New Feasible Grouping Algorithms for Virtual Link in AFDX Networks

YoungJun Cha, JuHo Lim, SunYoung Lee, DongJin Kim and Ki-Il Kim
Department of Informatics
Gyeongsang National University
Jinju 660-701, Korea
kikim@gnu.ac.kr

***Abstract*—As the Avionics Full-Duplex Switched Ethernet network has been employed for airborne data networks, most of related research work has mainly focused on the analysis of delay and jitter between the end systems with predetermined configuration. However, due to assumption on preconfigured virtual links, their optimal solutions seem unsuitable for implementation. This implies that configuration problem for the virtual link remains unsolved yet. To meet this requirement, in this paper, we propose new algorithms for configuring the virtual link by determining grouping of each flow while reducing the required bandwidth in AFDX. This algorithm is accomplished by taking two different approaches, bottom-up and top-down. Finally, we prove their suitability for configuration problem through experimental results by demonstrating not much difference between two algorithms in the aspect of required bandwidth.**

Keywords— AFDX; Payload; Grouping;

I. INTRODUCTION

The recent avionics system exchanges data among various digital devices mounted on the aircraft via Aircraft Data Network (ADN)[1-3]. Prior to existing ADN, diverse technologies such as ARINC 429, MIL-STD-1553 and ARINC 629 have been used to support communications in airborne networks. But, since they cannot meet the requirements of modern aircraft due to inherent problems, for example, low transmission rate, limited amount of data and high cost, the network has gradually evolved corresponding to the evolution of the digital devices. Recently, Avionics Full-Duplex Switched Ethernet (AFDX) has been introduced.

AFDX is based on IEEE 802.3 and standardized as a PART 7 in ARINC 664[4]. In the research area of AFDX, most of research work has been mainly conducted to determining the modeling of AFDX network and analysis of delay, jitter between end systems through simulations and mathematical modeling. For example, J. Javier Gutiérrez et al., have proposed a mechanism for response time analysis to study AFDX network scheduling and contention between end systems in [5]. Also, this paper addressed analysis of worst-case end-to-end latency. Other researchers have paid attention to the cause of the time in [6]. In this literature, they have showed impact of First Come First Service (FCFS) and traffic shaping in the aspect of delay reduction of network through token bucket as well as queuing theory on multiple virtual links.

It is also noticeable for measuring method with end-to-end delay and evaluated performance in AFDX network in [8]. Different from previous analysis method for AFDX, the authors in [9] modeled the AFDX network which consists of number of end systems, switch and virtual link on NS-2 platform also analyzed the end-to-end delay, jitter of AFDX network through simulations for performance evaluations. In parallel with previous works, the scheduling algorithm in AFDX switch has been studied in several literatures[10-11]. They employ Earliest Deadline First (EDF)[10], static priority scheduling algorithm[11] and analyze their impact for real-time communications.

According to previous research efforts, we can recognize that virtual link is very important configurable parameter to meet AFDX requirements. Consequently, each virtual link should meet the traffic scheduling, bandwidth and maximum jitter constraints. Since these parameters are guaranteed by virtual link, how to configure the virtual link becomes an important problem prior to analyzing the delay in AFDX network. Moreover, in the point of bandwidth, it is required to reduce bandwidth to meet all requirements. To solve this problem, frame packing strategies and a Hierarchical Traffic Shaping (HTS) algorithm to control downstream (actuators) flows was studied in [12]. However, their solutions target to optimize the bandwidth in preconfigured networks where all virtual links are already preconfigured.

Thus, configuration problem for virtual link is demanded to be explored. But, despite its importance, as far as the authors know, only one study has been conducted with regard to configuration in the paper by our research group [13]. Main contribution of the paper is to address the AFDX network configuration, especially for considering virtual link properties such as Bandwidth Allocation Gap (BAG) and Maximum Transfer Unit (MTU) in timely manner. The proposed algorithm derives all possible pairs of BAG and MTU satisfying scheduling constraints in each virtual link and then obtains feasible set of BAG and MTU satisfying other two constraints, that is, bandwidth and jitter. We also proposed algorithm for deriving a set that uses the least bandwidth in AFDX network.

However, in our previous paper, we have derived the feasible value of BAG and MTU only if the virtual link has already been configured. This means that it does not mention how to configure the actual virtual link. So, this configuration problem was addressed in our previous works in [14]. We provided bottom-up approach and experiment results. But, since the proposed scheme is too complex to get the result, simplified bottom-up approach is proposed here through dividing one virtual link into two sub ones depending on average value. Also, it is hard to identify their suitability without comparing it with other merging algorithms. So, in this paper, we extend our previous work by revising top-down approach and introducing bottom-up algorithm. Two algorithms are compared through experimental results in the aspect of required bandwidth for several cases to prove their suitability as a solution.

