

ANALYSIS OF DETERMINISTIC END-TO-END DELAY IN MULTI-HOP AFDX AVIONICS NETWORK SYSTEM *

Xiaoqiang Ji*, Huanzhong Li†, Jian Li*, Hairui Zhou*, Fei Hu*, Xue Liu‡ and Guchuan Zhu§

*School of Software, Shanghai Jiao Tong University, Shanghai, China

†School of Computer Science, McGill University, Montreal, Canada

‡Department of Computer Science & Engineering, University of Nebraska Lincoln, Lincoln, U.S.A.

§Department of Electrical Engineering, École Polytechnique Montréal, Montreal, Canada

Keywords: AFDX, Network calculus, Arrival curve, Service curve, Pay burst only once, Delay bound.

Abstract: Avionics Full Duplex Switched Ethernet (AFDX) is a deterministic communication protocol for real-time applications on Ethernet media. It is a promising technique that can improve the interconnection of electronic devices in an aircraft. One of the key challenges in employing AFDX is to determine the transmission delay in such a network. This paper aims at handling this challenge by theoretical analysis. A network calculus-based approach is presented for analyzing the end-to-end transmission delay of virtual links in an AFDX network that may consist of many network nodes with different scheduling disciplines. We further improve the approach by taking the specific effects of virtual links into account. In addition, we conduct a simulation to verify the validity of our approach. Simulation results show that switches with different scheduling disciplines can improve the delay performance in a multi-hop network.

1 INTRODUCTION

Avionics Full Duplex Switched Ethernet (AFDX) is an avionics data bus system emerging in recent years. AFDX standard provides a detailed description of the electrical requirements and protocol specification (ARINC, 2005). With bandwidth up to 100Mbps, AFDX network is a thousand times faster than its predecessor - ARINC429. With a continuously increasing number of a variety of devices and amount of data traffic, ARINC429 has gradually failed to meet the requirements of modern avionics systems. Thanks to its low cost, high speed, good scalability and other technical advantages, AFDX will become one of the future mainstream technologies to replace the existing avionics data buses, such as ARINC429, MIL-STD-1553B and so on.

In the actual deployment, the important issue is the calculation of the end-to-end transmission delay in an AFDX network. Ensuring a bounded transmission delay is essential to a hard real-time system such as the avionics system. One of the key challenges in assessing deterministic real-time guarantees and

bounded delay in data transmission is how to calculate the end-to-end delay effectively and accurately.

There are many methods for calculating the network transmission delay, for example, network simulation, network calculus, model checking technique, etc (Charara et al., 2006). Nowadays, due to the advantage of applicability and rigour, network calculus theory is often used for analyzing the real-time performance of communication networks, especially in the worst-case transmission delay analysis.

In previous works, the scenario that a number of virtual links pass through several switches with different scheduling algorithms is not involved, so the method for computing the end-to-end delays is not given to handle this scenario effectively. In this paper, we propose a novel framework to analyze the end-to-end transmission delay in a multi-hop AFDX network using network calculus. Based on the service curve model and delay analysis theorems of network calculus, an upper bound on the end-to-end delay is computed. Our network calculus-based approach can analyze most virtual links in an AFDX network that consists of different network nodes with common scheduling disciplines.

The main contributions of our research are sum-

*This work was supported by NSFC 61003011 and MOST 2010DFA11390.

marized as follows. We present models of AFDX nodes (End System and AFDX Switch), and calculate service curves of various scheduling disciplines that can be used in AFDX nodes. Using the “pay burst only once” phenomenon and service guarantee theorem of network calculus, we acquire an analytical upper bound on the end-to-end delay of a virtual link by computing the maximum horizontal distance between service curve and arrival curve. Note that we do not use the sum of all node delays as the end-to-end delay of a virtual link, because of the repetitive delay calculations in a multi-hop network. Taking the effects of virtual links that share a common physical link into account, we further improve the delay bound by optimizing the delay calculation in the FCFS scheduling. We conduct a simulation platform to verify the validity of our approach. Simulation results also show that switches with different scheduling algorithms can improve the delay performance in a multi-hop network.

