Proceedings of the European Control Conference 2007
Kos, Greece, July 2-5, 2007

Timing Analysis and Response Time of End to End Packet Delivery in Switched Ethernet Network

Vahid Hassani, H.A. Talebi, Masoud Shafiee, and Hassan Taheri

Abstract—**This paper presents an approach for timing analysis and response time of end to end packet delivery for Ethernet Networked Control System (ENCS). Ethernet is a popular network for control applications. However, traditional Ethernet network suffers from non-deterministic handling of the network communication caused by Carrier Sense Multiple Access Collision Detection (CSMA/CD) Protocol, where the nodes compete for access to the media. In this paper, a collision free network configuration model for Ethernet is used which is more suitable for control application as compared to the traditional network configurations. The office network is separated from the control network and a fully switched Ethernet network is employed for control network. The timing behaviour of a fully switched network is analysed and the response time of end to end packet delivery in switched Ethernet network is derived. This analysis can then be served as a time delay model for NCS.**

Key words— **Switched Ethernet Network, Full-Duplex Mode, Timing Analysis, Time Response.**

I. INTRODUCTION

Networked Control Systems (NCS) are feedback control systems in which the control loops are closed through a real-time network. These systems have advantages such as lower installation costs, increased flexibility, and rapid installation. However communication delay is a general problem in NCS, and it can destabilize the closed loop system. Ethernet is a popular network for control applications. Its single channel for communications results in fast speed, low cost and easy installation. However, traditional Ethernet networks are not suitable for real-time applications due to the non-deterministic handling of the network communication caused by CSMA/CD, access control protocol where the nodes compete for access to the media [4]. When two nodes try to send their messages simultaneously, a collision occurs, the transmission will cease and after a random amount of time, a retransmission will be tried. The most popular method of collision-avoidance, which presents a ***deterministic Ethernet***, is to introduce single collision domains for each node, thereby eliminating access contention. This is achieved by implementing a full duplex switched Ethernet Network.

The first author was with Amirkabir University of Technology; he is now with the Institute for Systems and Robotics (ISR) of IST, Lisbon, Portugal where he is pursuing his PhD studies. His work was partially supported by Iran Telecommunication Research Center (ITRC) (vahid@isr.ist.utl.pt) and the FCT-ISR/IST plurianual funding program (through the POS-Conhecimento Program initiative, in cooperation with FEDER).

H.A. Talebi, H. Taheri, and M. Shafiee are with EE department of Amirkabir University of Technology, Tehran, Iran.
{alit, shafiee, htaheri@aut.ac.ir}

Switched Ethernet has been used for real-time applications in [4], [5], [8], [9]. Periodic scheduling of real-time traffic in a switched network has been suggested in [4] to avoid the non-deterministic behaviour of Ethernet network. In [9], a soft real-time packet scheduling algorithm for MAC-layer in switched Ethernet network has been presented. In [10], [11], [16] the issue of modelling the Ethernet switch and performance evaluation of its real-time features have been considered, but the buffer overflow problem has not been addressed. In [12], the basic concepts of Ethernet's capability to deliver a real-time communication system has been introduced and in [13] some of the real-time solutions, available to industry to date are presented, but, the key feature of a real-time system, the ***timing analysis*** left intact. Flow control and congestion avoidance in switched Ethernet LANs has been considered in [8] with no timing analysis. In [14], the queuing delays inside the switch for periodic control system applications were estimated, but the buffer overflow problem was not considered. In [15], the influence of the service strategies in switched Ethernet on delays is investigated by simulation when the worst case source behaviour could be characterized. A comparison of the worst case analysis for end to end delay in different switched Ethernet architectures has been given in [17], but the effect of Pause operation has not been considered. In [18], design and evaluation methodology for a switched Ethernet architecture was proposed. They have used a known maximum end to end delay in real-time applications; In [19], new protocol for real-time applications in Switched Ethernet was presented. In [20], [21], an upper bound for maximum end to end delays of time-critical messages over the network was derived.

In this paper, a collision free network configuration model for Ethernet is used which is more suitable for control applications as compared to the traditional network configurations. The office network is separated from the control network and a fully switched Ethernet network is employed for control network. The timing behaviour of a fully switched network is analysed and the response time of end to end packet delivery in switched Ethernet network is derived. This analysis can then be served as a time delay model for NCS. The rest of the paper is organized as follows: In Section II, we present a traditional configuration of Ethernet and demonstrate its disadvantages in control application. Then, another network configuration is presented which is more suitable for control applications. In Section III, the timing analysis problem is defined and the main results of the paper, namely the timing analysis and deriving the response time for a set of successfully transmitted massages are given in Sections IV and V. A numerical example is presented in Section VI, and Section VII concludes the paper.

