Design and Implementation of Industrial Gateway for AUTBUS and TSN Protocol Conversion

Zhengqi Rong
Key Laboratory of Industrial Internet of Things and Networked Control, Ministry of Education
Chongqing University of Posts and Telecommunications
Chongqing, China
714773065@qq.com

Min Wei
Key Laboratory of Industrial Internet of Things and Networked Control, Ministry of Education
Chongqing University of Posts and Telecommunications
Chongqing, China
weimin@cqupt.edu.cn

Jiayu Dan
Key Laboratory of Industrial Internet of Things and Networked Control, Ministry of Education
Chongqing University of Posts and Telecommunications
Chongqing, China
1986062569@qq.com

Lei Zhang
Key Laboratory of Industrial Internet of Things and Networked Control, Ministry of Education
Chongqing University of Posts and Telecommunications
Chongqing, China
1742554119@qq.com

***Abstract*—The integration of AUTBUS and Time-Sensitive Networking (TSN) enables real-time, deterministic control data transmission to underlying devices while supporting long-distance, large-scale data transfer. Despite their potential, limited research has been conducted on the interworking between AUTBUS and TSN, hindering the combined deployment of these advanced network technologies in industrial applications. This paper addresses this gap by focusing on seamless interconnection between AUTBUS and TSN. It explores protocol conversion methodologies, examines a heterogeneous network architecture, and details data link layer and comprehensive network frame format conversion techniques, as well as flow identification mapping. Additionally, the paper presents the design of a protocol conversion gateway, covering both its software and hardware architecture. Experimental results demonstrate that the gateway effectively bridges AUTBUS and TSN, ensuring seamless communication and successful protocol conversion between devices adhering to their respective standards.**

Keywords—AUTBUS, Time-Sensitive Networking, Protocol Conversion, Industrial Network Gateway

I. INTRODUCTION

The industrial sector has seen the development of various fieldbus standards, including CAN, Profibus, and Modbus, each addressing specific communication needs. Recently, the AUTBUS standard has become a part of the IEC 61158 sixth edition international standard. AUTBUS introduces advanced technologies such as dual-wire broadband communication, high-speed data transmission, and clock synchronization deterministic communication. This advanced industrial communication protocol is set to revolutionize industrial automation by fulfilling critical requirements such as intelligence, long-distance communication, broadband support, real-time performance, multi-service integration, and unified addressing.

AUTBUS supports a bandwidth of up to 100 Mbps, facilitates both real-time and non-real-time data transmission, and connects up to 254 nodes over distances of up to 1000 meters. Time-Sensitive Networking (TSN), an extension of Ethernet, provides deterministic, low-latency, and highly reliable communication, which is essential for real-time industrial applications. In practical industrial environments, AUTBUS frequently needs to interface with existing network protocols, particularly TSN. The integration of AUTBUS and TSN is a key step to enable two main scenarios: first, the AUTBUS-based system can use the TSN backbone network to achieve seamless interconnection with other TSN-enabled networks; secondly, it facilitates data exchange between AUTBUS and various devices and systems connected through TSN. There are strong industrial demands behind these scenarios, but an urgent challenge is the lack of gateways that can bridge AUTBUS and TSN.

Gateways in the Internet of Things play a decisive role in routing pre-processed and filtered data to the cloud platform [1]. Filho et al. [2] designed a gateway with a classifier that prioritizes and queues messages for efficient delivery using the HTB algorithm. Hemmatpour et al. [3] developed a distributed IoT gateway (DIIG) that uses parallel real-time communication to ensure high-performance data transmission. Chen et al. [4] proposed a multi-conversion system architecture that enhances protocol conversion flexibility by separating data and configuration planes. Sun et al. [5] designed a cloud-based IoT gateway utilizing an intelligent data conversion method, which employs a mapping mechanism to efficiently transfer data between industrial networks. Implemented on a Raspberry Pi, the gateway includes an embedded web server for monitoring and configuration, and supports cloud integration through Representational State Transfer Application Programming Interfaces, allowing seamless integration of devices into the IoT ecosystem with minimal hardware or software modifications. Ray et al. [6] proposed a new data packet structure and data packet generation process at the gateway. Xie and Gao [7] developed a fieldbus protocol conversion gateway system that converts various fieldbus protocols into a unified format for Ethernet communication, demonstrating good real-time performance and stability in experiments. Wang et al. [8] designed a protocol conversion gateway called DoIP, which enables Digital Mobile Radio (DMR) devices to access communication services over the internet. The gateway employs a hierarchical model to map DMR frames to Internet Protocol (IP) packets, ensuring real-time and reliable transmission.

The integration of AUTBUS and TSN presents a robust solution for modern industrial networking needs, combining the strengths of both technologies to advance industrial automation and other applications requiring real-time data transmission.