The rest of this paper is organized as follows. We introduce the research background and brief related work in section I. In the section II, the problem statement and new proposed algorithms are given. And then, in the section III, we provide the analysis of performance evaluation. Finally, conclusions and further work will be presented in section IV.

II. Proposed Algorithms

A. System Model and Problem Definition

Since we deal with real-time AFDX messages, a message flow f_i is defined by (l_i, p_i), where l_i is the payload of the message in bytes and p_i is Message Transmit Cycle (MTC) of the message in msec. That is, a message of l_i bytes is generated every p_i time units and is delivered to the destination application through AFDX networks. A virtual link VL_i is defined by (BAG_i, MTU_i, F_i) as follows.

- BAG_i: bandwidth allocation gap or period of VL_i in a value of 2^k msec where $k = 0,1,\ldots,7$
- MTU_i: maximum transfer unit or message size of VL_i in bytes
- F_i: a set of flow id in VL_i, where the message flow with flow id j is denoted as $f_j = (l_j,p_j)$

For a given virtual link VL_i, MTU and BAG are configured so as to meet all the real-time requirements of message flows in the link. Since all BAGs of VLs are harmonic, the schedulability, bandwidth, and jitter analysis is easily derived by utilization analysis and defined as mandatory requirements in [4]. They are represented in equation (1) – (3). 强制性的

$$\sum_{j\in F_i}^{n_i} \frac{\left\lceil l_j / MTU_i \right\rceil}{p_j} \le \frac{1}{BAG_i} \qquad (1)$$

$$8\sum_{i=1}^{n} \frac{MTU_i + 67}{BAG_i} \times 10^3 \le B \qquad (2)$$

$$40 + \frac{8\sum_{i=1}^{n}(67 + MTU_i)}{B} \le 500 \qquad (3)$$

Based on these requirement, we address the configuration problem to reduce the required bandwidth when there are n flows represented by $\xi = \{f_1, f_2, \ldots, f_n\}$ between end system and switch. Our solution is to construct F_i in VL_i while reducing the required bandwidth as compared to the case that any specific algorithm is not given. To achieve this goal, we propose two heuristic algorithms depending on grouping metrics such as required bandwidth or flow property. By the proposed algorithms, two or more flows are grouped in one VL only if the required bandwidth for VL being merged flows is less than sum of bandwidth of merging flows.

In order to solve this problem, equation (2) which determines the required bandwidth for each virtual link needs deep analysis. To reduce the bandwidth of VL_i, the BAG of the 分母 denominator value increases and the MTU of the numerator value decreases, as shown in equation 2. So it can be known that by grouping the payload to be the BAG of smaller value and the MTU of greater value for VL_i can minimize the used bandwidth in AFDX network. The detail procedures are described in Algorithm 1. To configure the virtual link based on these basic analysis, this paper employs two approaches to minimizing the value of each virtual link and calculating the bandwidth with the values to reduce the total used bandwidth.

1) Simplified Average Based Top-Down Approach

Due to reduce the computing complexity in our previous algorithm, in this approach, we divide one virtual link into two sub ones. In Algorithm 1, one virtual link is created with all flows at the initial stage as shown in below step 3. And then, this large virtual link can be sequentially split into two small virtual links when the required bandwidth of split virtual link is less than one large virtual link. This means that split condition is dependent on how much bandwidth is required. The pseudo code for top-down approach is shown in algorithm 1.

Algorithm 1: Simplified Average Based Top-Down Approach

1: $VL_i = i^{th}$ virtual link
2: B_i = Required Bandwidth of VL_i
3: $VL_1 \leftarrow F_1 \cup F_2 \cup \ldots F_n$
4: $Max \leftarrow 1$

5: **for** VL_i **from** 1 **to** Max **do**
6: Avg = average(B_i)
7: **for** each flow j in VL_i
8: **If** $B_j <$ Avg
9: $TVL_1 \leftarrow TVL_1 \cup \{F_j\}$
10: **Else**
11: $TVL_2 \leftarrow TVL_2 \cup \{F_j\}$
12: **end for**
13: **If** ($B_{TVL1} + B_{TVL2} < B_i$) {
14: $VL_{i+1} \leftarrow TVL_1$ 这个地方绝对有问题，VL可能被覆盖
15: $VL_{i+2} \leftarrow TVL_2$ 掉，用队列就好了
16: $Max \leftarrow i+2$
17: $TVL_1 \leftarrow \{\ \}$
18: $TVL_2 \leftarrow \{\ \}$
19: }
20: **end for**

In algorithm 1, two temporary groups are created with the required bandwidth. One group is for bandwidth less than average of current *VL* and the other is for bandwidth larger than one. If two groups are suitable for splitting, one *VL* is divided into two *VLs* with flows which are classified by bandwidth. Otherwise, current *VL* is maintained. And then, current *VL* increases and temporary set becomes empty for next operation. By increasing *Max* variable in Algorithm 1, it iteratively performs split procedure until there is no more virtual link to be divided.