II. Network Configuration

A. Traditional Configuration

Fig. 1 shows a traditional networked feedback control system composed of sensors, actuators and controller attached to personal computers, printers and internet. It is visible that non-control traffic will affect the performance of the control system. Sending large files through the network, would utilize most network's capacity, and would cause slow transmission for other parties. In this scheme, entire network's users can be affected by a single intensive user. As a result, network delay among sensors, actuators and controller will be increased.

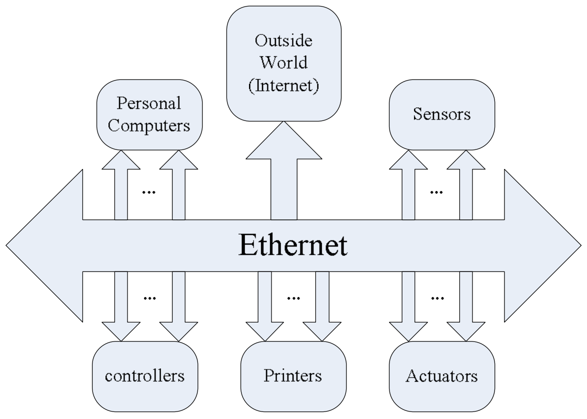

Fig. 1: Traditional Configuration

B. Networked Control Configuration

In this section, an alternative configuration is presented which will prevent an intensive user affect the entire network and will be more suitable for timing analysis. The large messages in the network correspond to non-control part and usually control messages are not large, hence an effective solution is to separate the control system from other parts of the network. We will use two sub-networks, office network and control network. The connection between the two networks and outside world similar to the Internet is established through a router. This scheme is shown in Fig. 2. In this scheme, the only device seeing all messages is the router and it will divide the control traffic and non-control traffic in two sub-networks by looking at the recipient's address. Therefore, the transmission between the two networks is done only when it is needed.

Suppose that a sensor is sending data to the controller, then all other parts in the control network that need data transfer have to wait until the bus becomes idle. This behaviour is called half-duplex that means data can only be transmitted in one direction at a time. Using a Local Area Network (LAN) switch (Fig. 3) increases the efficiency of the network. In this case all hubs of the Ethernet network are replaced with dedicated segments for every node. Each switch can support up to a hundred of these segments. The switch and the node are the only components that are connected to each segment. Hence, the switch can pick up every transmission before it reaches to other nodes and forward the frame to the appropriate segment. This means that, the message only reaches the planned recipient. In all switches, there are some high speed buffer memories to store frames before forwarding to proper segment. This mode of switching is called, store and forward. In this mode, faulty messages are not forward to the receiver because frames will be checked before the transmission. Another advantage of this configuration is that collision will not occur between different nodes. This network is called full duplex which is a collision free network. The policy of service in buffer is First in First out (FIFO). Messages more than buffer capacity will lead to buffer overflow. Similar configurations were also suggested in [4]-[6], [8], [10]. It has been shown in [4], [6], [8] that the advance configurations can increase the network's efficiency and eliminate data collisions.

