_j + \mu$ is the burstiness length, $\rho_j = \sigma_j / T_j$ is the average arrival rate. Cruz gave a method to calculate the maximum queuing delay in [9].

The worst-case queuing delay a cyclic variable experiences takes place: when the considered cyclic variable frame is prepared to transfer, there happens to be a frame of other cyclic variables with lower priority but the maximum frame length is just begin to transfer; And the much worse thing is when the cyclic frame with lower priority is transferring, all cyclic flows with higher priority have maximum burstiness. Denote c as the MAC output rate, and then the worst-case queuing delay of a cyclic variable experiences is

$$d_{queue} = \frac{\max(m_{j|j>i}) + \sum_{j=1}^{i-1} \sigma_j + \sigma_i}{c - \sum_{j=1}^{i-1} \rho_j}. \qquad (2)$$

The denominator of (2) is the left bandwidth after cyclic variables with higher priority are served.

When non-preemptive priority service based on IEEE802.1p is implemented in MAC of Ethernet switches and nodes, cyclic variables are delayed by acyclic data at most the time to transfer an acyclic frame with the maximum length $\max(m_e, m_m)$. The worst-case buffer response time of cyclic variables at source nodes or switches is

$$d_{response} = \frac{\max(m_e, m_m)}{c} + \frac{\max(\sigma_{j|j>i}) + \sum_{j=1}^{i-1} \sigma_j + \sigma_i}{c - \sum_{j=1}^{i-1} \rho_j}. \quad (3)$$

IV. Calculation of the Maximum Transfer Delay for Cyclic Variables

For a concrete switched industrial Ethernet model, the calculation method of maximum transfer delay for cyclic variables is given in this section. A two-level switched industrial Ethernet model with master-slaver architecture is assumed (Fig. 4.). The master station simulates the controller, and the slave stations simulate the field devices. All switches adopt store-forward mode. There are $2n$ slave stations, which are symmetrically connected to two second level switches.

The transfer rates of all communication channels are set to c . Assume the protocol process time at source node (encapsulation), the basic delay of switches, the protocol process time at destination node (dencapsulation), and the delay of a single transfer line are τ_1 , τ_2 , τ_3 , and τ_4 , respectively. The maximum buffer response time for cyclic variables at source nodes and switches can be derived following (3). Based on the analysis in Section II, the maximum delay for cyclic variables at each communication node can be derived as follows.

1) The maximum delay at source node is

$$d_{sender} = \tau_1 + \frac{\max(m_e, m_m)}{c} + \frac{\sigma_i}{c}. \quad (4)$$

2) The maximum delay at the second level switch is

$$d_{switch2} = \tau_2 + \frac{\max(m_e, m_m)}{c} + \frac{\max(\sigma'_{j|i+1<j<n}) + \sum_{j=1}^{i-1} \sigma'_j + \sigma'_i}{c - \sum_{j=1}^{i-1} \rho_j}, \quad (5)$$

where $\sigma'_j = \sigma_j + \rho_j \times d_{sender}$.

3) The maximum delay at the first level switch is

$$d_{switch1} = \tau_2 + \frac{\max(m_e, m_m)}{c} + \frac{\max(\sigma''_{j|i+1<j<2n}) + \sum_{j=1}^{i-1} \sigma''_j + \sigma''_i}{c - \sum_{j=1}^{i-1} \rho_j}, \quad (6)$$

where $\sigma''_j = \sigma_j + \rho_j \times (d_{sender} + d_{switch2})$.

4) The maximum delay at receive node is

$$d_{reciever} = \frac{\sigma_i}{c} + \tau_3. \quad (7)$$

5) The delay on transfer lines is

$$d_{lines} = 3\tau_4. \quad (8)$$

Summating all delays at each service node gives the maximum transfer delay for cyclic variables:

$$d_{\max} = d_{sender} + d_{switch2} + d_{switch1} + d_{reciever} + d_{lines} \quad (9)$$

The calculated result can be improved by methods introduced by George [6] or Koubaa [11], but the calculation process will be more complicated.