The rest of the paper is organized as follows. Section 2 elaborates the AFDX system model, including traffic model, End System (ES) model and Switch model. Section 3 studies rate-latency service curve provided by multiplexer (MUX) with multiple scheduling algorithms. Section 4 is the calculation and analysis of the end-to-end delay, and gives an optimization in FCFS scheduling calculation. In Section 5, we use a simulation platform to verify our results. The conclusion is drawn in section 6.

2 AFDX SYSTEM MODEL

AFDX network is made up of End Systems and Switches. ES receives a variety of data from avionic equipments and sends to a series of AFDX switches, which quickly forward data to the appropriate destination ESs. For the purpose of using arrival curve α and service curve β of Network Calculus (Le Boudec and Thiran, 2001; Chang, 2000) to obtain the end-to-end delay, the remaining section will build the model for divers components in AFDX networks.

2.1 Traffic Model

In AFDX network, virtual link (VL) mechanism (GE, 2007; techSAT, 2008) is utilized to establish logical communication links, which define logical unidirectional connections from a source ES to multiple deterministic terminal ESs.

Each virtual link has a maximum bandwidth determined by two parameters: Bandwidth Allocation Gap (BAG) and the maximum frame length (L_{\max}). BAG is the minimum transmission time interval of adjacent

data frames on the virtual link, whose value ranges in powers of 2 from 1 to 128ms. L_{\max} is the maximum length of data frame allowed to be transmitted on the virtual link, whose range is 64 ~ 1518 bytes.

Bandwidth of a virtual link is then given by

$$(L_{\max} \times 8 \times 1000) \div BAG \text{ bps.}$$

2.2 End System Model

The primary function of ES is to provide safe and reliable avionics data exchange services, whose model is shown in Figure 1.

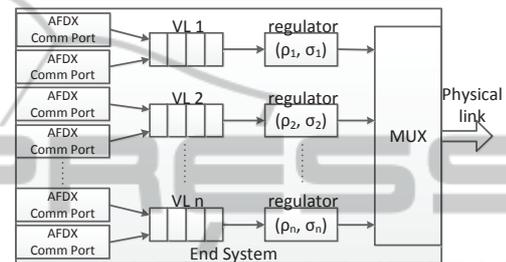


Figure 1: End System Model.

Data coming from avionics devices are received by communication ports and then are carried by virtual link queues.

Based on BAG and L_{\max} of a virtual link, a (σ, ρ) -traffic regulator (Cruz, 1991a; Loeser and Haertig, 2004) is used to pace adjacent frames of the virtual link queue, making them be transmitted at intervals that are not less than BAG (GE, 2007). The output flow is then constrained by the arrival curve that is express as an affine function $\alpha(t) = \rho \times t + \sigma$, where ρ is the sustainable rate of flow, and σ is the outburst.

Multiplexer (MUX) (Cruz, 1991a; Cruz, 1991b) is responsible for multiplexing the regulator outputs into a physical link. MUX could employ a variety of policies to schedule virtual links, such as FCFS, SP, etc. Note that the service curve of a virtual link depends on the scheduling policy. Service curve depicts the forwarding capability of network node, and is represented by the rate-latency function $\beta(t) = R \times (t - T)$, where R is the service rate and T is the latency experienced in a network node. It provides a simple and effective method to study various services offered by network nodes in the worst cases.

2.3 Switch Model

An AFDX Switch receives and forwards virtual link data, whose model is shown in Figure 2 (Charara et al., 2006). Each port of switch is connected to at most one ES. It is mainly constituted by packetizer, forwarding processor and MUX.

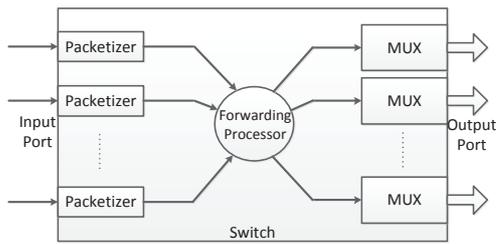


Figure 2: Switch Model.

Switch receives data bits from input ports and puts them into packetizers. When a data frame is completely received by a packetizer, it will be immediately moved to the MUX of appropriate output port by forwarding processor based on static forwarding table. This operation costs a technological forwarding latency that is considered to be fixed here. MUX could also use a number of scheduling policies to output arrival data just like those used in ES.