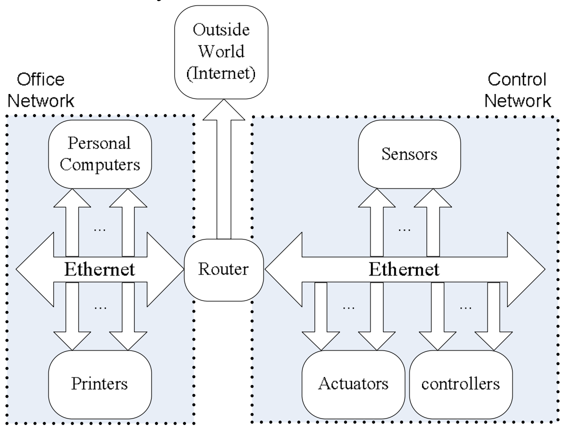

Fig. 2: Advance Configuration

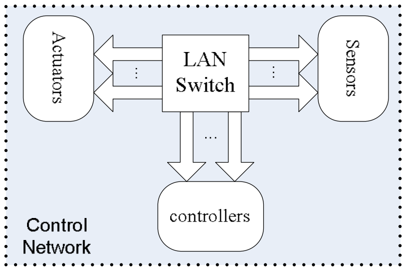

Fig. 3: Cotrol Network with Switch

III. Problem Definition

As mentioned in the previous section, each segment of the network has only two components, LAN switch and station. Moreover, in the case of buffer overflow, some messages would be dropped. Fortunately, 802.3 [2], [3] has the optional MAC control portion which makes it possible to be employed in a real-time manner in the Ethernet station. A special operation of PAUSE [2], [3], resolves buffer overflow problem in full duplex mode. The mechanism of this command is a simple "stop-wait-restart" form of flow control. In Fig. 4, a linear buffer with PAUSE operation is shown. When a node receives a PAUSE frame, it stops sending data frames for a specific time that is mentioned in the PAUSE frame. After expiring this time, sending data frame is restarted by the sender node from stopping point. The PAUSE command can be cancelled by another PAUSE frame with a parameter of zero time and similarly PAUSE period can be extended by another PAUSE frame with a non-zero time, before expiring the first PAUSE period. As can be seen in Fig. 4, when the buffer fills up to a predetermined high level, a PAUSE frame is send to prevent packets from being dropped, before buffer becomes full. After expiring PAUSE period or when the buffer empties below a predetermined low level, the PAUSE would be finished or cancelled (by sending a PAUSE command with zero time). In this case we use maximum capacity of switch without dropping any packet.

In sequel, we present the main results of the paper, i.e. analyzing the timing behaviour of the network.

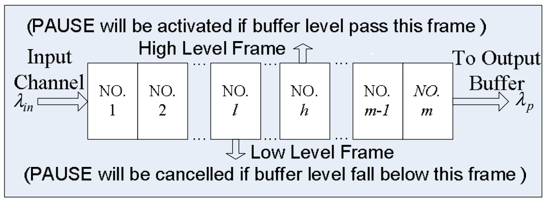

Fig. 4: Input Buffer with PAUSE Command

IV. Timing Analysis

We assume that all sensors in control network have an equal sampling time, hence the n stations transmit q packets to a target simultaneously and buffers can store m packets. We also assume that the input bandwidth λ_{in} *(frames / second)*, output bandwidth λ_{out} and switch's processing speed λ_p are known. Borrowed from queuing theory [1], the G/G/1 model is used for the switch which allows us to assume that each buffer is processed by a switch at a rate of λ_p / n. As mentioned in the previous sections, if the output buffer is full, transferring packets is stopped until some packet leaves the switch. On the other hand, if the input buffer reaches an upper level h, it will transmit a PAUSE frame to the sender and only one more message will be received from the sender due to the PAUSE frame propagation delay. Transmission cannot be started until the input buffer reaches a lower level l or the PAUSE frame times out.

Output Buffer

Output buffer will become full when the switch processing rate is higher than the output port's rate ($\lambda_p > \lambda_{out}$) and it will never fill up otherwise. Hence, the time takes for an output buffer with size m to be filled up can be expressed as:

$$T_{fill-o} = \begin{cases} \dfrac{m}{\lambda_p - \lambda_{OUT}} & if\ \lambda_p > \lambda_{out} \\ \infty & otherwise \end{cases} \tag{1}$$

Input Buffer

To assess the behaviour of the input buffer, four different working conditions can be considered:

1. Neither output buffers nor input buffers get full.
2. Input buffers never fill up but the output buffers get full.
3. Input buffers fill up but output buffers never fill up.
4. Both buffers fill up.

A. Case I

In this case, we have $\lambda_{out} \geq \lambda_p$ and $\lambda_{in} \leq (\lambda_p / n)$, the required time to receive q packets from each sender is q / λ_{in}.

B. Case II

In this case, after the output buffer gets full, it can receive another frame from the input buffers only after transmitting a frame to the receiver. Therefore, the equivalent processing rate can be given as:

$$\frac{1}{\lambda_{eq}} = \frac{1}{\lambda_{out}} + \frac{1}{\lambda_p} \tag{2}$$

As we described in the previous section, each buffer is processed by a switch at a rate of λ_{eq} / n. The time required for an input buffer to receive q packets from each sender is then q / λ_{in} and we have $\lambda_{in} \leq (\lambda_{eq} / n)$.

