IDH-CAN: A Hardware based ID Hopping CAN Mechanism with Enhanced Security for automotive Real-Time Applications

吴武飞

湖南大学

信息科学与工程学院

嵌入式系统与网络湖南省重点实验室

A history of the computer taking control

1912
First aircraft autopilot developed by Sperry Corporation

1947
U.S. Air Force C-54 makes a transatlantic flight, including takeoff and landing completely under the control of autopilot

1950s
GM and RCA test vehicles on automated highways using radio controls

2012
Google car passes Nevada driving test on 22 km route in Las Vegas

2015
Audi A7s drive from San Francisco to Las Vegas

2016
Uber and nuTonomy begin testing self-driving taxis in Pittsburgh and Singapore

Around 1900
Completely automated push-button elevators available in small apartments

1925
Houdina Radio Control demonstrates a radio-controlled car operated by a following car

1950s
Automatic push-button elevators introduced en masse

1998
Mercedes-Benz, Toyota and Mitsubishi begin offering adaptive cruise control

2013
Mercedes-Benz S-Class goes 100 km on highways and streets in Germany

2016
An Otto autonomous transport truck drives about 200 km to make a beer delivery in Colorado

THE GLOBE AND MAIL

Comparison of CAN bus Security Methods：

Protection against unauthorized manipulation and replay attacks.

Message authentication code (MAC) in the data field for CAN frame.

Drawback： Additional delay, payload occupy

Note: In case a MAC is used, it is possible to transmit and compare only parts of the MAC. This is known as MAC truncation.

Designing a New Protocol Based on CAN : CAN+.

Drawback： Slow the data rate of transmission， additional delay

ID anonymization—>（ID Hopping）

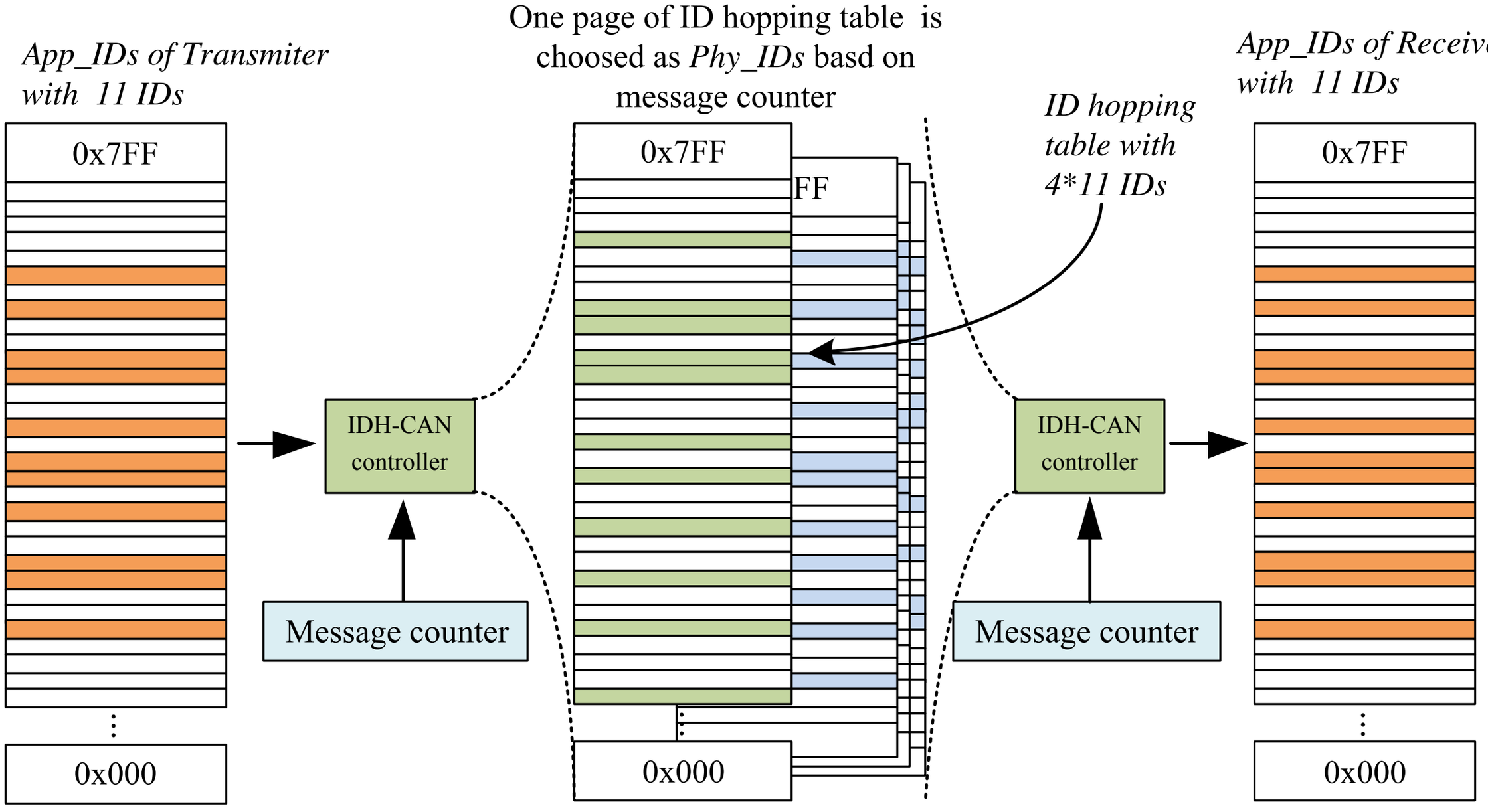
Table 1 Security comparison of CAN Security Enhancements