3 RATE-LATENCY SERVICE CURVE IN MUX

In ES and switch, virtual link data ultimately enter MUX and then are scheduled to output. This section will give different service curves offered by MUX corresponding to different scheduling algorithms.

A MUX can be seen as a combination of FCFS queues and a scheduler. After entering the MUX, data firstly go to FCFS queues to be buffered, and then the scheduler selects and outputs buffered data in queues according to the scheduling algorithm. On a typical industrial configuration, physical links of AFDX network are lightly loaded. Most of the links are loaded at less than 15% and no link is loaded at more than 21% (Ridouard et al., 2008). The sustainable rate ρ of the arrival VL is no more than the reserved rate allocated to the virtual link, i.e. data backlog in queues will not be unlimited growth to cause data overflow.

3.1 FCFS MUX

For a FCFS MUX with output rate C , only one FCFS queue is used. Serving one virtual link, MUX outputs data with service curve $\beta(t) = C \times t$, i.e. as long as a data frame is completely received by FCFS queue, it is immediately scheduled to output by scheduler.

When $VL1, VL2$ with arrival curves α_1, α_2 are multiplexed in FCFS MUX, each virtual link meets (σ_i, ρ_i) -restriction ($i = 1, 2$). From a property of FCFS (Corollary 6.2.3 in (Le Boudec and Thiran, 2001)), service curve β_1 provided to α_1 by FCFS MUX can

be derived and given by $\beta_1(t) = (C - \rho_2) \times (t - \frac{\sigma_2}{C})$.

In a general case where multiple virtual links with arrival curves $\alpha_1, \alpha_2, \dots, \alpha_i, \dots, \alpha_n$ are multiplexed in FCFS MUX, each arrival virtual link meets (σ_i, ρ_i) -restriction ($i = 1, 2, \dots, n$). To study the service curve β_s provided to a given VL S by MUX, we can divide input virtual links into two parts. The first one is a signal virtual link described by α_s ; The other one is the aggregate flow of the rest input virtual links, and can be expressed as $\alpha'(t) = \rho' \times t + \sigma'$, where $\rho' = \sum_i \rho_i - \rho_s$ and $\sigma' = \sum_i \sigma_i - \sigma_s$. According to the second case, we have the following theorem.

Theorem 3.1 When n virtual links satisfying $\alpha_1 \sim (\sigma_1, \rho_1), \dots, \alpha_s \sim (\sigma_s, \rho_s), \dots, \alpha_n \sim (\sigma_n, \rho_n)$ pass through FCFS MUX with output rate C , the service curve β_s provided to VL S by FCFS MUX is given by

$$\beta_{s,MUX}(t) = (C - \rho') \times (t - \frac{\sigma'}{C}) \quad (1)$$

The output curve satisfies $\alpha_s^* \sim (\sigma_s^*, \rho_s)$, and outburst is $\sigma_s^* = \rho_s \times \frac{\sigma'}{C} + \sigma_s$, where $\rho' = \sum_i \rho_i - \rho_s$, and $\sigma' = \sum_i \sigma_i - \sigma_s$.

3.2 Static Priority MUX

For a static priority (SP) MUX, each FCFS queue receives particular virtual links corresponding to a priority level. For example, queue i receives i -priority VLS. A bigger priority number i means a higher priority level. The scheduler always chooses data in the highest priority queue to output with a non-preemptive manner.

Assuming that all arrival virtual links can be divided into n subsets, each one corresponds to a priority level i ($i=1, 2, \dots, n$) and is received by an appropriate priority queue among $\{Q_1, Q_2, \dots, Q_n\}$. Arrival curve of every VL is represented by α_{ij} , where i is the priority level and j is an index. For the queue Q_k with k -priority, arrival curve of input VLS is $\alpha_k(t) = \rho_k \times t + \sigma_k$, where $\rho_k = \sum_{i=k} \rho_{ij}$ and $\sigma_k = \sum_{i=k} \sigma_{ij}$.

From the property of the SP scheduling (Proposition 1.3.4 in (Le Boudec and Thiran, 2001)), we know that if a server with output rate C serves two flows H and L , then the high priority flow α_H is guaranteed by service curve with rate C and latency $\frac{L_{max}^L}{C}$, where L_{max}^L is the maximum frame size for the low priority flow. The low priority flow α_L is guaranteed by service curve with rate $C - \rho_H$ and latency $\frac{\sigma_H}{C - \rho_H}$. We can extend this result to the case of multiple flows and get the following theorem.