C. Case III

The time required for the input buffer to reach the upper level is $T_h = h / (\lambda_{in} - \lambda_p / n)$. The Pause frame will then be sent to the sender after this time and the transmission will be restarted after the buffer reaches the lower level or PAUSE time expires. This time can be calculated as

$$T_r = \min\left\{ \frac{h - l + 1}{(\lambda_p / n)}, T_{pause} \right\} \tag{3}$$

Since $(\lambda_p / n) < \lambda_{in}$, the input buffer will reach the upper level again and oscillate between the upper and lower level. The time required for the input buffer to reach the upper level again is:

$$T_{r-h} = \frac{\lfloor T_r(\lambda_p / n) \rfloor}{\lambda_{in} - (\lambda_p / n)} \tag{4}$$

The floor function in this equation indicates that the switch does not accept or store partial frames.

D. Case IV

In this case, after the output buffer gets full, it can only receive a frame from the input buffers, after it transmits a frame to the receiver. The equivalent processing rate can then be obtained as (2) where each buffer is processed by a switch at a rate of λ_{eq} / n. In the event that the output buffer just gets full, $n_i = \lfloor T_{fill-o} \times (\lambda_{in} - (\lambda_p / n)) \rfloor$ frames exist in the input buffer. The time required for the input buffer to reach to the upper level is given by

$$T_h = \begin{cases} h/(\lambda_{ex}) & if \quad h/(\lambda_{ex}) < T_{fill-o} \\ (h-n_i)/\lambda_{ex} + T_{fill-o} & otherwise \end{cases} \quad (5)$$

where $\lambda_{ex} = \lambda_{in} - (\lambda_p / n)$. The first case in (5) is the case where the input buffer fills up faster than the output buffer and second case is the case where the input buffer fills up slower than the output buffer. The input buffer will reach to the lower level but the required time for this is different at different times, since the input buffers' levels vary between the lower and upper levels. We will denote T_r^i as the i th time for the sender to restart the transmission:

$$T_r^i = \begin{cases} t_1 & if\ T_h + t_3 \ge T_{fill-o} \\ min\{T_{pcd}^i, T_{pause}\} & if\ T_h + t_3 + t_2 > T_{fill-o} \\ t_2 & otherwise \end{cases} \quad (6)$$

where $t_1 = \min\{(h-l+1)/(\lambda_{eq}/n), T_{pause}\}$, and $t_2 = \min\{(h-l+1)/(\lambda_p/n), T_{pause}\}$, and $t_3 = \sum_{k=0}^{i-1}(T_r^k + T_{r-h}^k)$, where T_{r-h}^k is the time required to reach the upper level for the i th time, T_{r-h}^0 is assumed to be zero, and t_3 is the time between the first and the i th time that the buffer reaches the upper level h. In equation (6), the first case is related to the situation where the output buffer fills up before sending the PAUSE frame, the second case is related to the situation where the output buffer fills up after receiving r frames and T_{pcd}^i is given by:

$$T_{pcd}^i = \frac{r}{\lambda_p / n} + \frac{h-l-r+1}{\lambda_{eq}/n} \quad (7)$$

where r is the smallest integer satisfying:

$$T_h + \sum_{k=1}^{i-1}(T_r^k + T_{r-h}^k) + (r+1)/(\lambda_p/n) > T_{fill-o} \quad (8)$$

The third case in (6) is the case where the output buffer will not fill up during this high to low level process. To compute the time required to reach the upper level, the level of the input buffer before the senders restart the transmission shall be known. We will denote n_{inp}^i as the level of the input buffers after the i th pause time:

$$n_{inp}^i = \begin{cases} n_1 & if\ T_h + t_3 \ge T_{fill-o} \\ n_{pcd}^i & if\ T_h + t_3 + t_2 > T_{fill-o} \\ n_2 & otherwise \end{cases} \quad (9)$$

where $n_1 = max\{l, h+1-\lfloor Tpause(\lambda_{eq}/n)\rfloor\}$ and $n_2 = max\{l, h+1-\lfloor Tpause(\lambda_p/n)\rfloor\}$.