II. AUTBUS AND TSN NETWORK ARCHITECTURE AND PROTOCOL CONVERSION

A. Heterogeneous network architecture

A heterogeneous network architecture integrating AUTBUS and TSN, which consists of an AUTBUS network, a TSN, and a destination network, is illustrated in Fig. 1. The Centralized User Controller (CUC) collects and processes user requirements concerning data flow, priority, and transmission delays from various network nodes. These requirements are then translated into configuration parameters for the Centralized Network Controller (CNC), which optimizes and manages the network according to these parameters. The CNC handles resource allocation, scheduling strategies, and bandwidth control to ensure that data flows meet the predefined requirements. Within this framework, data collected by AUTBUS terminal nodes is converted to TSN format through the AUTBUS-TSN gateway, integrated into the TSN, and subsequently forwarded by the TSN switch to the target network or destination node, which could be another AUTBUS network or a TSN node.

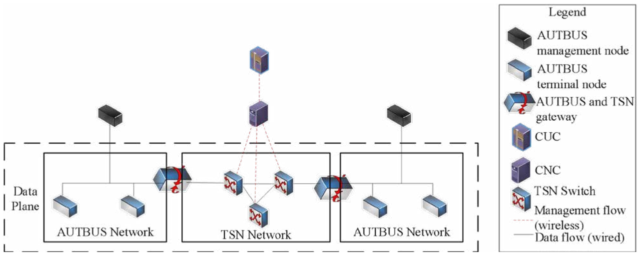


Fig. 1. AUTBUS and TSN heterogeneous network architecture

B. Overall Structure of Protocol Conversion Methodology

The overall structure of the AUTBUS to TSN conversion methodology is illustrated in Fig. 2. It primarily consists of the AUTBUS communication module, the protocol conversion module, and the TSN communication module. The AUTBUS communication module connects to the protocol conversion module via a serial port, enabling modularity and scalability, while the TSN communication module is integrated within the main control chip to enhance real-time performance and system efficiency.

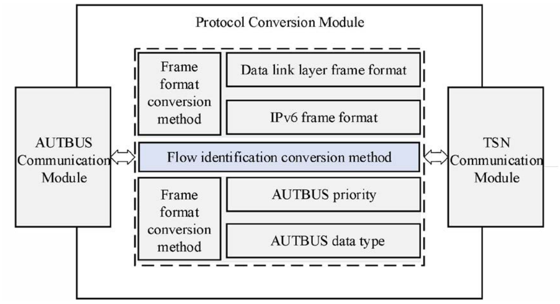


Fig. 2. AUTBUS and TSN transform the overall structure

The AUTBUS communication module performs several functions. It receives bus data from AUTBUS network nodes, encapsulates this data using the Serial Line Internet Protocol (SLIP), and forwards it to the protocol conversion module via the serial port. Additionally, it transmits TSN data from the protocol conversion module to the appropriate nodes based on the destination Network Addressable Object Identifier (NAOID).

The protocol conversion module is central to the conversion process between AUTBUS and TSN protocols. It addresses differences in protocol structure, frame formats, data priorities, and addressing schemes through four main functions: frame format conversion, priority mapping, flow identification mapping, and address mapping.

Frame format conversion is accomplished using two approaches. The first approach involves converting the AUTBUS frame structure into the TSN data link layer frame structure when TSN employs data link layer addressing. The second approach involves converting the AUTBUS frame structure into the TSN frame structure, which encompasses data link, network, and transport layers, when TSN utilizes IPv6 and UDP protocols.

Priority mapping ensures that data flow priorities remain consistent across both networks, taking into account the different industrial applications of AUTBUS and TSN. Address mapping involves translating NAOID addresses into MAC addresses or IPv6 addresses to facilitate cross-network communication.

The TSN communication module ensures compliance with TSN protocols for AUTBUS data flows and handles the reception and forwarding of TSN data to the protocol conversion module, maintaining continuous data flow across the network.

III. AUTBUS AND TSN PROTOCOL CONVERSION METHOD

A. Frame Format Conversion between AUTBUS and TSN

To achieve effective cross-network data transmission between AUTBUS and TSN, it is essential to address the substantial differences in protocol structures and frame formats. Two frame format conversion methods are proposed to facilitate seamless data exchange between AUTBUS and TSN. These methods are designed to enable the selection of the most appropriate conversion strategy according to different network conditions, thereby improving the overall efficiency of cross-network data transmission.

1) Data Link Layer Frame Format Conversion

To facilitate the conversion of AUTBUS data to the TSN data link layer, it is essential to extract key fields from the AUTBUS data frame, including data transmission mode, data type, NAOID address, and flow identification. These fields are then mapped to corresponding TSN fields: data type and transmission mode are mapped to the PCP field in the TSN VLAN tag, NAOID address is mapped to the TSN MAC address, and AUTBUS flow identification is mapped to TSN flow identification. To ensure consistent transmission characteristics and preserve data flow types and priorities, the mapped data must be integrated into the TSN data link layer frame format, as illustrated in Fig. 3. The mapped fields are highlighted in dark to denote the conversion process.