attack types	MAC[1][2]	CAN+[3]	ID anonymization [4][5]
Reverse engineering	√	×	√
Replay	×	×	√
Sniffing	√	√	√

Constraints of CAN security in autonomous vehicle

Real-time， Bus utilization, Schedulability analysis, Cost, Energy.

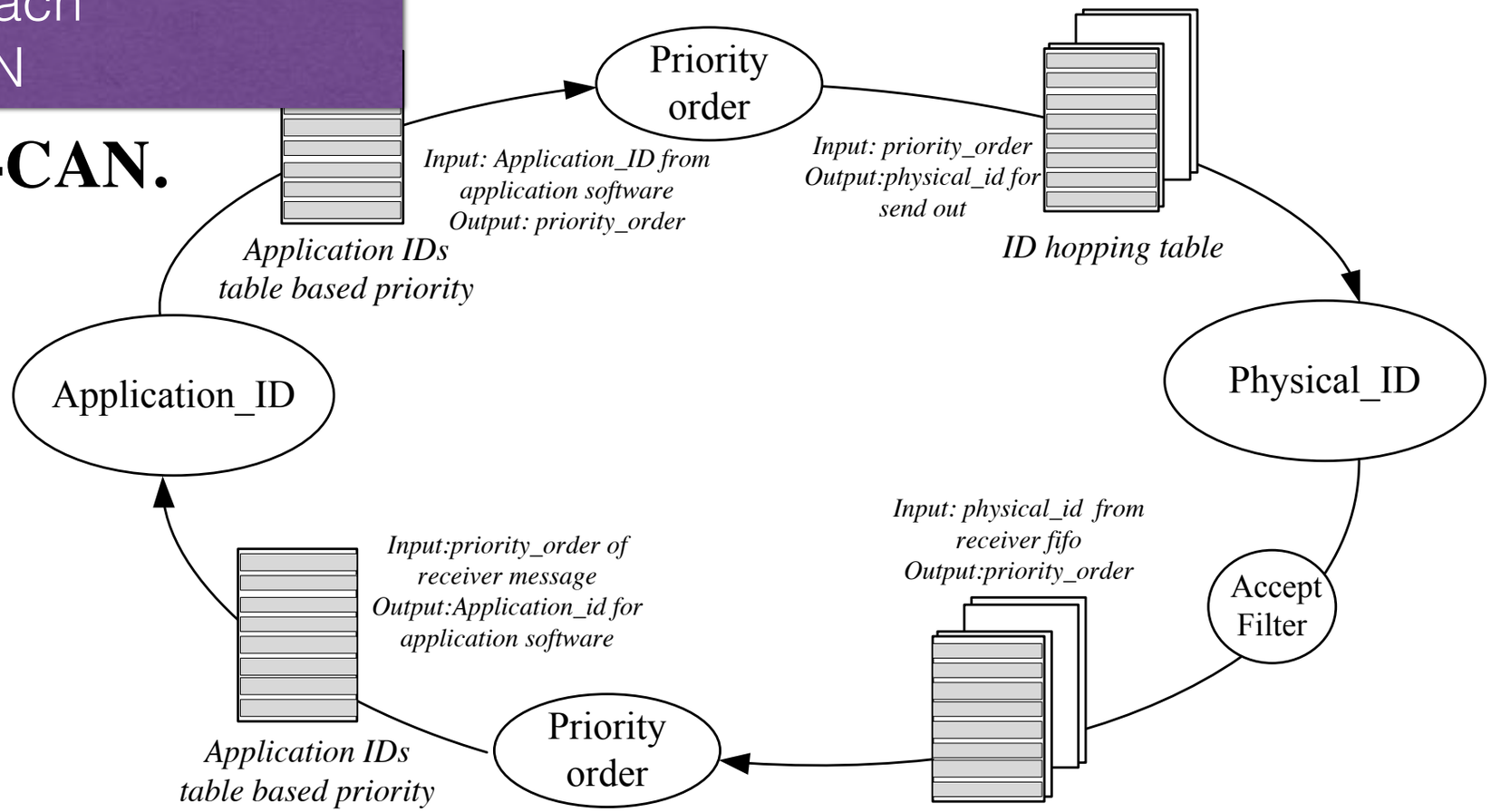
Our approach
IDH-CAN



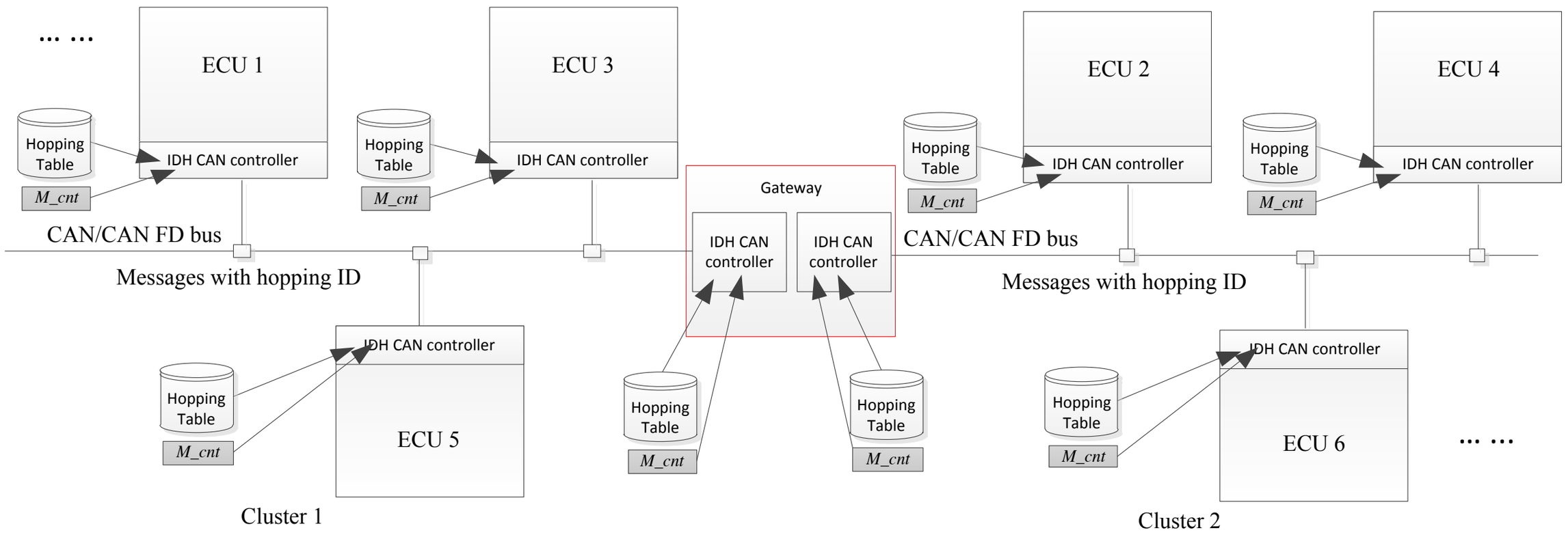
Physical IDs changed when ID hopping happened.