Theorems 3.2 When n data flows satisfying $\alpha_1 \sim (\sigma_1, \rho_1), \dots, \alpha_k \sim (\sigma_k, \rho_k), \dots, \alpha_n \sim (\sigma_n, \rho_n)$ pass through SP MUX with output rate C , and the priority of α_i is i (i is bigger while its priority level is higher,

and $1 \leq i \leq n$), the k -priority flow is guaranteed by a rate-latency service curve

$$\beta_k(t) = C_k \times (t - T_k) \quad (2)$$

where $C_k = C - \sum_{i>k} \rho_i$, and $T_k = \frac{\sum_{i>k} \sigma_i + \max_{i \leq k} L_{ij}}{C_k}$.

The formula (2) can be considered that the scheduler assigns output rate C_k to the k -priority queue Q_k , and data of Q_k will wait latency T_k to be scheduled. We could call T_k the scheduling preparation time.

If k -priority queue Q_k only receives data of one VL, the result of (2) will be the service curve of this VL. If Q_k receives multiple VLs and VL S conforms to (σ_{ks}, ρ_{ks}) -constraint, we would focus on the service curve provided to a VL S among multiple virtual links that enter queue Q_k . According to previous analysis of the FCFS scheduling, there has

$$\begin{aligned} & \beta_{s,MUX}(t) \\ &= (C_k - \rho_k') \times (t - T_k - \frac{\sigma_k'}{C_k}) \\ &= (C - \sum_{i \geq k} \rho_{ij} + \rho_{ks}) \times (t - \frac{\sum_{i \geq k} \sigma_{ij} - \sigma_{ks} + \max_{i \leq k} L_{ij}}{C - \sum_{i > k} \rho_{ij}}) \end{aligned}$$

The output curve satisfies $\alpha_{ks}^* \sim (\sigma_{ks}^*, \rho_{ks})$, and $\sigma_{ks}^* = \rho_{ks} \times (T_k + \frac{\sigma_k'}{C_k}) + \sigma_{ks}$, where $\rho_k' = \sum_{i=k} \rho_{ij} - \rho_{ks}$ and $\sigma_k' = \sum_{i=k} \sigma_{ij} - \sigma_{ks}$.

4 THE END-TO-END DELAY CALCULATION

We defines the end-to-end delay as the maximum interval that VL data experienced from the time emitted from the traffic regulator in the source ES to that received by the destination ES. As the propagation latency on physical links is usually small compared to that on ES and switch, it's ignored in the following.

To compute the end-to-end delay using the "pay burst only once" phenomenon, we first need to know service curves provided by the source ES and every switch that VL S passes and then convolute all service curves to get the entire one provided by AFDX network. The end-to-end delay can be computed based on the maximum horizontal distance between the arrival curve and the service curve, and expressed by

$$h(\alpha, \beta) = \sup\{\inf\{T : T \geq 0; \alpha(s) \leq \beta(s + T)\}\}.$$

"Pay burst only once" is an important phenomenon in Network Calculus. It states that when a data flow passes through two network nodes in sequence, the sum of the maximum delay on each node is greater than the end-to-end delay. For example, if a flow with arrival curve α passes through two network nodes S_1, S_2 with service curves β_1, β_2 , and $\beta_1 \otimes \beta_2$ is the overall service curve supplied by S_1, S_2 , we get

$$h(\alpha, \beta_1) + h(\alpha, \beta_2) \geq h(\alpha, \beta_1 \otimes \beta_2),$$

where α^* is the arrival curve of the flow output from S_1 . This phenomenon shows that the delay caused by the burst of the input flow is calculated twice in the former approach (Cen et al., 2008). For the case that a flow through multiple nodes, this phenomenon also exists, which is used to improve the end-to-end delay.

4.1 The End-to-End Service Curve

From the ES model presented in Section 2, it can be seen that the virtual link delay on the ES is mainly generated in MUX.