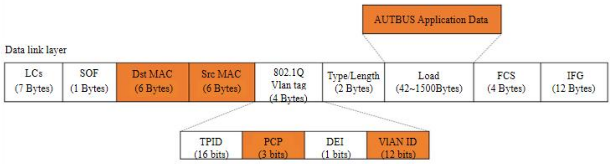


Fig. 3. AUTBUS converts to the TSN frame format at the data link layer

2) *Comprehensive Network Frame Format Conversion*

For comprehensive network conversion, the TSN data link layer is extended with network and transport layers. IPv6 is used for network layer addressing and UDP for the transport layer, which is essential for transmitting real-time data in industrial control systems. As illustrated in Fig. 4, convert and map AUTBUS data to TSN using IPv6 and UDP protocols. Specifically, convert AUTBUS short addresses to IPv6 addresses for the network layer. Place the AUTBUS payload into the UDP payload at the transport layer. At the data link layer, map AUTBUS data flow types and priorities to the corresponding TSN frame fields. Additionally, map AUTBUS flow identification to the TSN VLAN ID.

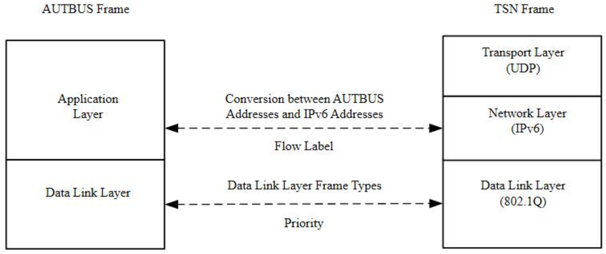


Fig. 4. Comprehensive Network Frame Format Conversion Process for AUTBUS and TSN

The TSN frame format, after converting from AUTBUS and adding the network and transport layers, is illustrated in Fig. 5.

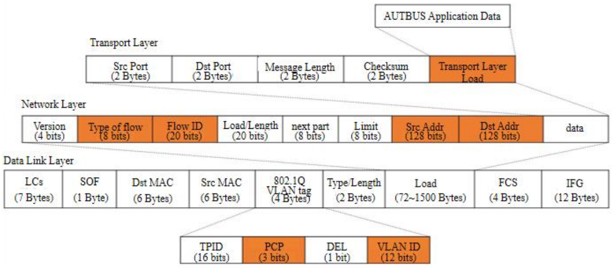


Fig. 5. TSN Frame Format After AUTBUS Conversion and Integration of Network and Transport Layers

B. *Priority Mapping Between AUTBUS and TSN*

1) *Data Link Layer Priority Mapping*

Data within AUTBUS is classified into four distinct types: real-time periodic, real-time non-periodic, non-real-time, and configuration. These are transmitted using either P/S (Producer/Subscriber) or C/S (Client/Server) modes. In TSN, eight priority levels are utilized, which are defined by the PCP field in the VLAN tag, with higher values representing higher priorities. To ensure alignment between AUTBUS and TSN priorities, a mapping table has been established that aligns AUTBUS data with the corresponding TSN VLAN priorities, maintaining consistency across the network.

AUTBUS data is mapped to higher TSN priority levels based on its inherent characteristics, with data following the P/S model being given higher priority than that following the C/S model. Detailed information regarding this mapping is provided in Table 1. For conversions from AUTBUS to TSN, the AST field is employed by the protocol conversion module to determine the data type, and the TSN priority is set accordingly. In the reverse process, where TSN data is converted back to AUTBUS, TSN data types are mapped to AUTBUS types, and the AST field is updated accordingly.

TABLE I. PRIORITY MAPPING BETWEEN AUTBUS DATA TYPE AND TSN DATA LINK LAYER

AUTBUS Data Type	TSN Data Link Layer Priority
Real-time periodic data (P/S mode)	Priority 7
Real-time aperiodic data (P/S mode)	Priority 6
Real-time aperiodic data (C/S mode)	Priority 5
Non-real-time data (P/S mode)	Priority 4
Non-real-time data (C/S mode)	Priority 3
Configuration data (P/S mode)	Priority 2
Configuration data (C/S mode)	Priority 1

2) *IPv6 Layer Priority Mapping*

In TSN using IPv6, priority mapping is essential. IPv6 assigns priorities via an 8-bit traffic type field and a 20-bit flow label. AUTBUS data types, such as real-time and non-real-time data, must be mapped to these fields to ensure quality. Real-time periodic data is mapped to the highest IPv6 priority for optimal service.

Table 2 illustrates the priority mapping between AUTBUS data types and IPv6. The first 3 bits of the 8-bit traffic type field indicate priority (0-7), with higher numbers representing higher priority. The remaining bits are set to 0.

TABLE II. MAPPING RELATIONSHIP BETWEEN AUTBUS DATA TYPE AND TSN NETWORK LAYER PRIORITY

AUTBUS Data Type	TSN IPv6 Layer Priority
Real-time periodic data (P/S mode)	Priority 7
Real-time aperiodic data (P/S mode)	Priority 6
Real-time aperiodic data (C/S mode)	Priority 5
Non-real-time data (P/S mode)	Priority 4
Non-real-time data (C/S mode)	Priority 3
Configuration data (P/S mode)	Priority 2
Configuration data (C/S mode)	Priority 1

C. *AUTBUS and TSN Stream Identification Mapping*

Flow identification is essential for accurate data flow management. AUTBUS uses DataID, while TSN employs StreamID, which combines VLAN ID and source MAC address. To ensure unique flow identification across networks, a mapping from DataID to StreamID is implemented. Additionally, the Multiple VLAN Registration Protocol (MVRP) is employed, this paper employs the Multiple VLAN Registration Protocol (MVRP).