In equation (9), the first case is the case where the output buffer is full. The second case is the case where the output buffer fills up in the pause period time and is described in equation (10). The third case is related to the case where the output buffer is not full. In equation (9), n_{pcd}^i is given by:

$$n_{pcd}^i = \begin{cases} h+1-\lfloor T_{pause}(\lambda_p/n)\rfloor & if\ T_{pause} \le t_{11} \\ n_{22} & if\ t_{11} < T_{pause} \le t_{12} \\ l & otherwise \end{cases} \quad (10)$$

where $n_{22} = h+1-r-\lfloor (T_{pause} - t_{11})(\lambda_{eq}/n)\rfloor$ and r can be found from equation (8), $t_{12} = r/(\lambda_p/n) + (h+1-r-l)/(\lambda_{eq}/n)$, and $t_{11} = r/(\lambda_p/n)$.

In equation (10), the first case is related to the case where the pause expires before the output buffer is filled up, the second case is the case where the pause expires after the output buffer is filled up, and the third case is related to the case where the input buffer returns to the low level before the pause expires. In a similar manner, we can calculate T_{r-h}^k:

$$T_{r-h}^i = \begin{cases} t_{13} & if \quad T_h + t_{33} \ge T_{fill-o} \\ T_{pcd}^i & if \quad T_h + t_{33} + t_{23} > T_{fill-o} \\ t_{23} & otherwise \end{cases} \quad (11)$$

where $t_{13} = (h - n_{inp}^i)/(\lambda_{in} - \lambda_{eq/n})$ and $t_{23} = (h - n_{inp}^i)/(\lambda_{in} - \lambda_{p/n})$, and $t_{33} = \sum_{k=1}^{i}(T_r^k + T_{r-h}^{k-1})$.

In equation (11), the first case is the case where output buffer has already filled up, the second case deals with the case that processing rate varies and the last case is related to the case where the output buffer will not get full, during this resume transmission to the high level. Similar to equation (6), T_{r-h}^0 is assumed to be zero and T_{pcd}^i is given by:

$$T_{pcd}^i = \frac{r}{\lambda_{in} - \lambda_p/n} + \frac{h-r-n_{inp}^i}{\lambda_{in} - \lambda_{eq}/n} \quad (12)$$

where r is the smallest integer satisfying:

$$T_h + \sum_{k=1}^{i-1}(T_r^k + T_{r-h}^{k-1}) + \frac{(r+1)}{(\lambda_p/n)} > T_{fill-o} \quad (13)$$

This is related to the case where the output buffer fills up after receiving r frames.

V. Time Response

After timing analysis for the advance configuration, we can now derive the time required for n sensors to transmit q packets through a switch to the controller at the same time. We can divide the time response into five sections:

1. T_1, the time required for travelling the first bit of data from the sender to the switch (propagation delay).

2. T_2, the time required for receiving all packets in the switch.

3. T_3, the time for clearing all packets in the input buffer.

4. T_4, the time required for clearing all packets in the output buffer.

5. T_5, the time required for travelling the last bit of data from switch to the receiver (propagation delay). It is clear that T_1 and T_5 depend on the physical distance. Now, similar to the previous section, four cases are considered:

A. Case I

In this case, the input and output buffers never fill up and PAUSE operation would not be activated. Hence, T_2 is equal to q/λ_{in}. When the input buffer receives the last bit of data, only one more frame will be left in the input buffers, so T_3 and T_4 are equal to $1/(\lambda_p/n)$ and $1/\lambda_{out}$, respectively. Then the total response time will be given as $T = T_1 + T_2 + T_3 + T_4 + T_5$.

B. Case II

In this case, the input buffers never fill up but the output buffer fills up. The output buffer can receive a frame from the input buffers, only after transmitting a frame to the receiver. Therefore, the equivalent processing rate can be computed as equation (2). Input buffers never fill up and T_2 will be equal to q/λ_{in}. When the input buffer receives the last bit of data, only one more frame will be left in the input buffers, hence T_3 can be expressed as:

$$T_3 = \begin{cases} 1/(\lambda_p/n) & \text{if } T_2 < T_{fill-o} \\ 1/(\lambda_{eq}/n) & \text{otherwise} \end{cases} \tag{14}$$

where T_4 is given by:

$$T_4 = \begin{cases} \dfrac{\lfloor (T_2+T_3)(\lambda_p - \lambda_{out}) \rfloor}{\lambda_{out}} & \text{if } T_2 + T_3 < T_{fill-o} \\ m/\lambda_{out} & \text{otherwise} \end{cases} \tag{15}$$

Then the total response time will be given by $T = T_1 + T_2 + T_3 + T_4 + T_5$.