Our approach
IDH-CAN

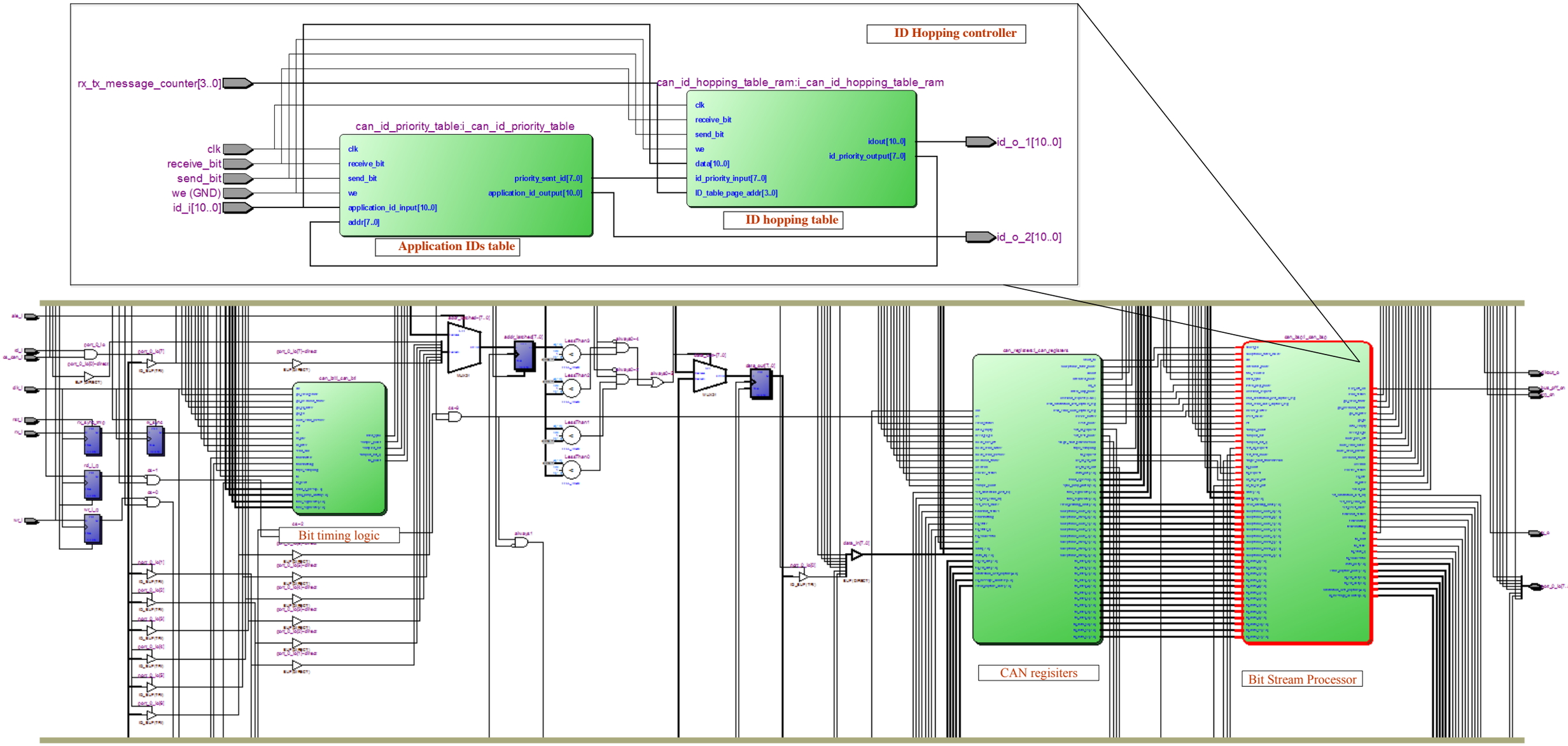
Data flow in IDH-CAN.



ID hopping based on the message counter and ID table.



Hardware based implementation



RTL view of the ID Hopping CAN controller.

Waveform simulation

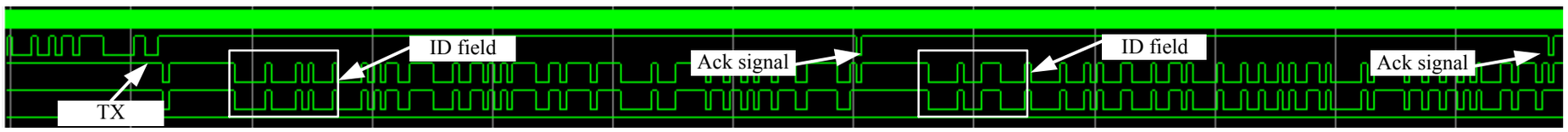
The message ID is written to the register.

```
task send_frame_basic;     // CAN IP core sends frames
  begin
    //write register for message transmitter send out
    write_register(8'd10, 8'h00); // Writing ID[10:3] = 0x00000000
    write_register(8'd11, 8'h08); // Writing ID[2:0] = 0x001, rtr = 0, length = 8
    write_register(8'd12, 8'h56); // data byte 1
    write_register(8'd13, 8'h78); // data byte 2
    write_register(8'd14, 8'h9a); // data byte 3
    write_register(8'd15, 8'hbc); // data byte 4
    write_register(8'd16, 8'hde); // data byte 5
    write_register(8'd17, 8'hf0); // data byte 6
    write_register(8'd18, 8'h0f); // data byte 7
    write_register(8'd19, 8'hed); // data byte 8
```