D. *AUTBUS and TSN Address Mapping*

1) *AUTBUS and TSN Network Addressing Methods*

AUTBUS employs NAOID for addressing, comprising Domain ID, Device ID, and Application Object ID. Domain ID defines the virtual bus domain, Device ID specifies the device number (with ranges for initialization, valid, multicast, and broadcast IDs), and Application Object ID identifies application objects, ensuring unique identification within the

AUTBUS network. On the other hand, TSN utilizes MAC and IPv6 addresses for addressing.

2) *AUTBUS Address to MAC Address Mapping*

In TSN networks using MAC address layer communication, mapping AUTBUS addresses to MAC addresses is essential, as illustrated in Fig. 6.

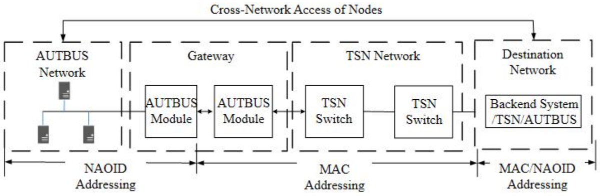


Fig. 6. NAOID Address, MAC Address, and IPv6 Address Addressing Diagram

When an AUTBUS node communicates with a TSN node, the gateway maps the AUTBUS NAOID to a MAC address for transmission. Conversely, the gateway translates the MAC address back to AUTBUS NAOID for accessing AUTBUS nodes. The AUTBUS NAOID (4 bytes) is mapped to the TSN MAC address (6 bytes) by placing the AUTBUS address in the lower 32 bits of the MAC address and setting the upper 16 bits to 1.

3) *AUTBUS Address to IPv6 Address Mapping*

In TSN with IPv6 and UDP, AUTBUS addresses are mapped to IPv6 addresses for communication. The gateway converts AUTBUS NAOID to IPv6 addresses for TSN nodes and vice versa. IPv6 addresses, which are 128 bits divided into 8 segments, are mapped by setting the first 7 bits to the local address prefix, placing the AUTBUS NAOID in the last 4 bytes, and setting the remaining bits to 0. This mapping enables TSN to access AUTBUS nodes using IPv6 addresses.

IV. HARDWARE AND SOFTWARE DESIGN OF PROTOCOL CONVERSION GATEWAY

A. *Protocol Conversion Gateway Hardware Design*

To address the functional requirements for AUTBUS and TSN protocol conversion, a solution employing a master control chip and a protocol chip has been developed. The protocol chip handles communication mechanisms, thereby simplifying protocol coding, while the master control chip is responsible for data protocol conversion. The hardware design of the gateway, as shown in Fig. 7, consists of three main modules: the AUTBUS communication module, the protocol conversion module, and the TSN communication module.

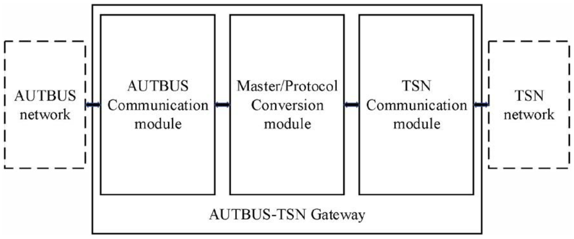


Fig. 7. General hardware design of AUTBUS-TSN gateway

1) *AUTBUS Communication Module*

The AUTBUS communication module is tasked with managing interactions within the AUTBUS network. It handles the reception and forwarding of data streams from AUTBUS nodes. Additionally, the module is responsible for receiving TSN data streams that have been converted by the protocol conversion module and directing them to the appropriate nodes within the AUTBUS network. This functionality is facilitated by the AUTBUS communication module developed by Beijing Neuron, utilizing the KY3001 AUTBUS chip, which supports both AUTBUS bus and SPI interfaces.

2) *Protocol Conversion Module*

This core module of the gateway is responsible for data conversion between AUTBUS and TSN protocols, encompassing data parsing, mapping, flow control, and packet encapsulation. It employs NXP's LS1028A main controller, which features a 64-bit Arm® Cortex®-A72 dual-core processor, providing high reliability and performance. The LS1028A includes various communication peripherals such as 2.5G TSN switch ports, 2.5G TSN MAC ports, a 2.5G Ethernet controller, and SPI interfaces. Additionally, it supports the Linux operating system, facilitating the development of protocol conversion applications.

3) *TSN Communication Module*

This module manages communication with the TSN, handling data streams from TSN nodes and forwarding AUTBUS data streams, converted by the protocol conversion module, to target TSN nodes. Integrated into the main control chip, it utilizes Qualcomm ’ s QSGMII PHY QCA8075 and SGMII PHY AR8031-AL1B for 2.5G TSN switch and MAC signal output, ensuring Ethernet interface support for TSN functions. Data transmission status is indicated by an LD3 LED.