2) Bottom-up Approach

Unlike top-down approach, another possible approach for configuration of virtual link is merging. In this approach, at the initial stage, each virtual link is occupied by one only flow. And then, each flow is merged with others to reduce the required bandwidth. This procedure continues until no more available virtual link remains. The pseudo code is shown in Algorithm 2.

Algorithm 2: Bottom-up Approach

$VL_i = i^{th}$ virtual link
B_i = Required Bandwidth of VL_i
for i **from** 1 **to** n **do**
  $VL_i \leftarrow F_i$
  Compute B_i
end for

for i **from** 1 **to** n **do** 如果第一个就没有merge就停了吗？为什么遍历终止条件是n？
  $k \leftarrow$ Find flow id with minimum B_i
  $j \leftarrow$ Find flow id with maximum B_i
  If ($B_k + B_j < B_{k \cup j}$) { 这个位置是不是应该是>号
    $VL_k \leftarrow F_k \cup F_j$
  }
end for

In Algorithm 2, each virtual link consists of each respective flow. And then, required bandwidth is computed for each flow by our previous work. Upon finalizing this procedure, we choose the flow with minimum required bandwidth and maximum required bandwidth. And then, the merge condition is comparison of required bandwidth as the same as top-down approach. If the merged bandwidth is less than current required bandwidth of two flows, two flows are merged. If the no available flow is found, virtual link is maintained. This procedure will be performed as many as number of flows because one merging is done at each round. In order to reduce the number of round for merging, it is possible to accomplish several merging procedures in a round. In this case, more merged virtual links are created but their granularity is coarse.
间隔尺寸

III. Performance Evaluation

In this section, we provide the experiment results for proposed scheme through diverse flows sets. In the experiments, we assume eight message flows for simplicity. The payload of a message is randomly generated from 20 to 80 bytes. The MTC or period of a message is randomly selected among five different intervals. Five intervals range from 10 to 260 msec and the duration of each interval is 50 msec. The network bandwidth is set to 6Mbps. In order to model eight flows, three different data sets are tested as shown in Table 1. Each set has different rule for payload and MTC where Fixed represents the same value for parameter and Varied does varying one depending on experiment setting explained above. Through these three sets, we can identify the impact of each parameter and relationship between them. For the comparison, two performance parameters, amount of required bandwidth and number of configured virtual link, are evaluated for the mentioned three experiment sets.

Table 1. Data sets for experiments

	Payload	MTC
Data Set 1	Varied	Fixed
Data Set 2	Fixed	Varied
Data Set 3	Varied	Varied

Figure 1 illustrates the configured bandwidth for three data sets. In this figure, ONEVL represents a case for algorithm 1 which indicates only one VL is considered. On the other hand, NVL represents the case where virtual links as many as n is created. For the all cases, ONEVL requires the largest bandwidth. Similarly, it shows larger bandwidth than NVL because merging with different procedure requires additional bandwidth. This result implies that good configuration algorithm is essential.

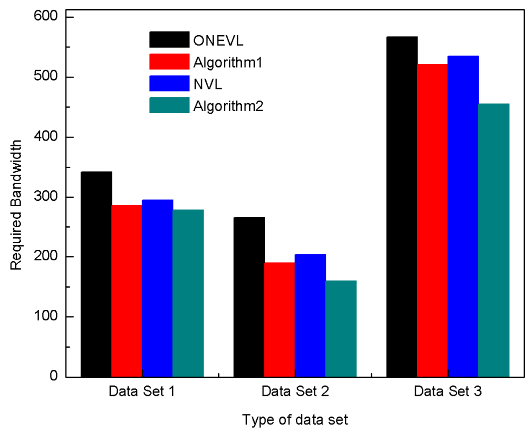

Figure 1. Comparison of required bandwidth

When it comes to compare two algorithms, they reveal better performance than both ONEVL and NVL. Moreover, bottom-up approach shows better performance than top-down approach. This is because that more cases for bandwidth comparison are available in bottom-up cases. However, in top-down approach, dividing two groups are not effective and efficient for grouping. So, there are many cases to finish split procedure at the early time. Also, with regard to data set, data set 2 shows the biggest difference in required bandwidth. This means that MTC is more sensitive to compute the required bandwidth than payload. This is because BAG is mostly affected by MTC. As mentioned before, merging procedure is more likely to find possible pairs rather than split one. In the split procedure, if two groups have the slight difference, split procedure creates suitable two sub-groups. However, there are many cases to require more sub-groups than two. In this case, current algorithms fail to find out the suitable sub-groups.