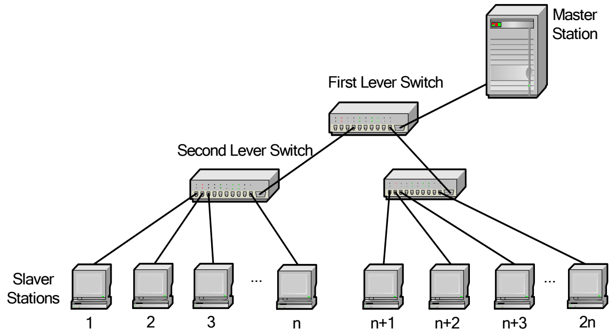

Fig. 4. A switched industrial Ethernet model with tree topology.

V. Simulation Research

We use OPNET Modeler [12] to simulate the switched industrial Ethernet model depicted in Fig. 4. The user priorities of data are set at source node, and priority scheduling based on IEEE802.1p is implemented by modifying the MAC process model of Ethernet switches and stations. EDF queuing feature is implemented in cyclic variables queues and event variables queues. At source nodes, cyclic variables, event variables, and messages are generated using three different source modules. The destination nodes use three corresponding sink modules to gather the corresponding arrival frames, which are used to derive the simulation results.

All MAC outputs adopt $10M$ rate, and the switches adopt store-forward mode. The lengths of all transfer lines are set to $200m$. Because the load of the line between the first level switch and master station is the heaviest, it is enough consider the load of this line to estimate the load of the whole network. Define that cyclic variables account for 80% of the total communication load, event variables and message accounts for 10%, respectively. The number of slave stations is set to 30. The cyclic and event variables adopt minimum Ethernet frame length, whereas messages adopt maximum Ethernet frame length.

Considering that the minimum frame length of IEEE802.1Q Ethernet format is 64 bytes, the maximum frame length is 1522 bytes, the synchronous code is eight bytes, and the interframe is 12 bytes, to saturate the line between the first level switch and the master station, each slave station should generate $\frac{10Mbits \times 80\%}{[(64+8) \times 8bits + 96bits] \times 30} = 416.1/\mathrm{s}$ frames destined to the master station (*the generate cycle is 0.0024s*). At the same time each slave station should generate $\frac{10Mbits \times 10\%}{[(64+8) \times 8bits + 96bits] \times 30} = 52.01/\mathrm{s}$ event variables (*the generate interval is exponential distribution with parameter 0.0192s*) and

$\frac{10Mbits\times 10\%}{[(1522+8)\times 8bits+96bits]\times 30}=2.83/\mathrm{s}$ messages (*the generate interval is exponential distribution with parameter 0.353s*), and all have the master station as the destination.

When 30 slave stations all send data to the master station, the network is unstable, so the maximum number of slave stations connected to the network is set to 28. Change the number of slave stations connected to the network, and collect the average delay and maximum delay of three types of frames. The simulation results are depicted in Fig.5. It can see that priority scheduling achieves desired results. The transfer delays of cyclic variables are always has the minimum values, and the transfer delays of event variables are also better than those of messages.

Using (4-9) we can verify the maximum delay of cyclic variables. For example, when there are 20 slaves station connected to the network, we can obtain the respective delays at each service nodes:

$d_{sender}=50\mu s+(1530\times 8+84\times 8)/10M=1.34\times 10^{-3}\mathrm{s};$

$d_{switch2}=70\mu s+\frac{1530\times 8}{10M}+\frac{10\times 128.7\times 8}{10M-9\times 2.67\times 10^{5}}=2.65\times 10^{-3}s;$

$d_{switch1}=70\mu s+\frac{1530\times 8}{10M}+\frac{20\times 217.0\times 8}{10M-19\times 2.67\times 10^{5}}=8.33\times 10^{-3}\mathrm{s};$

$d_{reciever}=50\mu s$;

$d_{lines}=3\times 1\mu s=3.0\times 10^{-6}s.$

The calculated maximum transfer delay for cyclic variables is 12.3ms. It is much larger than 6.5ms obtained from the simulation result. The deviation is due to the fact that, in the simulation the bursts on the different connections are not necessary simultaneous, whereas Network Calculus admits simultaneous bursts, which is the worst-case. When the simulation time is infinite, the simulation result will approach the calculated result.