In a switch, the main components are packetizer, forward processor, and MUX. The maximum delay on packetizer is $\frac{L_{\max}}{C}$ and the service curve provided to VL S can be expressed as $\delta(t - \frac{L_{\max}}{C})$, where C is the maximum transmission rate of physical links connected to each switch. Forward processor with a fixed forwarding latency T_f can be seen as a fixed delay line (Cruz, 1991a) whose service curve is $\delta(t - T_f)$. The entire service curve offered by a switch can be expressed as

$$\begin{aligned} \beta_s &= \delta(t - \frac{L_{\max}}{C}) \otimes \delta(t - T_f) \otimes \beta_{s,MUX} \\ &= C_s \times (t - \frac{L_{\max}}{C} - T_f - T_{ms}) \end{aligned}$$

We assume that VL S passes n switches, and make a convolution over all the service curves provided by the source ES and switches. Finally the end-to-end service curve β_{e-e} offered to VL S is given by

$$\beta_{e-e} = C_e \times (t - T_e) = \beta_{s0} \otimes \beta_{s1} \otimes \beta_{s2} \otimes \dots \otimes \beta_{sn} \quad (3)$$

where β_{si} is the service curve provided by the source ES ($i = 0$) or switch i ($i > 0$).

When VL S shares its queue k with other virtual links in node i , we not only take the scheduling delay into account, but also add the latency caused by FCFS service in FCFS queue. Thus, $T_{msi} = T_{ki} + \frac{\sigma_{si}'}{C_{si}}$, and

$$\beta_{si} = \begin{cases} (C_{si} - \rho_{si}') \times (t - T_{ki} - \frac{\sigma_{si}'}{C_{si}}), & i = 0 \\ (C_{si} - \rho_{si}') \times (t - T_f - \frac{L_{\max}}{C} - T_{ki} - \frac{\sigma_{si}'}{C_{si}}), & i > 0 \end{cases}$$

We put β_{si} into (3) and obtain the service curve of the entire AFDX network offered to VL S :

$$\beta_{e-e} = \min_{0 \leq i \leq n} (C_{si} - \rho_{si}') \times [t - n \times T_f - n \times \frac{L_{\max}}{C} - \sum_{0 \leq i \leq n} (T_{ki} + \frac{\sigma_{si}'}{C_{si}})] \quad (4)$$

4.2 The End-to-End Delay Calculation

By calculating the maximum horizontal distance between β_{e-e} and arrival curve α_s of VL S , the end-to-end delay we acquire is

$$\begin{aligned} Delay_{e-e} &= \frac{\sigma_s}{C_e} + T_e \\ &= \frac{\sigma_s}{\min_{0 \leq i \leq n} (C_{si} - \rho_{si}')} + n \times T_f + n \times \frac{L_{\max}}{C} \\ &\quad + \sum_{0 \leq i \leq n} (T_{ki} + \frac{\sigma_{si}'}{C_{si}}) \end{aligned} \quad (5)$$

To sum up, the calculation of the end-to-end delay of virtual link requires specific parameters about the network conflagration, such as forwarding table of switch, scheduling policy, transmission bandwidth, property of virtual link, L_{max} , BAG and so on. These would allow network nodes to supply different scheduling services, making the design of the end-to-end delay very flexible.

4.3 The Optimization in FCFS Scheduling Calculation

In the research on the end-to-end delay in AFDX networks, we found that in many cases, frames of some virtual links are transmitted on the same physical link and output from the same port of next connected node. Since frames are transmitted in serial on a physical link and received in the order of arrival at connected ES/Switch, this phenomenon has not been taken into account in the FCFS scheduling calculation of (5), and leads to a counteractive delay estimation. In order to optimize the delay calculation and to obtain better results, the effect from sharing one physical link by multiple VLs should be eliminated in the FCFS scheduling calculation.

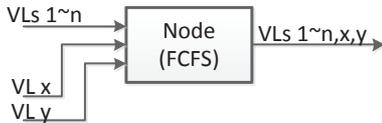


Figure 3: Example of the optimization.

