Hybrid Scheduling Technology of Time-triggered Ethernet Switches: A Review

Xiaofang Ma and Askar Hamdulla*
Xinjiang University, Institute of Information Science and Engineering, Urumqi, 830046, China
*Corresponding Author Email: askar@xju.edu.cn

Abstract. Switches are one of the key devices in the time-triggered Ethernet. It is very necessary to design high-performance switches to achieve high real-time data transmission. This article sorts out relevant documents on converters, and first introduces the development of avionics system networks. Then, based on Crossbar's switch structure and related scheduling algorithms, compare the performance of different algorithms. Compare the performance of different algorithms. Secondly, it introduces the current research methods of TTE switches. Finally, it summarizes the future development direction of time-triggered switching hybrid scheduling technology.

1. Introduction
With the rapid development of avionics system technology, Distributed Integrated Modular Architecture (DIMA) is the development direction of future avionics systems [1]. DIMA has the characteristics of distributed and mixed key security, and has high requirements for network throughput, speed, delay, real-time and deterministic data transmission, security and reliability [2].

The traditional avionics system represented by 1553B bus network has many disadvantages in the face of excessive amount of data and equipment, such as low transmission rate, dependence on the bus controller, and if the terminal fails, the whole bus system will be completely paralyzed [30].

Professor Kopetz of Vienna University of Technology proposed the mechanism of time trigger in 1993 [3]. The time trigger mechanism is to forward the data according to the prescribed network communication task list through the global synchronous clock. Then Professor Kopetz design idea of Time-triggered Ethernet [4]. Time-triggered Ethernet is a new real-time communication protocol that is fully compatible with Ethernet. It has the characteristics of high certainty, reliability, real-time and strong scalability [5].

Studies have shown that the advantages of time-triggered power are very suitable for the development of future avionics systems, so it is of great significance to study the key device switches for time-triggered communication. This article first briefly describes the importance of studying time-triggered Ethernet switches, and then introduces the Crossbar structure type with buffer and the corresponding scheduling algorithm, followed by examples of current implementation methods for TTE switches, and finally discuss the problems that TTE switches need to solve.

2. The Switch Cache Structure
Depending on the location of the cache in the switching structure, crossbar-based the switching structure is divided into Output-Queued (OQ), Input-Queued (IQ), Input-Output Queued (CIOQ), and Combined Input-Crosspoint Queued (CICQ). The Crossbar-based structure as presents in figure 1.
The OQ switching structure is to set up a buffer at each output port. At the input port, the data is directly forwarded to the corresponding output port when it arrives, as shown in figure 2. When multiple data arrive at the input port at the same time, all the data needs to be forwarded to the output end. At this time, there must be a higher rate inside the switch to ensure that data is not lost, so OQ structure is not suitable for high-speed networks [6].

The IQ switching structure places the buffer in the input port and sets a FIFO for each input. Exchange structure indicated in Fig. 3. But for the use of a single FIFO structure, there will be a HoL problem [7]. If the output port to which the head-of-line data is sent in the FIFO queue is occupied by other input ports at the same time, even if the destination port of the next data in the queue is free, data cannot be sent. As shown in figure 3, output port 1 has been occupied by input port 2, and the first data of the FIFO queue in input port 1 cannot be transmitted, causing other data in the queue to be unable to be transmitted. If the problem of head blocking occurs, it will greatly reduce the throughput of the switch. In order to solve the HoL problem, a VOQ is proposed [8], the figure 4 is an input queue structure with a VOQ structure.

The CIOQ structure combines the advantages of IQ and OQ. A FIFO buffer is placed at each input and output port. For co-occupation of simultaneous output ports, the data is buffered first. This method can reduce the transmission delay caused by the header segmentation problem and increase the data transmission rate [9]. Another CIOQ structure uses VOQ on the output side, and still uses a single FIFO queue to buffer data on the output side, as show in figure 5 [10]. CIOQ must take into account the centralized scheduling of input and output. Algorithm complexity is high, not suitable for hardware implementation.
The first several structures only consider input and output, and the intersection is only the state of cross and bar. CIQ is to set up a buffer at the cross point, generally using FIFO, and using VOQ at the input to eliminate HoL problems, as show in figure 6 [11]. Scheduling algorithms based on cross-point buffering are all used in switches based on the CIQ structure. Since the buffer queue is added at the cross-point to separate the bandwidth conflicts between the input and output ends, the input and output scheduling algorithms can be independent, so only the cross-point is considered. Data interaction between cache queues.