C. Case III

In this case, the input buffers fill up but the output buffer never fills up. Then, the equivalent processing rate can be computed as equation (2). As mentioned before, two operation modes, namely pause and transmit can be considered. Hence, we can compute the time required for each transmit cycle, and the number of messages which are transmitted during one cycle. The number of cycles required for n senders to transmit q packets from each node to the switch can be computed as the smallest integer k satisfying :

$$\lfloor T_h \lambda_{in} \rfloor + \sum_{i=o}^{k} \lfloor T_{r-h}^{i} \lambda_{in} \rfloor \geq q \tag{16}$$

Now, T_2 can be calculated as:

$$T_2 = \begin{cases} T_h + \sum_{i=0}^{k-1} (T_{r-h}^{i} + T_r^{i}) + T_r^{k} + \dfrac{R_{es}}{\lambda_{in}} & \text{if } k > 0 \\ q/\lambda_{in} & \text{otherwise} \end{cases} \tag{17}$$

where k is defined in (16) and $R_{es} = q - \lfloor T_h \lambda_{in} \rfloor - \sum_{i=0}^{k-1} \lfloor T_{r-h}^{i} \lambda_{in} \rfloor$ which describes the number of remaining messages after $k-1$ cycles, T_h is the time required for the input buffer to reach the upper level for the first time and $\sum_{i=0}^{k-1} (T_{r-h}^{i} + T_r^{i})$ $\sum_{i=0}^{k-1} \lfloor T_{r-h}^{i} \lambda_{in} \rfloor$ gives the time between the first and k th time, that the input buffer reaches the upper level. In order to calculate T_3, we need to know the number of frames in the buffer when the switch receives the last frame:

$$n_{t3} = \begin{cases} \left\lfloor \dfrac{q}{\lambda_{in}} (\lambda_{in} - \dfrac{\lambda_p}{n}) \right\rfloor & \text{if } \lfloor T_h \lambda_{in} \rfloor \geq q \\ h + 1 - \left\lfloor T_r \dfrac{\lambda_p}{n} \right\rfloor + \left\lfloor \dfrac{R_{es}}{\lambda_{in}} (\lambda_{in} - \dfrac{\lambda_p}{n}) \right\rfloor & \text{otherwise} \end{cases} \tag{18}$$

The first case in this equation is related to the case where the switch has received q frames, before the input buffer reaches the upper level for the first time and $h + 1 - \left\lfloor T_r \dfrac{\lambda_p}{n} \right\rfloor$ is the number of frames in the input buffer after $(k-1)$ th Pause time. We can calculate the number of frames which will be received by the input buffer after $(k-1)$ th Pause time, by $\left\lfloor \dfrac{R_{es}}{\lambda_{in}} (\lambda_{in} - \dfrac{\lambda_p}{n}) \right\rfloor$. Now, T_3 can be expressed as:

$$T_3 = \frac{n_{t3}}{(\lambda_p/n)} \tag{19}$$

When the output buffer receives the last bit, only one more frame will be left in the output buffer, so T_4 is equal to $(1/\lambda_{out})$.

Then the total response time will be given as $T = T_1 + T_2 + T_3 + T_4 + T_5$.

D. Case IV

In this case, when the output buffer fills up, it can receive a frame from the input buffers only after transmitting a frame to the receiver. Therefore, the equivalent processing rate can be computed as (2). We can assume each buffer is processed by a switch at a rate of λ_{eq}/n [1]. The input buffers will fill up if $\lambda_{in} > (\lambda_{eq}/n)$. Noting the two modes of operations, i.e. pause and transmit, the required number of cycles for n senders to transmit q packets from each node to the switch can be computed as the smallest integer k satisfying:

$$\lfloor T_h \lambda_{in} \rfloor + \sum_{i=o}^{k} \lfloor T_{r-h}^{i} \lambda_{in} \rfloor \geq q \tag{20}$$