We assume that $VLs1 \sim n$ share a physical link in Figure 3, and meet (σ_i, ρ_i) -constraint ($1 \leq i \leq n$). They are transmitted to a Node with FCFS scheduling, and then output from a port with VL x and y . Because of sharing a common link, the aggregate flow z of $VLs1 \sim n$ is not characterized by $\alpha_z(t) = \sum_{1 \leq i \leq n} \rho_i \times t + \sum_{1 \leq i \leq n} \sigma_i$, but by $\alpha_z(t) = \sum_{1 \leq i \leq n} \rho_i \times t + L_{max}$, where L_{max} is the largest frame size of $VLs1 \sim n$ (Bauer et al., 2009). The delay of flow z is just influenced by flow x and y , and so are $VLs1 \sim n$. Service curves of $VLs1 \sim n$ are also the same with flow z . So when obtaining the service curve of one of $VLs1 \sim n$, we just need to calculate the bursts of VL x and y and ignore burst influences of $VLs1 \sim n$, thus decreasing $\frac{\sigma_{si}}{C_{si}}$ in (5). For VL x and y , they are not influenced by bursts of $VLs1 \sim n$, but only by L_{max} . Obviously, taking into account actual virtual link path distribution, would further optimize the computation of the end-to-end delay.

5 EVALUATION AND ANALYSIS

In order to verify the validity of the end-to-end delay calculation, simulation experiments are carried out to measure the end-to-end delay of different virtual links. The simulation results regarding different scheduling strategies are compared with the calculated delay using the network calculus in this section.

5.1 Simulation Scenario

The experiments are carried on a small AFDX network depicted in Figure 4. It is composed of 3 interconnected switches and 7 ESs. In these simulation experiments, 7 VLs are studied with different scheduling strategies, and the end-to-end delay of each virtual link is obtained. Taking $VL1$ for example, its source ES is $ES1$, destination ES is $ES6$, and the path is $ES1 - S1 - S3 - ES6$.

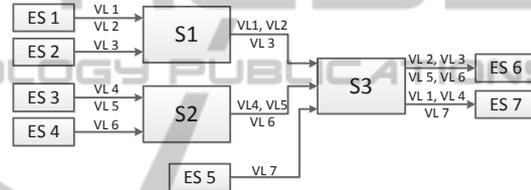


Figure 4: Small AFDX architecture and VL paths.

The latency of forwarding processor is assumed to be a fix value $16\mu s$ in switches (Charara et al., 2006). According to restrictions on BAG and L_{max} in AFDX network, the specific configuration of $VL1 \sim VL7$ is shown in Table 1. Each VL has a priority number: the bigger the number, the higher the priority.

Table 1: Parameters of VLs.

VL NO.	BAG	L_{max}	priority
VL1, VL4	$4000\mu s$	120 bytes	3
VL2, VL5	$16000\mu s$	320 bytes	2
VL3, VL6	$32000\mu s$	600 bytes	1
VL7	$2000\mu s$	80 bytes	4

5.2 Simulation with TrueTime Simulator

The networked control system simulation software - TrueTime - is used as the experiment platform. It is a Matlab/Simulink-based simulator for real-time control systems (<http://www.control.lth.se/truetime/>). TrueTime Toolbox includes two main interface modules - TrueTime Kernel and TrueTime Network. The Kernel module is composed of flexible real-time kernel, A/D and D/A converters, network interface and

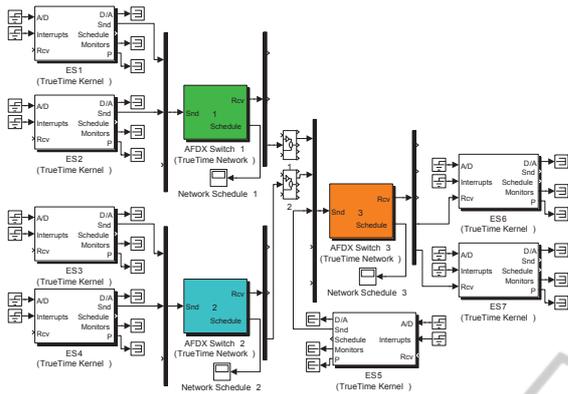


Figure 5: TrueTime simulation platform.

external access. The network module provides various network blocks, including CSMA/CD(Ethernet), CSMA/AMP(CAN), Switched Ethernet, etc.

this simulation platform is built as an AFDX network with transmission rate 100Mbps. Switched Ethernet block is used as the basis for AFDX network (switch function has been implemented in Switched Ethernet block), and TrueTime Kernel is used as the ES. Every VL is achieved with a periodic task. FCFS is the default scheduling method in the switch. By modifying the source code, we add SP and other scheduling methods to the switch. Figure 5 illustrates the implementation of simulation platform in TrueTime. Each VL meets its requirements of *BAG* and the lengths of VL frames comply with the evenly distribution between 64 bytes to L_{max} .