3. Scheduling Algorithm
The transmission of data from the input end to the output end requires a scheduling algorithm to solve the problem of data resource occupation. And a good scheduling algorithm will affect the throughput, delay and other performance of the switch. Scheduling algorithms based on crossbar switches can be divided into parallel iteration, weight, polling and intersections caching.

3.1. Scheduling Algorithm Based on Parallel Iteration
Parallel iteration scheduling algorithm is mainly to solve the input port to the output port data transmission matching as far as possible, to avoid data has been waiting for the state. An iterative process consists of three steps: Request, Grant, Accept. The Fig. 7 shows the three steps.

![Figure 7. The three steps of an iteration](image)

The PIM algorithm uses randomness in the grant and acceptance process [12]. The output port randomly selects an input port to send the request. Because of the random way, it is difficult to implement on the hardware. iSLIP is a de-randomization algorithm [13], change the random method to the priority method, which solves the shortcomings of the PIM algorithm. Presents a QPS-r algorithm solve the complexity of algorithm, in the request process, using some input ports VOQ, where the ratio of the length of each FIFO to the total length of VOQ determines the standard to send a request to the output port, and in the grant process, authorized to the input port with the maximum voq length [14]. Hybrid i-SLIP Algorithm [15], the algorithm will be divided into different equipment master core and slave core. In the master core pointer to a fixed priority level, and in the Slave core uses the i-SLIP algorithm, to reduce delay and increase the rate of the system.

3.2. Scheduling Algorithm Based on Weight
MWM calculates the maximum total weight on the input and output matching in a different way, which is the input port on which the data is to be sent and the destination port on which it is to be sent [16]. LQF uses the length in the queue as the method to calculate the weight [17], but for some queues whose length is always short, there will be a state of waiting. In order to solve the problem of long waiting time, OCF using the time of waiting in the queue as the weight is proposed [18]. There is also LPF that using the sum of output possession and input possession as weights [19]. Input possession is the total amount of data currently waiting on the input port to be sent to each output port, and output possession is the total amount of data waiting on all input ports to be sent to one output port. Although the idea of using the heavy method is simple, the algorithm complexity is too high to be realized in hardware. Therefore, in order to reduce the algorithm complexity, the idea of iteration is added to the algorithm, such as i-LQF [16], i-OCF [17] and i-LPF [19] algorithms.
3.3. Scheduling Algorithm Based on Polling Mechanism
For unfair scheduling algorithms, the polling mechanism is a fair choice for all queues. In the simplest sequential polling Round-Robin algorithm (RR), the output of each queue is undifferentiated, but the uneven distribution of traffic results in some queues with long waits [20]. The EDRRM algorithm [21], an improved algorithm based on DRRM, does not modify the pointer until the queue is empty. Improvement in the case of non-uniform flow can also achieve good effect, SRRR algorithm [20], before the request stage increased step, each output sends a signal to its highest priority input. This makes know that the output of the corresponding input end, input can choose to send the request in the request process, on the output side whether pointer is to update the receipt of a request, this way to reduce the delay.

3.4. Scheduling Algorithm Based on Cross-point Queue Status
As the name suggests, the method based on cross-point caching mainly considers the strategy of information interaction between the corresponding cross-point and input and output respectively. MCBF algorithm [22], the SBF method is used in input scheduling, and the LBF is used in output scheduling. Every time input data is to be cached, the corresponding remaining space is always the largest among all the caches, and the one with the smallest remaining cache space is the first to output data, freeing up space and ensuring load balance within a certain length. This algorithm reduces the complexity of the algorithm and can reduce the complexity of the hardware implementation of the switching structure with cross-point cache. But now that the data traffic increases, the performance of MCBF will decrease, and the intersection itself needs a certain amount of cache space. An algorithm that requires the state of the input queue, SCBF algorithm [23]. This algorithm mainly selects the output port with the smallest occupancy rate for the buffer queue corresponding to the input terminal, so that each output terminal keeps working as much as possible. Sometimes can achieve 100% throughput. But the algorithm needs to understand the status of each input and output terminal.

4. TTE Switch
In time-triggered Ethernet, data communication between terminal systems requires TTE switches. Handling different types of data frames requires different design requirements from traditional switches.