B. *Protocol Conversion Gateway Software Design*

The software implementation framework for the AUTBUS network and TSN protocol conversion gateway, which is illustrated in Fig. 8, demonstrates the key components and their interactions. This framework comprises two main components: the AUTBUS program and the main control program. The AUTBUS program handles communication with AUTBUS devices, managing data reception and forwarding through the KY3001 bus chip and its driver. It transmits data to the protocol conversion module via SLIP serial port and receives data from the protocol conversion module for forwarding to the AUTBUS network. The main control program is responsible for processing both AUTBUS and TSN data, performing functions such as frame format conversion, priority mapping, flow identification, and address translation to form TSN or AUTBUS frames. Additionally, it manages communication with TSN devices for data reception and forwarding.

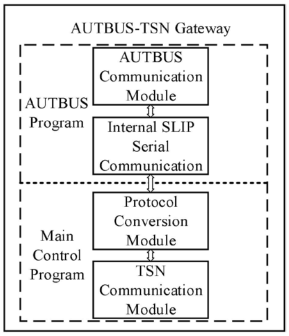


Fig. 8. AUTBUS-TSN gateway software implementation framework

1) Implementation of Priority Mapping and Flow Identification

Based on the flow identification mapping design, priority mapping between AUTBUS and TSN includes both the data link layer and the comprehensive network conversions. AUTBUS data types are mapped to TSN categories and priorities, filling the VLAN PCP and IPv6 traffic class fields. The IPv6 flow label is derived from AUTBUS data types, with real-time periodic data using it for QoS guarantees, while others default to 0. The protocol conversion module maps AUTBUS BlockID to TSN StreamID as follows:

a) Step 1

The AUTBUS communication module joins the network and allocates communication resources. The AUTBUS node then completes data acquisition and interacts with the module.

b) Step 2

The AUTBUS communication module receives bus data and forwards it to the protocol conversion module via the SLIP serial port.

c) Step 3

The protocol conversion module verifies AUTBUS data integrity and compliance. Failed packets are discarded and re-received. If the packet passes, the module determines the data type and transmission mode. Real-time periodic data receives the highest priority, enhanced by the use of IPv6 flow labels. Real-time non-periodic data is prioritized next, followed by non-real-time data, with configuration data receiving the lowest priority.

d) Step 4

The protocol conversion module updates the PCP field based on the priority mapping, extracts the BlockID from the AUTBUS data, inserts it into the VLAN ID field, and registers the VLAN ID using the MVRP protocol.

e) Step 5

The source MAC address and VLAN ID are used to determine the StreamID, ensuring unique identification of AUTBUS data within the TSN.

2) Implementation of AUTBUS Network Address to MAC Address Mapping

In the AUTBUS network, NAOID addresses, consisting of 4 bytes, are used for addressing. During mapping, the NAOID is placed in the lower 4 bytes of the MAC address, while the upper bits are set to 1. After the AUTBUS network allocates communication resources and initiates bus communication, the protocol conversion unit receives and processes data packets from the AUTBUS communication unit. It extracts the NAOID from the packets, constructs the source MAC address, and forms a TSN data frame. The Multiple VLAN Registration Protocol (MVRP) is then used to register VLAN information with the destination port.

3) Implementation of AUTBUS Network Address to IPv6 Address Mapping

The TSN employs IPv6 with UDP at the transport layer, utilizing IPv6 addressing. To map AUTBUS addresses to TSN IPv6 addresses, AUTBUS source addresses are converted into unique local IPv6 addresses.

In the conversion process, AUTBUS network addresses are mapped to IPv6 addresses in the unique local address format. The IPv6 packet is constructed by populating the IPv6 header with the flow label, traffic class, destination address, and source address fields. To ensure VLAN-tagged data flows are properly recognized across TSN ports, the Multiple VLAN Registration Protocol (MVRP) registers VLAN information for each port.

RESULTS AND ANALYSIS

In the verification process, the AUTBUS-TSN gateway was integrated with the AUTBUS industrial real-time broadband bus system, which includes three layers: monitoring (supervisory control software), field control (PLC controllers), and field devices (transmitters and AUTBUS terminal nodes). The gateway interfaces with the field device layer via AUTBUS and the TSN infrastructure, which comprises switches and nodes. PLC controllers are connected to TSN nodes, facilitating the integration of the AUTBUS system with the TSN. This configuration enables long-distance, high-volume data communication via TSN while ensuring real-time performance. The gateway test platform is illustrated in Fig. 9.