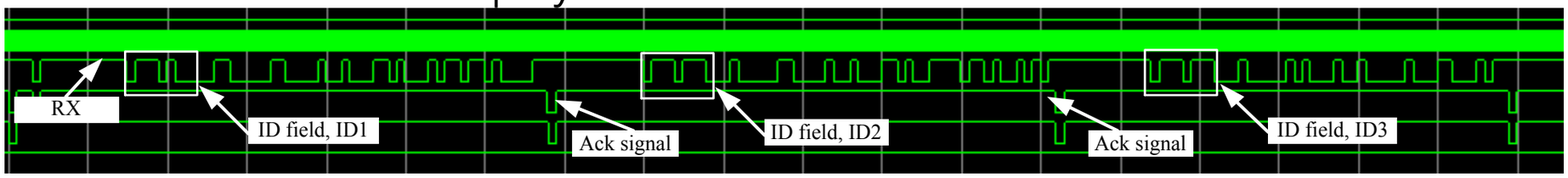
The application layer's ID is written to the register

For transmitter

The waveform on the physical CAN bus.



The waveform on the physical CAN bus.



For receiver

```
# (490884400)
# (490924900) Reading register [20] = 0x0
# (490962400) Reading register [21] = 0x61
# (490999900) Reading register [22] = 0x12
# (491037400) Reading register [23] = 0x0
# (491074900) Reading register [24] = 0x62
# (491112400) Reading register [25] = 0x12
# (491149900) Reading register [26] = 0x34
# (491187400) Reading register [27] = 0x1
# (491224900) Reading register [28] = 0xc2
# (491262400) Reading register [29] = 0x12
# (491262500)
# (491299900)
# (491299900) Writing register [1] with 0x4
# (491337400) Rx buffer released.
```

ID 1

ID 2

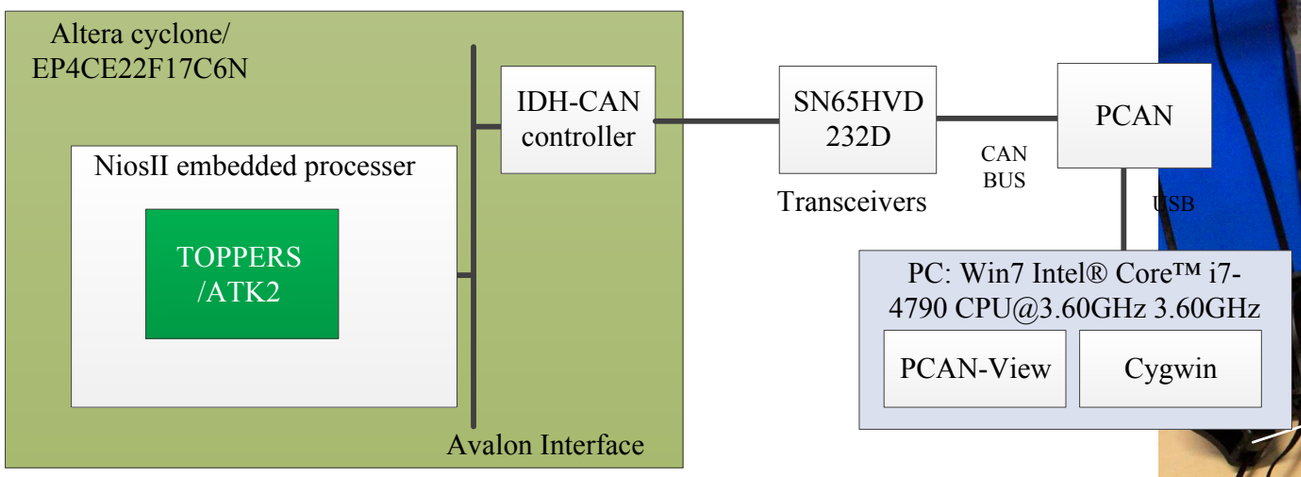
ID 3

The IDs read from the FIFO buffer

The message ID is written to the register.