Now, T_2 can be calculated as:

$$T_2 = \begin{cases} T_h + \sum_{i=0}^{k-1} (T^{\,i}_{r-h} + T^{\,i}_{r}) + T^{\,k}_{r} + \dfrac{R_{es}}{\lambda_{in}} & if\ k > 0 \\ q / \lambda_{in} & otherwise \end{cases} \tag{21}$$

where k is defined in (20) and $\lfloor T_h \lambda_{in} \rfloor + \sum_{i=o}^{k} \lfloor T^{\,i}_{r-h} \lambda_{in} \rfloor \geq q$, which describes the number of remaining messages after $(k-1)$ cycles. In order to obtain T_3, we need to know the number of frames in the buffer when the switch receives the last frame:

$$n_{t3} = \begin{cases} n^{k}_{in} + \left\lfloor \dfrac{R_{es}}{\lambda_{in}} (\lambda_{in} - \lambda_p / n) \right\rfloor \\ \quad if\ k \neq 0 \,\&\, T_2 < T_{fill-o} \\ \left\lfloor \dfrac{q}{\lambda_{in}} (\lambda_{in} - \lambda_p / n) \right\rfloor \\ \quad if\ k = 0 \,\&\, T_2 < T_{fill-o} \\ \left\lfloor (\lambda_{in} - \dfrac{\lambda_p}{n}) T_{fill-o} \right\rfloor + \left\lfloor (\dfrac{q}{\lambda_{in}} - T_{fill-o})(\lambda_{in} - \dfrac{\lambda_{eq}}{n}) \right\rfloor \\ \quad if\ k = 0 \,\&\, T_2 \geq T_{fill-o} \\ n^{k}_{in} + \left\lfloor \dfrac{R_{es}}{\lambda_{in}} (\lambda_{in} - \lambda_{eq} / n) \right\rfloor \\ \quad if\ k \neq 0 \,\&\, T_2 - \dfrac{R_{es}}{\lambda_{in}} > T_{fill-o} \\ n^{k}_{in} + \left\lfloor \dfrac{R_{es} - n_{eq}}{\lambda_{in}} (\lambda_{in} - \lambda_p / n) \right\rfloor + n_{eq} \\ \quad otherwise \end{cases} \tag{22}$$

Where $n_{eq} = \lfloor (T_2 - T_{fill-o})(\lambda_{in} - \lambda_{eq} / n) \rfloor$ is the number of frames received by the input buffer after the output buffer filled up and k is defined in (20). In equation (22), the first case is the case where the output buffer is not full and PAUSE is active in the input buffers. The second case is the case where the output buffer is not full and PAUSE is not active in the input buffers, the third case is related to the case where the output buffer is full and PAUSE is not active in the input buffers. The fourth case is the case where the output buffer becomes full in the last $k-1$ cycles and PAUSE has been activated in the input buffers. The last case is the case where the output buffer becomes full in the k th cycle and PAUSE is active in the input buffers. Now T_3 can be expressed as:

$$T_3 = \begin{cases} \dfrac{n_{t3}}{(\lambda_p / n)} & if\ T_2 + \dfrac{n_{t3}}{(\lambda_p / n)} \leq T_{fill-o} \\ \dfrac{k}{(\lambda_p / n)} + \dfrac{n_{t3} - k}{(\lambda_{eq} / n)} & otherwise \end{cases} \tag{23}$$

where k is smallest integer satisfying $T_2 + (k+1)/\lambda_p > T_{fill-o}$. After transmitting all frames from the input buffers to the output buffer, the number of remaining frames in the output buffer would be equal to $min\{ \lfloor (T_2 + T_3)(\lambda_p - \lambda_{out}) \rfloor , m \}$. Hence, T_4 can be expressed as:

$$T_4 = \frac{min\{ \lfloor (T_2 + T_3)(\lambda_p - \lambda_{out}) \rfloor , m \}}{\lambda_{out}} \tag{24}$$

Then the total response time for one transmitting cycle will be given as $T = T_1 + T_2 + T_3 + T_4 + T_5$.

VI. Numerical Example

Consider the control network shown in Fig. 2 with 10 sensors and 1 controller. Input and output ports have been connected to 100 Mbps Ethernet cables with the effective bandwidth of 99.1 Mbps (effective bandwidth is the amount of information except the erroneous frames, transmitted in one second), each frame has a size of 12204 bits (maximum frame size, λ_p is equal to 14880 frames per second (using Cisco Catalyst 6500 Ethernet switch), size of the buffers is equal to 16 frames and the low level and the high level in the buffers are at 9 and 14 frames, respectively. The total response time, if each of these 10 sensors transmits 50 frames to a controller, would be obtained as 0.107 seconds. The results are summarized in Table I.