5.3 Computed End-to-End Delay Bounds

The end-to-end delay of a virtual link can be computed by a direct sum of the delay of all the nodes that the virtual link traverses or by the application of “bay burst only once”. Figure 6 shows the end-to-end delay bounds for VL2 with different scheduling combinations in S1 and S3 using these two methods respectively. In the first method, because of the repetitive burst calculations in a multi-hop network, it makes the calculated delay larger. But in the second method, the concatenation theorem is used to eliminate the effect from the repetitive calculations, leading to a better result than that of the first method.

5.4 End-to-End Delay Analysis on Different Scheduling Combinations

In simulation studies, FCFS scheduling policy is adopted in ESs and various scheduling strategies in

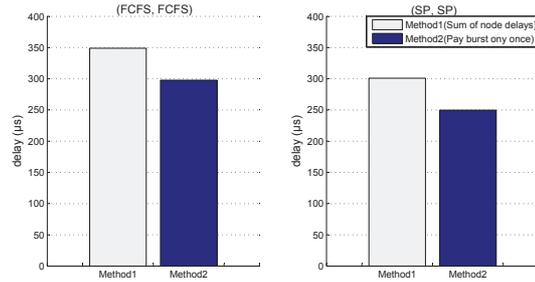


Figure 6: Analysis of the end-to-end delay with two calculation methods.

switches. Since the priorities of the virtual links are not the same, we observed that when different combinations of FCFS and SP are used in Switch1, Switch2 and Switch3, virtual links’ delay through the network are different. Scheduling combination is represented as (S1, S2, S3)-configuration. After several simulations, the maximum delay values of virtual links are obtained. The measured values and theoretical values without optimization are shown in Figure 7.

In (FCFS, FCFS, FCFS)-configuration, theoretical values of virtual links with different priority are not well distinguished. While SP scheduling being added in the switch, the end-to-end delay corresponding to each virtual link varies. In the (FCFS, FCFS, SP) and (SP, SP, FCFS) configurations, each one only uses SP scheduling once on virtual links, but the delays of the same virtual links are different. In the (SP, SP, SP) network, SP scheduling are applied twice in the network, leading to a better distinguishment of virtual links with three priority levels. Since network

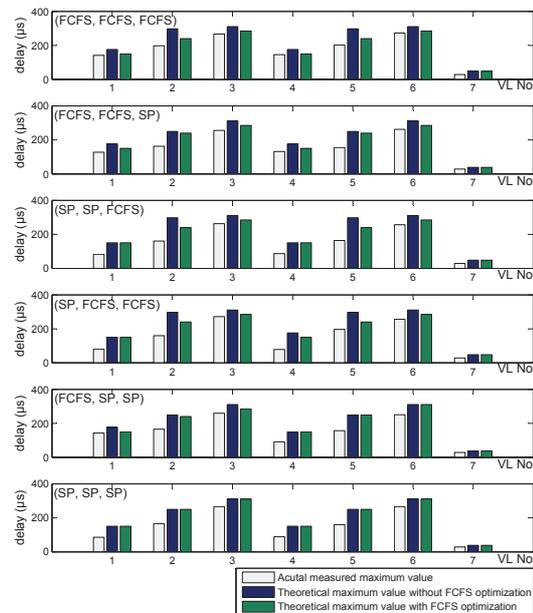


Figure 7: Each VL’s delay with different scheduling combinations.

structure of Switch1 and Switch2 is symmetrical, situations of (FCFS, SP, FCFS) and (SP, FCFS, SP) are similar to that of (SP, FCFS, FCFS) and (FCFS, SP, SP). Thus they are ignored in Figure 7.

The difference between actual measured values and theoretical ones is mainly due to the fact that theoretical values are results of the worst case delay analysis which is obviously pessimistic. In network calculus, the worst case scenario is considered on each node visited by each VL and the maximal possible latency of competition is taken into account. This approach always gives guaranteed upper bounds on the end-to-end delay that usually can never happen and leads to impossible scenarios. For example, virtual link data are not always to be sent with the maximal frame length. Although theoretical values are certainly larger than actual measured values, they could well reflect the overall trend of the delay variation.