Evaluation

Block diagram of evaluation platform



Evaluation platform

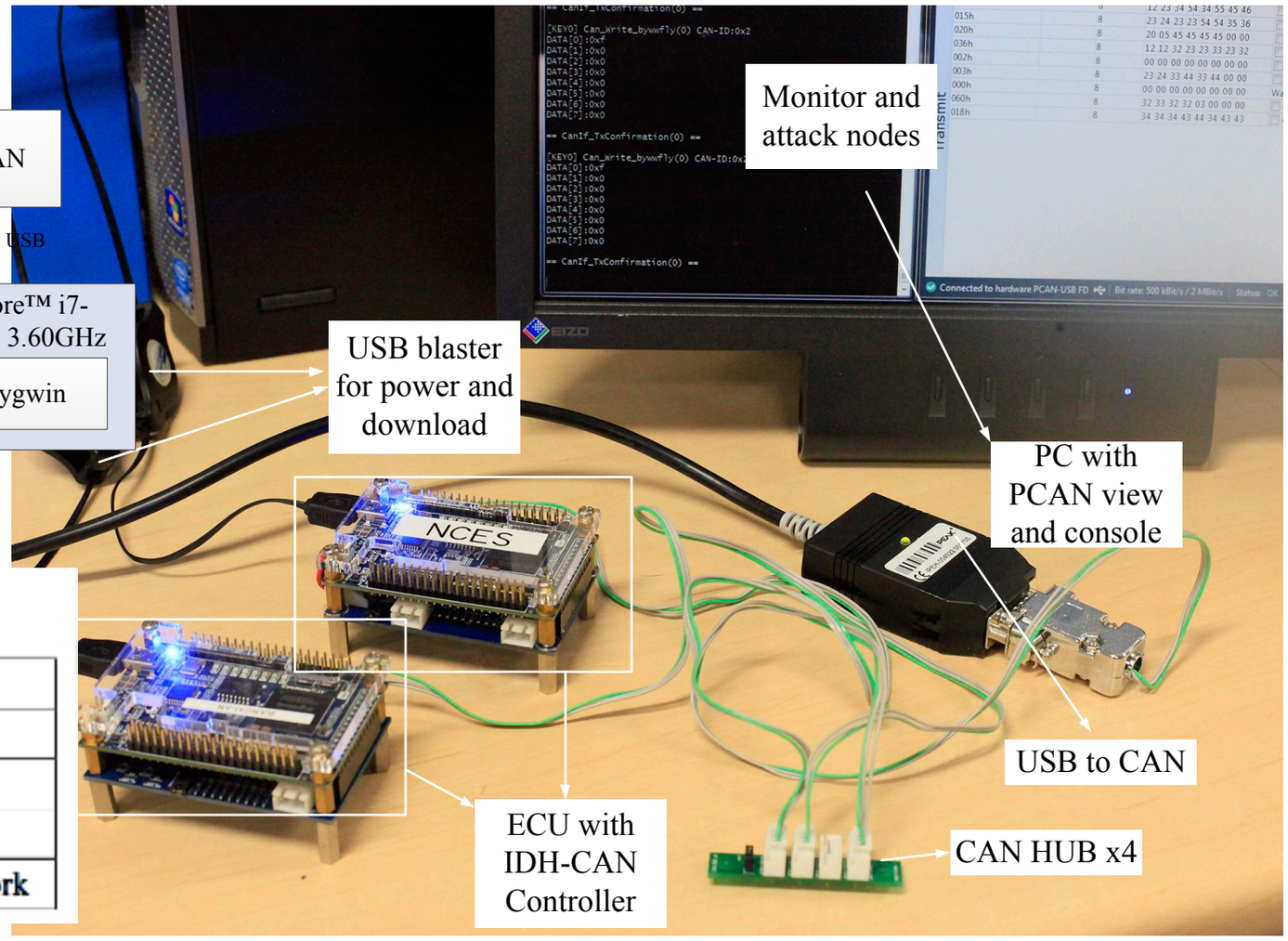


TABLE IV: Tools used for the experiment

Product	Model Name	note
FPGA Platfrom	DE0-Nano	Altera EP4CE22F17C6
Computer	Dell Optiplex 9020	No
Connector	PCAN	Interface Device
SW	PCAN-view, Cygwin	Network Tool for Vehicle Nethork

TABLE II
PERFORMANCE COMPARISON OF CAN SECURITY ENHANCEMENT METHODS

	MAC[12] [13]	CAN+ [15][24][11] [14]	IA-CAN[16]	ID hopping[17]	Proposed
Computational complexity	High	Middle	Low	Low	Low
Time delay	Middle	High	No	Low	No
Play load consumption	High	No need	Middle	No need	No need
Additional messages	No need	Need	No need	Need	One
Schedulability analysis	Easy	Complex	Complex	Complex	Easy