Table I. Time Response

	time
T_1	5 μs
T_2	86.22ms
T_3	19ms
T_4	2ms
T_5	5 μs

VII. Conclusion

In this paper, an approach for timing analysis and response time of end to end packet delivery for Ethernet Networked Control System (ENCS) was considered. We considered a collision free network configuration which is more suitable for control application as compared to the traditional network configurations. The office network was separated from the control network and a fully switched Ethernet network is employed for control network. The timing behaviour of a fully switched network has been analysed and the response time of end to end packet delivery in switched Ethernet network has derived.

References

[1] Dimitri Bertsekas, *"Data Networks"*. Prentice Hall, 1992.

[2] IEEE Computer Society, *"IEEE standard 802.3"*, 2002.

[3] IEEE Computer Society, *"IEEE standard 802.3ae"*, 2002.

[4] Anders Martinson, "Scheduling of Real-Time Traffic in a Switched Ethernet Network", Master Thesis, Department of Automatic Control, Lund Institute of Technology, March 2002.

[5] N-Torn (The Industrial Network Company), "Why Use Industrial Ethernet Switches?" http://www.n-tron.com/pdf/why_use_industrial .pdf, 2003.

[6] Bill Moss, "Real-time Control on Ethernet", *Dedicated Systems Magazine*, pp 53-60, 2000.

[7] Andrew S. Tannenbaum, *“Computer Networks”*, 4th Ed, Prentice Hall, 2002.

[8] J.F. Ren, R. Landry, “Flow Control and Congestion Avoidance in Switched Ethernet LANs”, *IEEE International Conference on Communications*, Vol. 1, pages: 508-512, 1997.

[9] J. Wang and B. Ravindran, “Time-Utility Function-Driven Switched Ethernet: Packet Scheduling Algorithm, Implementation, and Feasibility Analysis”. *IEEE Transactions on Parallel and Distributed Systems*, Vol. 15, No. 2, Feb. 2004.

[10] Y. Song, A. Koubaa and F. Simonot, “Switched Ethernet for Real-Time Industrial Communication: Modelling and Message Buffering Delay Evaluation”, *4th IEEE International Workshop on Factory Communication Systems*, Aug. 2002.

[11] J Jasperneit, P Neumann, “Switched Ethernet for Factory Communication”. *Proceeding of 8th IEEE International Conference on Emerging Technologies and Factory Automation*, 2001.

[12] Paula Doyle, “Introduction to Real-Time Ethernet 1”. *Contemporary Control Systems*, Vol. 5, Issue 3, 2004.

[13] Paula Doyle, “Introduction to Real-Time Ethernet 2”. *Contemporary Control Systems*, Vol. 5, Issue 4, 2004.

[14] B. Y. Choi, S. Song, N. Birch and J. Huang, “Probabilistic Approach to Switched Ethernet for Real-Time Control Applications”. *Proceedings of the 8th International Conference on Real-Time Computing Systems and Applications*, Dec. 2000.

[15] J. Jasperneit, P. Neumann, M. Theis, K. Watson, “Deterministic Real-Time Communication with Switched Ethernet”, *4th IEEE International Workshop on Factory Communication Systems*, Aug. 2002.

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

[17] J. P. Georges, T. Divoux and E. Rondeau, “Comparison of Switched Ethernet Architectures Models”, *Proceeding of 9th IEEE International Conference on Emerging Technologies and Factory Automation*, pages: 357-382, 2003.

[18] N. Krommenacker, J. P. Georges, E. Rondeau, and T. Divoux, “Designing, Modelling and Evaluating Switched Ethernet Networks in Factory Communication Systems”, *1st International Workshop on Real-Time LANS in the Internet Age*, Jun. 2002.

[19] A. Yiming, and T. Eisaka, “A Switched Ethernet Protocol for Hard Real-Time Embedded System Applications”, *Proceedings of the 19th IEEE International Conference on Advanced Information Networking and Applications (AINA’05)*, 2005.

[20] J. P. Georges, T. Divoux, and E. Rondeau, “A formal method to guarantee a deterministic behavior of switched Ethernet networks for time-critical applications”, *IEEE International Symposium on Computer Aided Control System Design*, Sep. 2004.

[21] J. P. Georges, T. Divoux, and E. Rondeau, “Strict Priority versus Weighted Fair Queueing in Switched Ethernet networks for time critical applications”, *Proceedings of the 19th IEEE International Parallel and Distributed Processing Symposium (IPDPS’05)*, 2005.