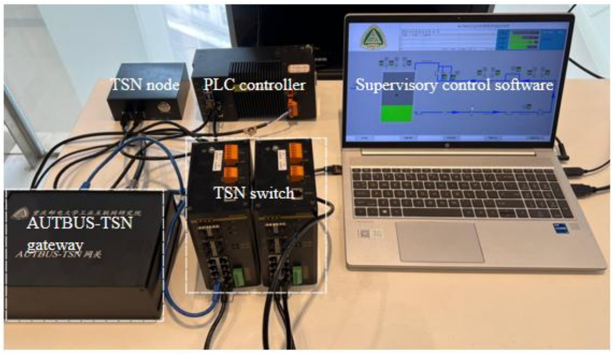
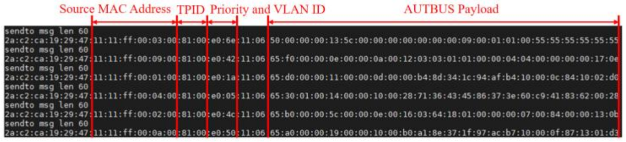


Fig. 9. Test platform monitoring layer and control layer physical

A. Protocol Conversion Testing

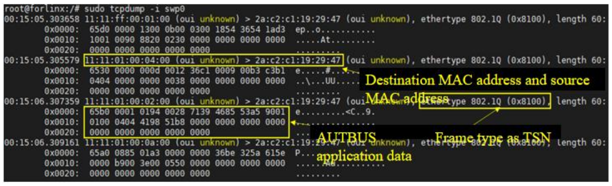
1) Data Link Layer Frame Format Conversion Testing

Data link layer frame format conversion involves mapping key fields from the AUTBUS network frame format to the TSN data link layer frame format, with the AUTBUS application layer payload being mapped to the TSN data link layer payload. The resulting TSN frame format after conversion from AUTBUS data frames is illustrated in Fig. 10.



```
sendto msg len 60
2a:c2:ca:19:29:47:11:11:ff:00:03:00:81:00:e0:6e:11:06 50:00:00:00:13:5c:00:00:00:00:00:00:09:00:01:01:00:55:55:55:55:55:55
sendto msg len 60
2a:c2:ca:19:29:47:11:11:ff:00:09:00:81:00:e0:42:11:06 65:f0:00:00:0e:00:00:0a:00:12:03:03:01:01:00:00:04:04:00:00:00:00:17:0e
sendto msg len 60
2a:c2:ca:19:29:47:11:11:ff:00:01:00:81:00:e0:1a:11:06 65:d0:00:00:11:00:00:0d:00:00:b4:8d:34:1c:94:af:b4:10:00:0c:84:10:02:d0
sendto msg len 60
2a:c2:ca:19:29:47:11:11:ff:00:04:00:81:00:e0:05:11:06 65:30:01:00:14:00:00:10:00:28:71:36:43:45:86:37:3e:60:c9:41:83:62:00:28
sendto msg len 60
2a:c2:ca:19:29:47:11:11:ff:00:02:00:81:00:e0:4c:11:06 65:b0:00:00:5c:00:00:0e:00:16:03:64:18:01:00:00:00:07:00:84:00:00:13:0b
sendto msg len 60
2a:c2:ca:19:29:47:11:11:ff:00:0a:00:81:00:e0:50:11:06 65:a0:00:00:19:00:00:10:00:b0:a1:8e:37:1f:97:ac:b7:10:00:0f:87:13:01:d3
```

Fig. 10. AUTBUS converts the data Link layer TSN frame



```
root@forlinx:/# sudo tcpdump -i swp0
00:15:05.303658 11:11:ff:00:01:00 (oui unknown) > 2a:c2:c1:19:29:47 (oui unknown), ethertype 802.1Q (0x8100), length 60:
        0x0000:  65d0 0000 1300 0b00 0300 1854 3654 1ad3  ep..o...........
        0x0010:  1001 0090 8820 0230 0000 0000 0000 0000  .....At.........
        0x0020:  0000 0000 0000 0000 0000                 ..........
00:15:05.305579 11:11:01:00:04:00 (oui unknown) > 2a:c2:c1:19:29:47 (oui unknown), ethertype 802.1Q (0x8100), length 60:
        0x0000:  6530 0000 000d 0012 36c1 0009 00b3 c3b1  e.....#...
        0x0010:  0404 0000 0000 0038 0000 0000 0000 0000  ..\...UU.....
        0x0020:  0000 0000 0000 0000 0000                 ..........
00:15:06.307359 11:11:01:00:02:00 (oui unknown) > 2a:c2:c1:19:29:47 (oui unknown), ethertype 802.1Q (0x8100), length 60:
        0x0000:  65b0 0001 0194 0028 7139 4685 53a5 9001  e.........<C..9.
        0x0010:  0100 0404 4198 51b8 0000 0000 0000 0000  ................
        0x0020:  0000 0000 0000 0000 0000                 ...
00:15:06.309161 11:11:01:00:0a:00 (oui unknown) > 2a:c2:c1:19:29:47 (oui unknown), ethertype 802.1Q (0x8100), length 60:
        0x0000:  65a0 0885 01a3 0000 0000 36be 325a 615e  P...
        0x0010:  0000 b900 3e00 0550 0000 0000 0000 0000  ......Ad..........
        0x0020:  0000 0000 0000 0000 0000                 ..........
```

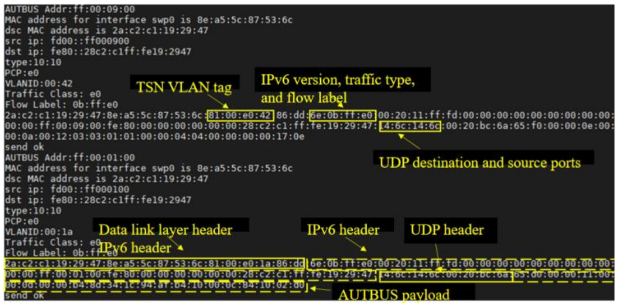
Fig. 11. AUTBUS converts the data Link layer TSN frame

After the gateway forwards TSN data utilizing MAC addressing, the tcpdump packet capture tool on the gateway's

Linux system was employed to capture packets at the gateway's output port. The captured packet, illustrated in Fig. 11, encompasses the destination MAC address, source MAC address, frame type, and AUTBUS application payload. The captured data conforms to the design specifications.