4.1. TTE Network Data Flow
TTE network integrates the event trigger mechanism in traditional Ethernet and adds the time trigger mechanism to improve the real-time and deterministic performance of the network. Event trigger mechanism: the transmission message can be allocated according to the demand, and can be adjusted at any time, with good flexibility. However, in this way, link occupancy conflicts will occur, which is unpredictable and cannot be isolated in advance in case of network node failure, with poor reliability. Time trigger mechanism: All kinds of messages are transmitted through the specified schedule, without conflicts on the data link, the transmission time and delay of messages are predictable, with high reliability, but this will lose flexibility.

There are two different kinds of traffic in TTE network. The traffic based on event trigger mechanism, namely the traditional Ethernet data flow (ET frame), the TT frame and PCF frame based on time trigger mechanism [24]. And ET frames are divided into RC and BE frames. PCF frames have the highest priority, followed by TT frames, then RC frames and finally BE frames.

4.2. Design of TTE Switch
In time-triggered Ethernet, data communication between terminal systems requires TTE switches. In addition, TTE switches process TT messages and ET messages of two types of data traffic and are also compatible with traditional Ethernet standards. Therefore, TTE switches should have the following functions [25]: (1) classify ET frames and TT frames; (2) Support TT message pre-emption to ET message; (3) Retransmit the interrupted ET messages; (4) Transmit ET frames according to Ethernet standards.
Pengfei Gao used a two-layer structure to divide the data processing of TT frame and ET frame into two layers [26]. For the TT layer, it was based on the storage of Crossbar structure, and dual-port RAM was used to store scheduling table entries to control the TT frames arranged for offline scheduling. In the ET layer, the CIOQ structure of single FIFO cache queue is used, and the ET frame use the polling scheduling strategy at the output end. In the end, although the function of the switch is realized, the performance of the switch is not tested and analysed.

Facing the complexity of data transmission in spacecraft, Wu Shaokang proposed an adaptive QoS priority scheduling algorithm [27]. The TT frame achieves real-time effects in a pre-emptive manner. RC data frames use token buckets to control forwarding, and discard and forward according to the length of the data frame and the number of tokens. The BE frame uses the remaining bandwidth for transmission.

Realize TTE switch based on CIOQ switch structure, Lu Wei [28] added the improvement of time synchronization to the function of the switch. The TT frame is specified according to the off-line scheduling table, the isolated clock domain is added, and the TT frame is cached in a special FIFO. The ET frames are forwarded by polling, and the output side uses RAM to store the ET frames. Finally, FPGA was used to realize the function of the switch, and the performance of the switch was tested. The transmission rate was about 936Mbit/s, the dither was about 0.047ms, and the packet loss rate was about 0.14%, with good transmission speed and low delay.

In order to make different data frames change without interference, reduce transmission delay. Zhang Jinguo divided the internal network of the switch into three exchange planes, namely PCF frame, TT frame and ET frame, according to the different data frames to be processed [29]. Such a design makes the transmission of the three frames do not affect each other and reduces the transmission delay of each frame to a certain extent. And the scheduling of time-triggered frame is improved. TT frame, by scheduling table will be sent to the input, use a simple exchange of the interconnection structure will forward it to the output side. Finally proves the feasibility of the switch design, but no further analysis the performance of the switch.

At present, the research on TTE switches mainly realizes the functions of the switches, and all add time-synchronized data frames to ensure that the time reference error of data interaction is within a small range. There are also TTE designs that combine the mature technology of traditional Ethernet switches. The structure of the switch basically uses a polling-based scheduling method for the transmission of ET data frames, which reflects the fairness of data transmission. However, there is no detailed analysis on the packet loss rate, data delay and jitter of the TTE switch.

5. Conclusions
The technology of time-triggered Ethernet is difficult to explore a new technology due to its late start time and immature technology. Time-triggered Ethernet has added time-triggered data frames. This requires TTE switches to have a more complicated data transmission process than traditional switches. Time synchronization data frames are added to synchronize the global time reference, so it requires the switch must have high accuracy. No matter from what point of view, the time-triggered switch has a more complex design to meet the current development of avionics systems.

Now the basic method is to use FPGA to verify and realize the basic data transmission function of TTE switch. There are many aspects that affect the performance of the switch, but the main thing is the data transmission method and the switch buffer structure. In the face of different types of data streams, their scheduling methods have something in common.

