Low-latency Deterministic Forwarding for Multi-modal Connection Using Time-Sensitive Networking

Weichen Zhang
Global Innovation Exchange
Tsinghua University
Beijing, China
weiczh02@gmail.com

Fan Dang*
Global Innovation Exchange
Tsinghua University
Beijing, China
dangfan@tsinghua.edu.cn

Xu Wang
Global Innovation Exchange
Tsinghua University
Beijing, China
xu_wang@mail.thu.edu.cn

ABSTRACT

This poster proposes a low-latency deterministic forwarding technology for multi-modal data based on time-sensitive networks. It presents the design and implementation of an FPGA-based switch that can handle data frames of different protocols, including Ethernet, PROFINET, and EtherCAT. The switch consists of an input port, an output port, a clock module, an exchange module, and a gate control module. Its core is the switch module, which comprises four sub-modules: protocol identification, caching, handling, and conversion. To prioritize critical traffic over normal traffic, the switch implements a priority-based scheduling algorithm. It also provides timestamping for performance analysis. Through the proposed technology and switch design, the poster aims to improve low-latency deterministic forwarding for time-sensitive multi-modal data.

CCS CONCEPTS

• **Networks** → **Link-layer protocols**; • **Hardware** → **Networking hardware**.

KEYWORDS

data protocols, switch, industrial Internet, time-sensitive networks

ACM Reference Format:
Weichen Zhang, Fan Dang, and Xu Wang. 2023. Low-latency Deterministic Forwarding for Multi-modal Connection Using Time-Sensitive Networking. In *ACM Turing Award Celebration Conference 2023 (ACM TURC '23), July 28–30, 2023, Wuhan, China.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3603165.3607395

1 INTRODUCTION

A new round of industrial revolution with the industrial Internet at its core is booming across the world. The industrial Internet supports emerging industrial applications, such as smart manufacturing with a wide future [6]. However, the current network architecture based on a single mode network creates isolation between Information Technology (IT) systems and industrial Operation Technology (OT) systems, as well as between different OT systems, etc. It has affected the further improvement of man-machine interconnection of industrial system and is difficult to support the further development of industrial Internet in the future.

In the aspect of real-time communication, the existing research mainly focuses on two aspects: network protocol standard and network scheduling algorithm. The development of an industrial real-time communication protocol standard gradually shows the trend of integration with standard EtherNet [5]. Siemens, Schneider, Rockwell, and Omron and other foreign enterprises have put forward their own industrial Ethernet protocols, including PROFINET [2], EtherNet [1] and EtherCAT [3]. In recent years, research has focused on the time-sensitive networking (TSN), which makes it necessary to rely on the time sensitive network multi-mode data low delay deterministic forwarding technology [4].

In this paper, a multi-mode data conversion technology based on FPGA is proposed, which can be transmitted in real time depending on TSN. This technology has an important contribution to realizing seamless connection between different protocols and improving network efficiency and stability. Realizing cross-modal data conversion through FPGA can greatly simplify the number and complexity of network devices, and improve the scalability and reliability of the network.

2 DESIGN AND FRAMEWORK

We propose a low-latency deterministic forwarding technology for multi-modal data based on TSN and design an FPGA-based switch, the framework of which is shown in Figure 1.

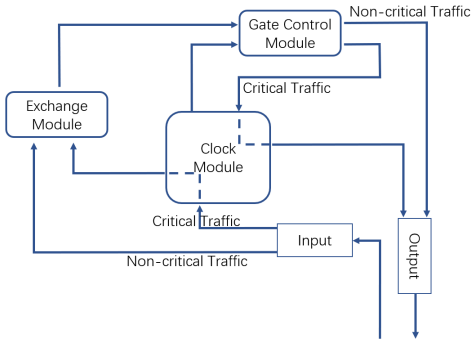

Figure 1: The framework of the switch

*Corresponding Author.
This work is supported in part by the National Key Research Plan under grant No. 2021YFB2900100, and the NSFC under grant 62202263.

ACM TURC '23, July 28–30, 2023, Wuhan, China

ACM ISBN 979-8-4007-0233-4/23/07.
https://doi.org/10.1145/3603165.3607395

The switch is mainly composed of an input port, an output ports, a clock module, an exchange module, and a gate control module. Input data frames should be marked as either critical traffic or non-critical(normal) traffic before entering the switch. Critical traffic passes through the clock module where it is timestamped and enters the exchange module. Normal traffic enters the exchange module directly. The switch module converts data frames of different protocols into data frames of the target protocol (such as Ethernet, PROFINET, etc.) and puts them into the priority queue of the gate control module. The gate control module will transfer the data frames from the corresponding port according to the pre-set priority status and start time table for each flow. When no command is received, the gate control module caches the critical traffic and outputs the normal traffic directly. When the real-time timing of the clock module reaches the preset time point in the start time table, which means that the gate control module has received the command, the gate control module will prioritize the output of critical traffic and temporarily suspend the output of normal traffic. Next, the critical traffic is updated with a timestamp field in the time module and then output, which helps to analyze the performance indicators such as delay jitter of forwarding later. Normal traffic is outputted directly.