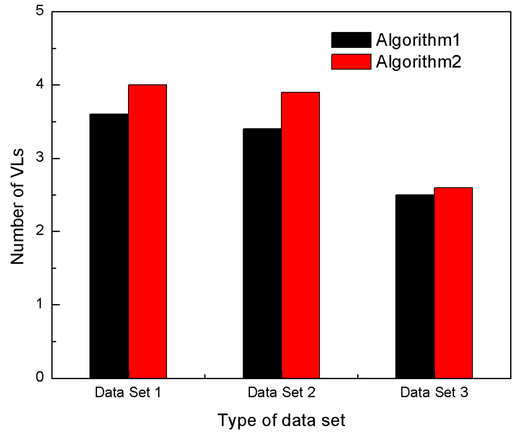

Figure 2. Comparison of configured virtual links

IV. Conclusion

In this paper, unlike previous research work, we proposed the algorithms of virtual link configuration to reduce the required bandwidth in AFDX network. We evaluated the results through simulations and analyzed the proposed algorithms. In the simulation results, the two algorithms show the possibility to reduce the required bandwidth in AFDX network and bottom-up algorithm reveals better performance than top-down one.

Related to this research, we will additionally conduct research how to configure the virtual link considering the property of application such as MTC to reduce the required bandwidth in AFDX network and evaluate the performance with various scenarios through simulations. Also, grouping method for each algorithm needs to be elaborated toward optimization by considering number of groups for splitting or grouping order in merging.

Acknowledgment

This work was supported by the BK21 Plus Program (Research Team for Software Platform on Unmanned Aerial Vehicle, 21A20131600012) funded by the Ministry of Education (MOE, Korea).

References

[1] AFDX/ARINC 664 Protocol Tutorial by General Electric

[2] AFDX / ARINC 664 Tutorial – TechSAT. http://www.techsat.com/fileadmin/media/pdf/infokiosk/TechSAT_TUT-AFDX-EN.pdf.

[3] AFDX / ARINC 664 Tutorial – CEMS, http://www.cems.uwe.ac.uk/~ngunton/afdx_detailed.pdf.

[4] ARINC Aircraft Data Network Part 7 Avionics Full Duplex Switched Ethernet Network, ARINC Specification 664 P7, Sep 2009.

[5] J. Javier Gutiérrez, J. Carlos Palencia, Michael González Harbour, "Holistic schedulability analysis for multipacket messages in AFDX networks," Journal of Real-Time Systems, vol. 50, no. 2, pp. 230-269, Mar. 2014.

[6] S. Wang, J. Shi, D. Sun, M. Tomovic, "Time delay oriented reliability analysis of Avionics Full Duplex Switched Ethernet," In Proc. of IEEE Industrial Electronics and Applications (ICIEA), pp. 982 - 987, June. 2013.

[7] H. Charara, J. Scharbarg, J. Ermont, and C. Fraboul, "Methods for bounding end-to-end delays on an AFDX network," In Proc. Of 18th Euromicro Conference on Real-time Systems, pp. 193-202, July 2006.

[8] H. Bauer, J. Scharbarg, and C. Fraboul, "Worst-case end-to-end delay analysis of an avionics AFDX network," In Proc. of Design, Automation & Test in Europe Conference & Exhibition, pp. 1220-1224, March 2010.

[9] D. Lina, S. Dong, Z. Xingxing, H. Qiong, "The Research of AFDX System Simulation Model," In Proc. of Multimedia Technology (ICMT), pp. 1 - 4, Oct. 2010.

[10] J. Zhang, S. Qiao, D. Li, G. Shi, "Modeling and simulation of EDF scheduling algorithm on AFDX switch," IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC), Sep. 2011.

[11] R.T. Flores, R.T. M. Boyer, "Performance analysis of the Disrupted Static Priority scheduling for AFDX," In Proc. of ACM/IEEE International Conference on Formal Methods and Models for Codesign (MEMOCODE), Oct. 2014.

[12] H. Ayed, H. A. Mifdaoui, C. Fraboul, "Hierarchical traffic shaping and frame packing to reduce bandwidth utilization in the AFDX," In Proc. of IEEE International Symposium on Industrial Embedded Systems (SIES), Jun. 2014.

[13] D. H. An, H.W. Jeon, K. H. Kim, K. I. Kim, "A feasible configuration of AFDX networks for real-time flows in avionics systems," in Proc. of International Workshop on Real-Time and Distributed Computing in Emerging Applications (REACTION-13), Dec. 2013.

[14] Y. J. Cha, K. I. Kim, "Heuristic Algorithm for Virtual Link Configuration in AFDX Networks," in Proc. of International Workshop on Real-Time and Distributed Computing in Emerging Applications (REACTION-14), Dec. 2014.