Formal Verification of Protocol Stack for MOST Network Service using SPIN

Dong-Ah Lee*, Sanghyun Yoon, Mu-Youl Lee, Hyun-Wook Jin, Junbeom Yoo

Contents

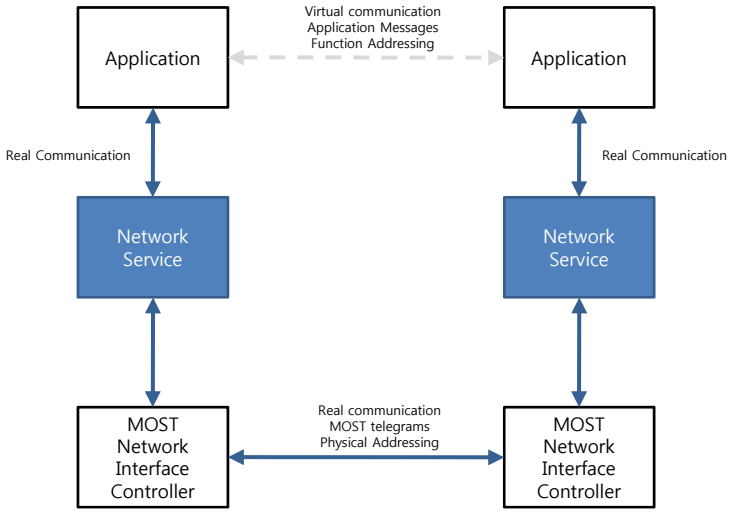
Overview

- Network System for the Information of Automobile Status
  - CAN(Controller Area Network), LIN(Local Interconnect Network)

- Broadband Network System for In-vehicle Multimedia System
  - MOST(Media Oriented Systems Transport)

MOST Network Service

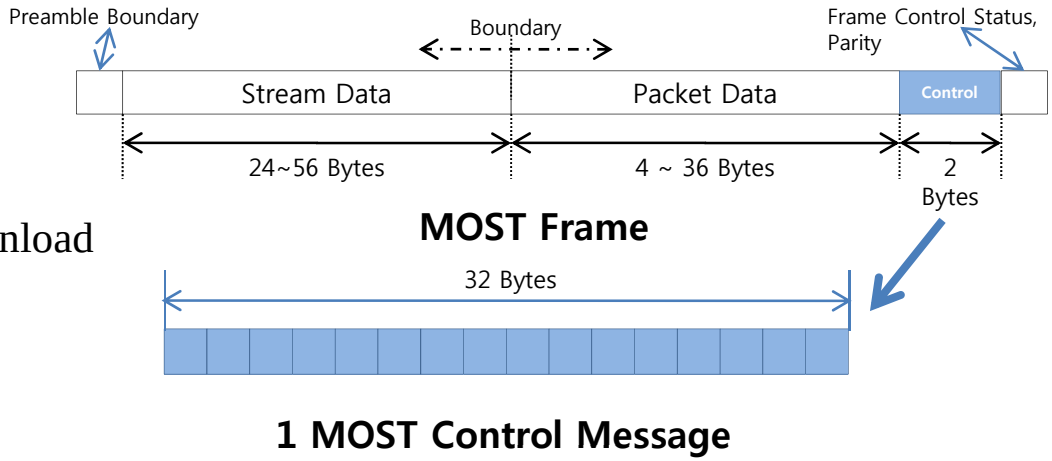
- Communication over MOST
  - Application
    - Virtual communication using Functional addressing
  - Network Service
    - APIs for implementing applications
    - Framework for a MOST device
    - Handling Messages
  - Network Interface Controller
    - Real communication using Physical addressing



MOST network architecture

Protocol Specification

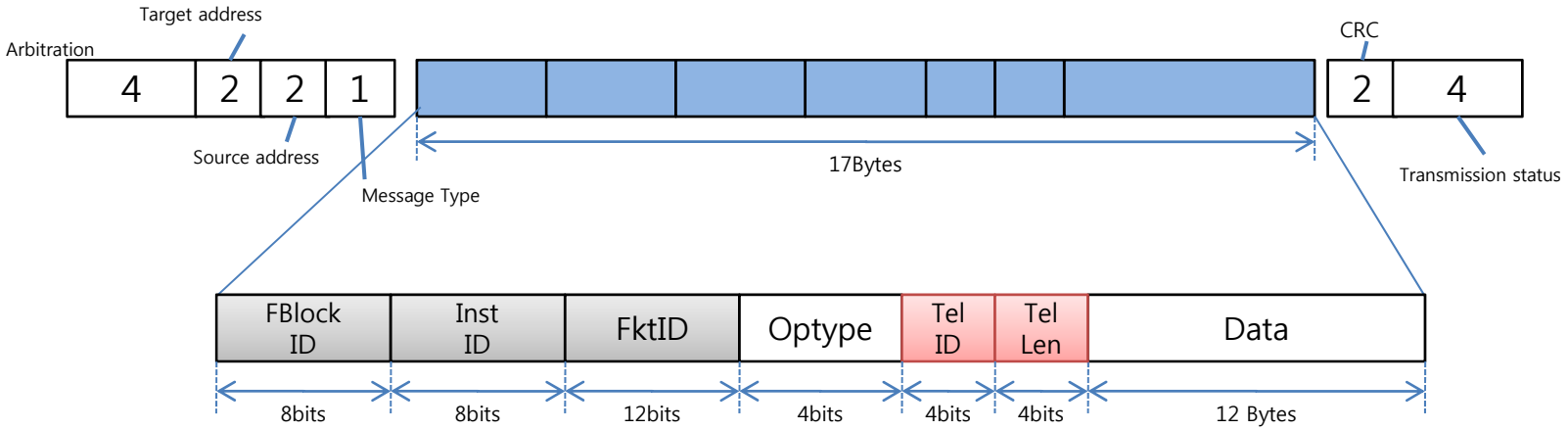
- Stream Data
  - Audio, Video real time Data
- Packet Data
  - Navigation data, software download
- Control Data
  - Controlling of MOST devices