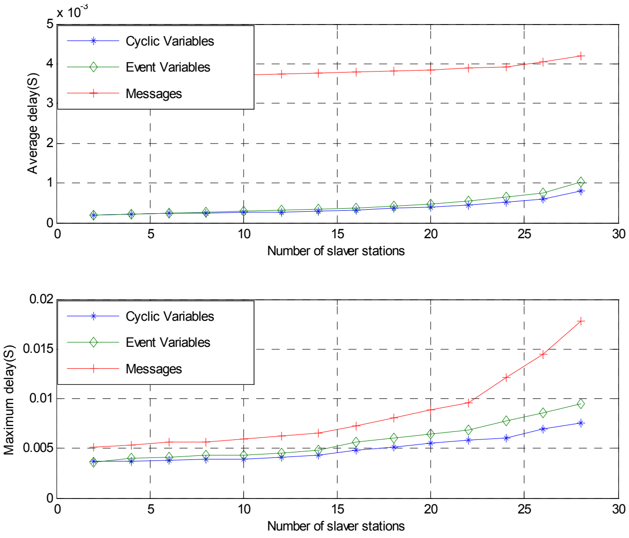

Fig. 5. The average delays and the maximum delays change as the connected slave stations change.

VI. Conclusion

In this paper we show the priority scheduling based on IEEE802.1p is a good choice to enhance the real-time performance of switch industrial Ethernet, which can prevent real-time data from delayed due to contending for the same MAC output with non-real-time data. We also propose inserting a deadline field into real-time frames to achieve EDF queuing feature in real-time queues, which is a desirable feature in industrial communication. For a concrete industrial Ethernet model, the paper also gives the method based on Network Calculus to estimate the maximum transfer delay of cyclic variables. Finally a simulation is carried out to verify the effectiveness of the proposed priority scheduling method.

References

[1] P. Pedreiras, R. Leite, and L. Almeida, “Characterizing the real-time behavior of prioritized switched Ethernet,” in *Proc. of 2nd International Workshop on Real-Time LANs in the Internet Age*, Oporto, Portugal, 2003.

[2] Z. Wang, Y. Song, J. Chen, and Y. Sun, “Real-time characteristics of Ethernet and its improvement,” in *Proc. of the 4th World Congress on Intelligent Control and Automation*, Shanghai, P.R. China, 2002.

[3] J. Chen, Z. Wang, and Y. Sun, “Switch real-time industrial Ethernet with mixed scheduling policy,” presented at The 28th Annual Conference of IEEE Industrial Electronics Society, Sevilla, Spain, 2002.

[4] Y. Song, “Time constrained communication over switched Ethernet,” presented at 4th IFAC International Conference on Fieldbus Systems and Their Applications, Nancy, France, 2001.

[5] Y. Song and A. Koubaa, “Switched Ethernet for real-time industrial communication: modeling and message buffering delay evaluation,” presented at 4th IEEE International Workshop on Factory Communication Systems, Vasteras, Sweden, 2002.

[6] J. P. Georges, E. Rondeau, and T. Divoux, “Evaluation of switched Ethernet in an industrial context by using the Network Calculus,” presented at 4th IEEE International Workshop on Factory Communication Systems, Sweden, 2002.

[7] J. Jasperneite, P. Neumann, M. Theis, and K. Watson, “Deterministic real-time communication with switched Ethernet,” presented at 4th International Workshop on Factory Communication Systems, New York, NY, 2002.

[8] R. L. Cruz, “A calculus for transfer delay, part II: network analysis,” *IEEE trans. on Information Theory*, vol. 37, pp. 132-141, 1991.

[9] R. L. Cruz, “A calculus for transfer delay part I: network elements in Isolation,” *IEEE trans. on Information Theory*, vol. 37, pp. 114-131, 1991.

[10] X. Xie, *Computer network,* 4 ed. Beijing: Publishing House of Electronics Industry, 2003, pp. 389-390.

[11] A. Koubaa and Y. Song, “Upper bound evaluation of response time for real-time communication,” presented at 11th Conference RTS Embedded Systems, Paris, 2003.

[12] OPNET Modeler, 8.0 ed. OPNET Technologies Inc., 2001.