2) Comprehensive Network Frame Format Conversion Test

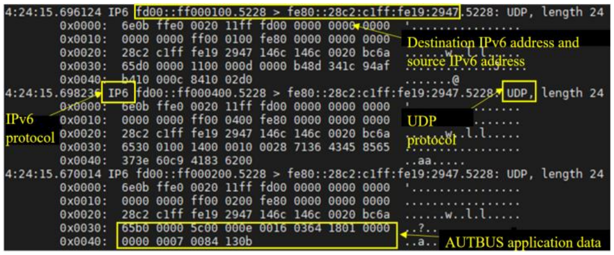
In TSN, the introduction of upper-layer protocols IPv6 and UDP involves mapping AUTBUS data types to both data link layer priorities and IPv6 header fields, including traffic class and flow label. Additionally, the AUTBUS address is mapped to the source IPv6 address. The converted frame format incorporates the data link layer, network layer, transport layer, and AUTBUS payload, as illustrated in Fig. 12.



```
AUTBUS Addr:ff:00:09:00
MAC address for interface swp0 is 8e:a5:5c:87:53:6c
dsc MAC address is 2a:c2:c1:19:29:47
src ip: fd00::ff000900
dst ip: fe80::28c2:c1ff:fe19:2947
type:10:10
PCP:e0
VLANID:00:42
Traffic Class: e0
Flow Label: 0b:ff:e0
2a:c2:c1:19:29:47:8e:a5:5c:87:53:6c:81:00:e0:42:86:dd:6e:0b:ff:e0:00:20:11:ff:fd:00:00:00:00:00:00:00:00:00:
00:00:ff:00:09:00:fe:80:00:00:00:00:00:00:28:c2:c1:ff:fe:19:29:47:14:6c:14:6c:00:20:bc:6a:65:f0:00:00:0e:00:
00:0a:00:12:03:03:01:01:00:00:04:04:00:00:00:00:17:0e
send ok
AUTBUS Addr:ff:00:01:00
MAC address for interface swp0 is 8e:a5:5c:87:53:6c
dsc MAC address is 2a:c2:c1:19:29:47
src ip: fd00::ff000100
dst ip: fe80::28c2:c1ff:fe19:2947
type:10:10
PCP:e0
VLANID:00:1a
Traffic Class: e0
Flow Label: 0b:ff:e0
2a:c2:c1:19:29:47:8e:a5:5c:87:53:6c:81:00:e0:1a:86:dd:6e:0b:ff:e0:00:20:11:ff:fd:00:00:00:00:00:00:00:00:00:
00:00:ff:00:01:00:fe:80:00:00:00:00:00:00:28:c2:c1:ff:fe:19:29:47:14:6c:14:6c:00:20:bc:6a:65:d0:00:00:11:00:
00:0d:00:00:b4:8d:34:1c:94:af:b4:10:00:0c:84:10:02:d0
send ok
```

Fig. 12. AUTBUS converts to IPv6 TSN frames

After the gateway forwards the TSN frame with IPv6 and UDP protocols, packets are captured at the gateway's output port using tcpdump. The captured packets, shown in Fig. 13, include data link layer, network layer, and transport layer information, along with the AUTBUS application payload. The mapped fields align with design expectations.

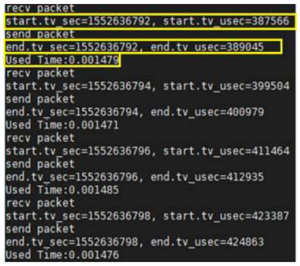


```
4:24:15.696124 IP6 fd00::ff000100.5228 > fe80::28c2:c1ff:fe19:2947.5228: UDP, length 24
        0x0000:  6e0b ffe0 0020 11ff fd00 0000 0000 0000  '...............
        0x0010:  0000 0000 ff00 0100 fe80 0000 0000 0000
        0x0020:  28c2 c1ff fe19 2947 146c 146c 0020 bc6a
        0x0030:  65d0 0000 1100 000d 0000 b48d 341c 94af
        0x0040:  b410 000c 8410 02d0                      .......@
4:24:15.69823 IP6 fd00::ff000400.5228 > fe80::28c2:c1ff:fe19:2947.5228: UDP, length 24
        0x0000:  6e0b ffe0 0020 11ff fd00 0000 0000 0000  '
        0x0010:  0000 0000 ff00 0400 fe80 0000 0000 0000
        0x0020:  28c2 c1ff fe19 2947 146c 146c 0020 bc6a
        0x0030:  6530 0100 1400 0010 0028 7136 4345 8565
        0x0040:  373e 60c9 4183 6200                      ..aa.....
4:24:15.670014 IP6 fd00::ff000200.5228 > fe80::28c2:c1ff:fe19:2947.5228: UDP, length 24
        0x0000:  6e0b ffe0 0020 11ff fd00 0000 0000 0000  '...............
        0x0010:  0000 0000 ff00 0200 fe80 0000 0000 0000  ................
        0x0020:  28c2 c1ff fe19 2947 146c 146c 0020 bc6a  ......w..l.l.....
        0x0030:  65b0 0000 5c00 000e 0016 0364 1801 0000  ..?..
        0x0040:  0000 0007 0084 130b                      ..a..
```