Entropy Analysis

TABLE II: SAE benchmark based message set

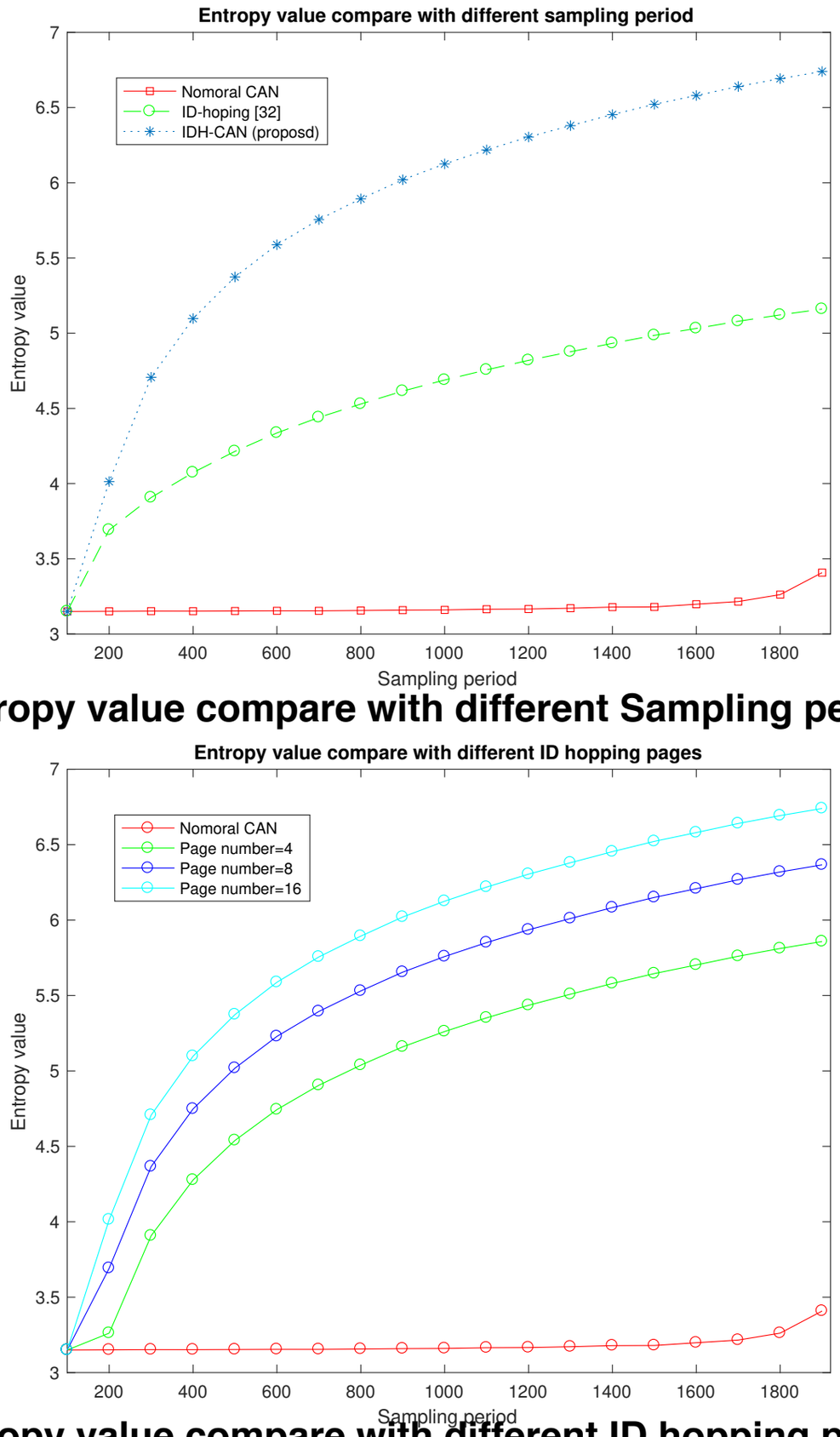
	Message ID	Size (bytes)	T(ms)	D(ms)
1	0x23F	1	50.0	5.0
2	0x24F	2	5.0	5.0
3	0x33F	1	5.0	5.0
4	0x30F	2	5.0	5.0
5	0x31F	6	10.0	5.0
6	0x32F	3	10.0	5.0
7	0x33F	6	50.0	10.0
8	0x34F	4	50.0	20.0
9	0x35F	1	100.0	50.0
10	0x41F	4	100.0	50.0
11	0x43F	1	500.0	50.0
12	0x440	1	1000.0	200.0
13	0x505	2	1000.0	1000.0
14	0x545	8	1000.0	1000.0
15	0x59B	8	1000.0	1000.0
16	0x7DF	8	1200.0	1000.0
17	0x7E0	8	1500.0	1200.0

In this experiments we consider the system consisiting of 200 messages in single CAN cluster (We expanded the message set based on the Table II), the probability of message obtaind based on Equation (9), and the average entropy of IDs in CAN cluster in sampling period T is obtaind by Equation (12). The results of this comparison are shown in Fig.13.