When taking into account FCFS optimization, theoretical delay bounds with optimization are better than that without optimization and closer to actual values (shown in Figure 7). In (SP, SP, FCFS) and (SP, FCFS, FCFS) configurations, the delay of VL1 has not been significantly improved, because VL1 shares an output port of S3 with VL4 and VL7, but doesn't share the physical link from S1 to S3. Owing to the fact that FCFS isn't applied in (SP, SP, SP)-configuration, no optimization is obtained in this case.

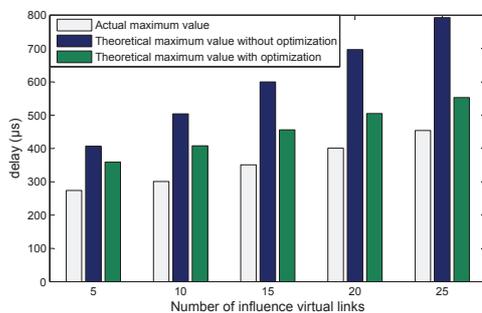


Figure 8: VL3's end-to-end delay with influence links.

In order to better show the effect of FCFS optimization, more influence virtual links are introduced in the physical link that VL3 passes through from S1 to S3, sharing an output port of S3 with VL3. The variations of VL3's end-to-end delay with the increasing number of influence links are displayed in the Figure 8 in (FCFS, FCFS, FCFS)-configuration.

From Figure 8, we notice that simulation values change a little ($274\mu s \sim 454\mu s$) and theoretical values without optimization vary greatly ($407\mu s \sim 793\mu s$). This is mainly due to the fact that virtual links share the same physical link and data are transmitted in serial. With optimization, theoretical delay varies in $359\mu s \sim 553\mu s$, which has been significantly reduced.

6 CONCLUSIONS

This paper presents a network calculus-based approach for the end-to-end delay analysis in multi-hop AFDX networks. Using the service curve model to describe the transmitting service, we obtain an analytical upper bound on the end-to-end delay of VL. In order to derive the overall service curve offered by the whole network, we model various AFDX network nodes and study diverse scheduling disciplines. This approach can analyze most VLs in an AFDX network that consists of different nodes with common scheduling disciplines. Additionally, a simulation platform is conducted to verify the validity of our approach.

REFERENCES

- ARINC (2005). Arinc 664: Aircraft data network. part 7, avionics full duplex switched ethernet (afdx) network. Annapolis, Maryland, USA.
- Bauer, H., Scharbarg, J.-L., and Fraboul, C. (2009). Applying and optimizing trajectory approach for performance evaluation of afdx avionics network. In *Proceedings of the 14th International Conference on Emerging Technologies and Factory Automation*, Mallorca, Spain. IEEE.
- Cen, F., Xing, T., and Wu, K.-T. (2008). Real-time performance evaluation of line topology switched ethernet. *International Journal of Automation and Computing*, 05(4):376–380.
- Chang, C. (2000). *Performance Guarantees in communication networks*. Springer, London.
- Charara, H., Scharbarg, J.-L., Ermant, J., and Fraboul, C. (2006). Methods for bounding end-to-end delays on an afdx network. In *Proceedings of the 18th Euromicro Conference on Real-Time Systems*.
- Cruz, R. (1991a). A calculus for network delay, part i: Network elements in isolation. *IEEE Transactions on information theory*, 37(1):114–131.
- Cruz, R. (1991b). A calculus for network delay, part ii: Network analysis. *IEEE Transactions on information theory*, 37(1):132–141.
- GE (2007). Afdx/arinc 664 protocol tutorial. Technical report.
- Le Boudec, J.-Y. and Thiran, P. (2001). *Network Calculus, A Theory of Deterministic Queuing Systems for the Internet*. Springer, Berlin.
- Loeser, J. and Haertig, H. (2004). Low latency hard real-time communication over switched ethernet. In *Proceedings of Euromicro Conference on Real-Time Systems*.
- Ridouard, F., Scharbarg, J.-L., and Fraboul, C. (2008). Stochastic upper bounds for heterogeneous flows using a static priority queueing on an afdx network. *Dans Journées FAC'2008, SVF/FRIA, 2008*.
- techSAT (2008). Afdx/arinc 664 tutorial. Technical report.