TT frames are all based on the offline schedule, which is transmitted at a specified time, and pre-emptive methods are used in the switch. For the ET frame, it is also stored in the FIFO queue first, and then forwarded by polling, which achieves a higher transmission rate and low delay. However, in the actual data forwarding process, burst data frames will appear. If there is a problem that the burst TT data frame happens to be sent to the same output port as the TT frame that has been transmitted at the specified time, there is no solution.

Although the traditional Ethernet switch scheduling method can be used for ET frames, the addition of TT frames increases the difficulty of hardware implementation of the switch, so the forwarding of ET frames should use scheduling algorithms with low algorithm complexity and good
A simple structure will affect the transmission rate. If you choose a complex structure, the hardware implementation is more difficult, so it is a difficult problem to balance the two.

6. Acknowledgments
This research is supported by the National Natural Science Foundation for Young Scholars of China (61701276). The authors are very much grateful to the Professor Zhu Jihong’s research team of Tsinghua University for providing the experimental environments and their insightful comments.

7. References
[1] Zhou X, He F and Xiong H G 2019 Comput. Eng. Des. 40 1
[2] Hao Y K, Zhang X H, Cui X and Huang B L 2016 Comput. Inf. Tel. Syst (China: Beijing) pp 1-8
[3] Kopetz H, Ademaj A, Grillinger P and Steinhammer K 2005 Object-Oriented Real-Time Distributed Computing (Washington State: Seattle) pp 22-33,
[4] Steinbach T, Korf F and Schmidt T C 2011 IEEE International Conference on Consumer Electronics -Berlin (Germany: Berlin) pp 216-220
[5] Hou M H and Chen C H 1997 IEEE Computer Society (Taiwan: Taipei) pp 244-250
[6] Liao J and Wu P 2010 ISITA (Taiwan: Taichung) pp 332-336
[7] Kyriakos A, Patronas I, Tzimas G, Kitsakis V and Reisis D 2017 PACET (Greece: Xanthi) pp 1-4
[8] Chen J and Stern T 1991 IEEE J. Sel. Areas Commun. 9 439
[9] Attiya H, Hay D and Keslasy I 2010 IEEE Trans. Comput. 59 1378
[10] Jebalia M, Letaïfa A B and Tabbane S 2015 COMNET (Tunisia: Tunis) pp 1-7
[11] Liu K, Yan J, Lu J and Chen X 2016 IEEE Commun. Lett 20 498
[12] Anderson T E, Owicki S S, Saxe J B and Thacker C P 1993 ACM Trans. Comput. Syst. 11 319
[13] Mahobiya D 2014 IJAR 3 2250
[14] Gong L, Xu J, Liu L and Liu L 2019 arXiv preprint arXiv 1005
[15] Mbarek I B, Mazouzi M, Hasnaoui S and K. Jelassi 2015 Web Application and networking. World. Symposium. 2nd (Tunisia: Sousse) pp 1-7.
[16] Wang L K, Ye T, Lee T T and Hu W S 2018 IEEE Trans. Parallel Distrib. Syst. 29 1456
[17] Hu B, Yeung K L, Zhou Q and He C 2016 IEEE ACM T Network 24 3565
[18] Li X K and Elhanany I 2008 IEEE Trans. Commun. 56 21
[19] Radonjic M, Rudusinovic I and Veljovic Z 2009 IEEE EUROCON (Russia: St.-Petersburg) pp 1844-1851
[20] Lin D, Jiang Y and Hamdi M 2011 ICC. (Japan: Kyoto) pp 1-5
[21] Li Y H, Panwar S and Chao H J 2002 IEEE HPSR (Japan: Kobe) pp 58-63
[22] Mhamdi L and Hamdi M 2003 IEEE Commun. Lett 7 451
[23] Zhang X and Bhuyan L N 2004 IEEE Global Telecommun. Conf. (Texas: Dallas) pp 1168-1173
[24] Zhou X, He F, Xiong H G and Li E S 2019 IEEE/AIAA 38th DASC (USA: San Diego) pp 1-8
[25] Steinhammer K, Grillinger P, Ademaj A and Kopetz H 2006 DATE (Germany: Munich) pp 1-6
[26] Gao P F 2014 Xidian Univer
[27] Wu S K 2016 National University of Defense Technology
[28] Lu W, Zhu J H, Yang J L and Yu S L 2019 Computer Measurement & Control 27 264
[29] Zhang J 2020 University of Electronic Science and Technology of China
[30] Peng An 2010 World Congress on Intelligent Control and Automation (China: Jinan) pp 953-956