The core of the switch design is the exchange module, which converts link layer data frames of Ethernet (IPv4, IPv6), PROFINET (RT, NRT), EtherCAT. The structure of the exchange module is shown in Figure 2. The switch module consists of four sub-modules: protocol identification module, protocol caching module, protocol handling module, and protocol conversion module.

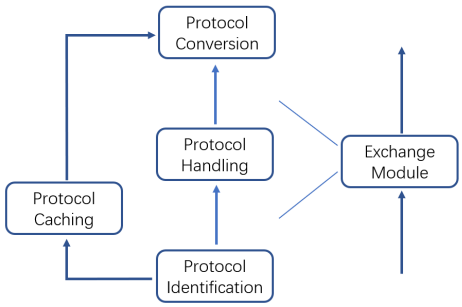

Figure 2: The structure of the exchange module

The protocol identificaiton module firstly creates a state machine for parsing the received data frame. Then, this sub-module uses multiple parallel comparators to compare the EtherType field at the 13th-14th byte of the data frame to complete protocol recognition within one clock cycle. Finally, the sub-module encodes the recognized protocol type as a numerical value for subsequent processing.

The protocol caching module first creates a hash table using BRAM to store the mapping relationship of MAC addresses to protocol types and solves hash conflicts using chaining or open addressing. Then, this module stores the protocol type code output from the protocol identification module in the hash table together with the target and source MAC address of the data frame.

The protocol handling module is designed using pipelining to improve processing speed. Then, based on the output value of the protocol identification module, the data frame is passed to the corresponding protocol handling sub-module. In each protocol handling sub-module, the corresponding header and protocol header are skipped, and the payload is extracted. Data alignment should be particularly noted.

Table 1: The mapping table of protocols and EtherTypes

Protocol	Ethernet -IPV4	Ethernet -IPV6	PROFINET -RT	PROFINET -NRT	EtherCAT
EtherType	0x0800	0x86DD	0x8892	0x8893	0x88A4

The protocol conversion module first extracts the corresponding target MAC address and source MAC address from the hash table of the protocol cache module according to the user-input target protocol value. Then, the protocol conversion module generates the header of the target protocol and pays attention to the correct values of each field. At this step, the target MAC address and the source MAC address should be arranged in order and the corresponding EtherType field should be added at the end to combine into a link layer data frame header. The protocol type corresponds to the EtherType fields as shown in the Table 1.

Next, generate the protocol header of the target protocol. For IP protocol, it is necessary to generate the correct IP packet header, including version, header length, service type, total length, identification, fragment offset, time to live, protocol, header checksum, source address, and destination address fields. Different protocol types should be processed separately by different sub-modules to generate the corresponding protocol header. Finally, the generated frame header, protocol header, and payload are concatenated to generate a new link layer data frame.

3 CONCLUSION

This paper proposes a low-latency deterministic forwarding technology for multi-modal data based on time-sensitive networks and presents the design and implementation of an FPGA-based switch. The switch is capable of handling data frames of different protocols and implements a priority-based scheduling algorithm to prioritize critical traffic over normal traffic. The proposed technology and switch design contribute to the improvement of real-time reliable cross-mode interconnection based on time-sensitive network.

REFERENCES

[1] Paul Brooks. 2001. Ethernet/IP–industrial protocol. In *Proceedings of the IEEE Conference on Emerging Technologies and Factory Automation (EFTA).* IEEE, 397–400.

[2] P. Ferrari, A. Flammini, D. Marioli, and A. Taroni. 2004. Experimental evaluation of PROFINET performance. In *IEEE International Workshop on Factory Communication Systems, 2004. Proceedings.* 331–334. https://doi.org/10.1109/WFCS.2004.1377739

[3] D. Jansen. 2004. Real-time Ethernet: the EtherCAT solution. *Computing and Control Engineering* 15 (February 2004), 16–21(5). Issue 1. https://digital-library.theiet.org/content/journals/10.1049/cce_20040104

[4] John L. Messenger. 2018. Time-Sensitive Networking: An Introduction. *IEEE Communications Standards Magazine* 2, 2 (2018), 29–33. https://doi.org/10.1109/MCOMSTD.2018.1700047

[5] Shiyong Wang, Jian Ouyang, Di Li, and Chengliang Liu. 2017. An Integrated Industrial Ethernet Solution for the Implementation of Smart Factory. *IEEE Access* 5 (2017), 25455–25462. https://doi.org/10.1109/ACCESS.2017.2770180

[6] Hansong Xu, Wei Yu, David Griffith, and Nada Golmie. 2018. A Survey on Industrial Internet of Things: A Cyber-Physical Systems Perspective. *IEEE Access* 6 (2018), 78238–78259. https://doi.org/10.1109/ACCESS.2018.2884906