Fig. 13. The TSN IPv6 packet forwarded by the gateway

B. Protocol Conversion Latency Test

The protocol conversion time is measured from the moment the gateway's protocol conversion module identifies data in AUTBUS format until it is transmitted in TSN format. This time is recorded and shown in Fig. 14. In this experiment, 50 randomly selected data packets were analyzed. The maximum conversion time was 1.534 ms, the minimum was 1.404 ms, and the average was 1.455 ms. These results validate the gateway's performance in supporting cross-network communication between AUTBUS and TSN, offering practical benefits for improving real-time communication efficiency in industrial systems.



```
recv packet
start.tv_sec=1552636792, start.tv_usec=387566
send packet
end.tv_sec=1552636792, end.tv_usec=389045
Used Time:0.001479
recv packet
start.tv_sec=1552636794, start.tv_usec=399504
send packet
end.tv_sec=1552636794, end.tv_usec=400979
Used Time:0.001471
recv packet
start.tv_sec=1552636796, start.tv_usec=411464
send packet
end.tv_sec=1552636796, end.tv_usec=412935
Used Time:0.001485
recv packet
start.tv_sec=1552636798, start.tv_usec=423387
send packet
end.tv_sec=1552636798, end.tv_usec=424863
Used Time:0.001476
```

Fig. 14. Protocol conversion time print information

CONCLUSION

Addressing the gap in cross-network transmission solutions for AUTBUS and Time-Sensitive Networking (TSN), this paper analyzes the characteristics of both protocols and proposes a conversion method. It details the design of the gateway's hardware and software and establishes a testing platform. The results show that the AUTBUS-TSN protocol conversion gateway effectively supports cross-network data transmission while preserving priority and flow identification. This research offers foundational insights for integrating industrial bus systems with TSN and may aid future developments in this field. Future work will focus on exploring additional performance metrics to comprehensively evaluate system performance.

ACKNOWLEDGMENT

This research was funded by The National Key Research and Development Program of China (2022YFE0204500) and Chongqing Natural Science Foundation (CSTB2023 NSCQ-LZX0123).

REFERENCES

[1] B. Gunjan, A. Singhrova, "A systematic literature review on IoT gateways," Journal of King Saud University-Computer and Information Sciences, vol. 34, no. 10, pp. 9541-9563, 2022.

[2] F. L. d. C. Filho, R. L. Rocha, C. J. B. Abbasy, L. M. C. E. Martins, E. D. Canedo, "QoS scheduling algorithm for a fog IoT gateway," 2019 Workshop on Communication Networks and Power Systems (WCNPS), Brasilia, Brazil, pp. 1-6, 2019.

[3] M. Hemmatpour, M. Ghazivakili, B. Montrucchio, M. Rebaudengo, "DIIG: A Distributed Industrial IoT Gateway," 2017 IEEE 41st Annual Computer Software and Applications Conference (COMPSAC), pp. 755-759, 2017.

[4] Y. X. Chen, Q. M. Xu, J. Zhang, L. Xu, L. Z. Li, "TSN-compatible Industrial Wired/Wireless Multi-protocol Conversion Mechanism and Module," IECON 2022–48th Annual Conference of the IEEE Industrial Electronics Society, pp. 1-6, 2022.

[5] C. Sun, G. Liu, Z. Xu, D. Hu and F. Zheng, "Design of Cloud-based IoT Gateway for CAN Bus to Modbus RTU Integration," 2022 China Automation Congress (CAC), pp. 1-5, 2020.

[6] P.P. Ray, N. Thapa, D. Dash, "Implementation and performance analysis of interoperable and heterogeneous IoT-edge gateway for pervasive wellness care," IEEE Transactions on Consumer Electronics, vol. 65, no. 4, pp. 464-473, 2019.

[7] J. Xie and Q. Gao, "Design and Implementation of Embedded Protocol Conversion Gateway for Intelligent Buildings," 2019 IEEE 10th International Conference on Software Engineering and Service Science (ICSESS), Beijing, China, pp. 1-5, 2019

[8] W. Wang, L. Zhu, T. H. Luan and C. Li, "DoIP: A Parallel Protocol Conversion Gateway for DMR over Internet Protocol," 2023 IEEE 97th Vehicular Technology Conference (VTC2023-Spring), Florence, Italy, pp. 1-5, 2023