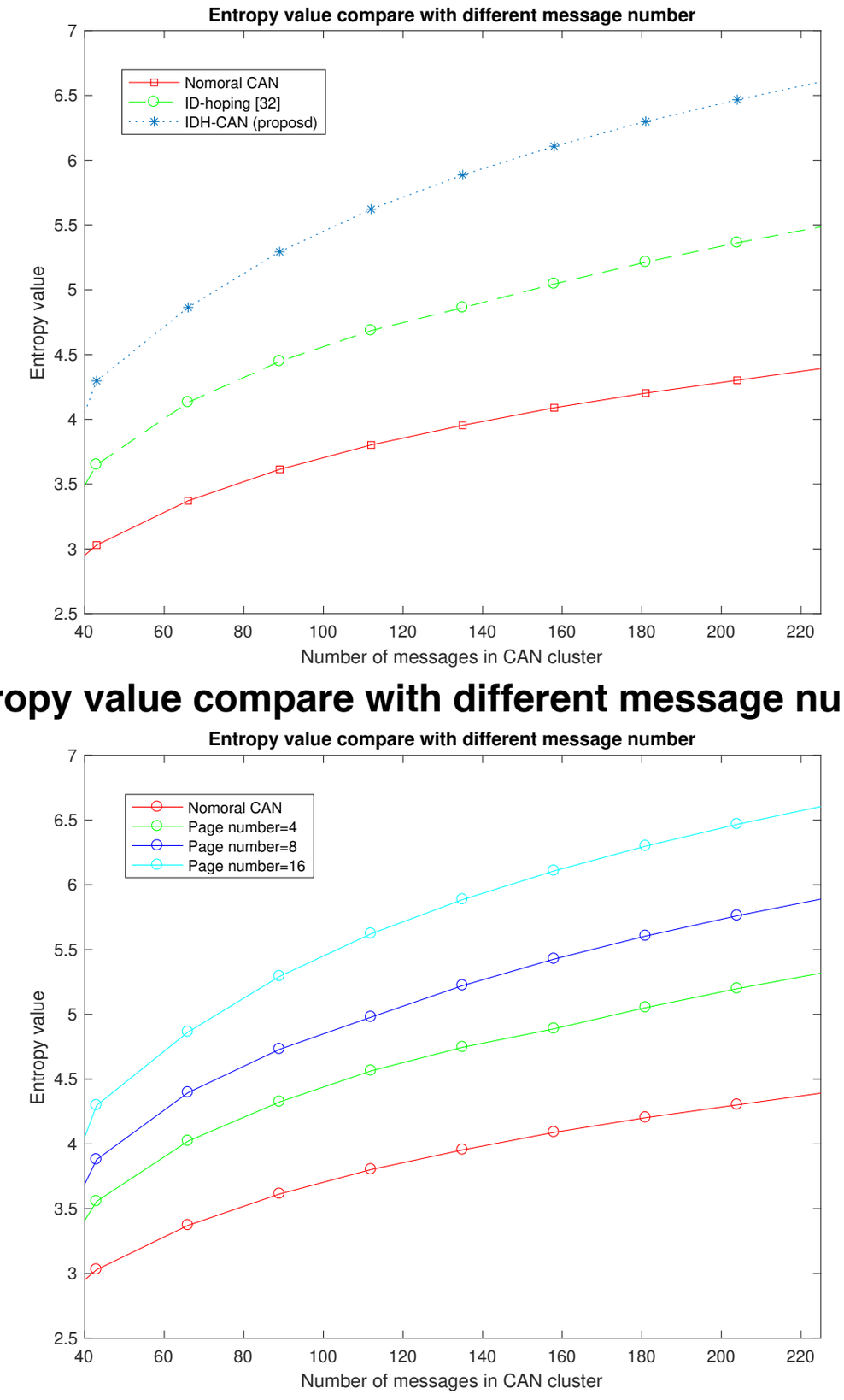
TABLE III: ASIL level and N_pages Configuration

ASIL	N_page	$Memorysize$
A	4	10Kbits
B	8	20Kbits
C	16	40Kbits

Entropy comparison Analysis



Entropy value compare with different Sampling period.



Entropy value compare with different message number.

Entropy value compare with different ID hopping pages.

Entropy value compare with different message number.

Thank you!

日本学习生活方面的归纳总结

一、学习模式：每周讨论+汇报+发表模式

小组讨论追细节；汇报求效率把方向；发表锻炼（课长负责制，时间固定，人员平均一年两次）。

对我而言，更多的学习，在于向同研究室同学的学习和交流。

二、实验室活动：

每年四次聚会（迎新会，忘年会，送别会，结业会）

课长负责组织，自愿参加，全员AA（教师会多点）

三、实验室环境：

软件方面：内部wiki网站（实验室管理，资料），事务秘书。

硬件方面：设备好一点，面积大一点，所有电脑联网的打印机，大型海报打印机。

谢谢～！

提问