- 1 MOST Control Message = 16 Control Data Frame

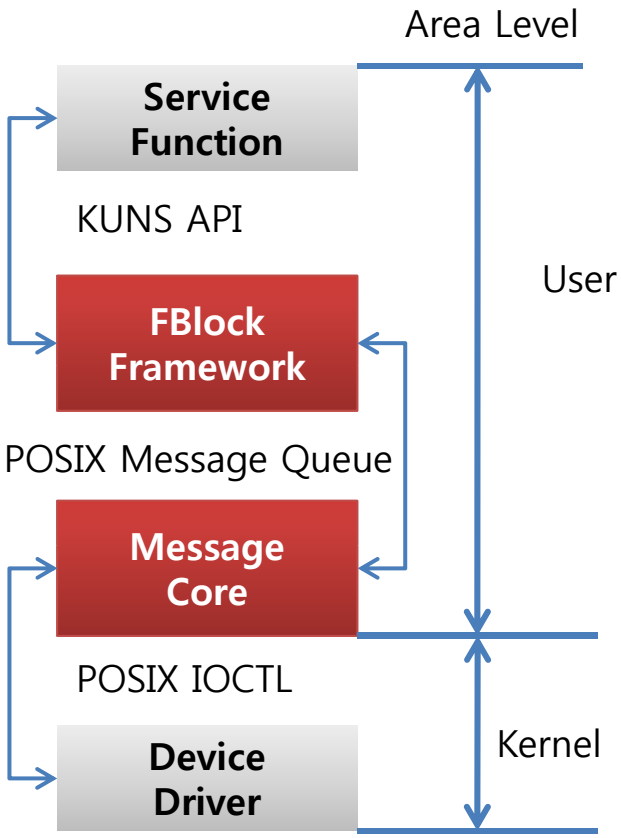
Protocol Specification(Cont'd)

- Control data to manage user application
- 5 Bytes for Header
  - Information of Address (IDs) & Telegram(Tel ID, Tel Len)
- 12 Bytes for User Data



u-OMNiPro

- Structure of u-OMNiPro 1.0
- FBlock Framework
  - develop and manage FBlock
- Message Core
  - communication btw FBlock and Device Driver
- IPC
  - POSIX Message Queue
  - POSIX IOCTL



Overview of
u-OMNiPro 1.0 Architecture

FBlock Framework

- APIs for implementing applications
  - Registration and Unregistration of FBlock
  - Registration and Unregistration of service function
  - Message transportation

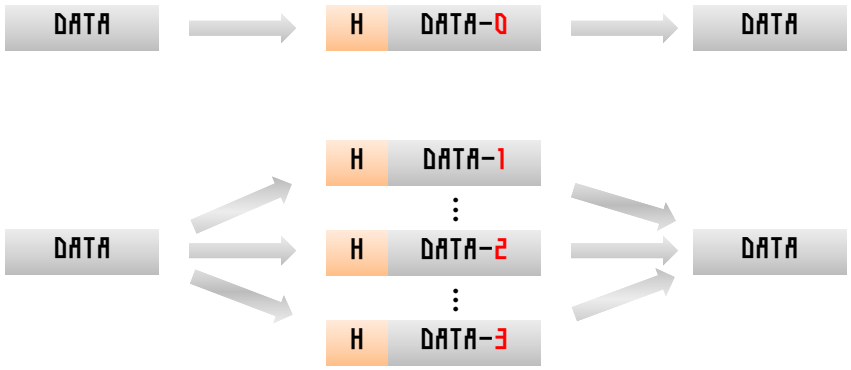
APIs	Description
KUNS_FBlock_Register	FBlock Registration
KUNS_FBlock_Unregister	FBlock Unregistration
KUNS_FuncktionID_Register	Function Registration on FBlock
KUNS_FunctionID_Unregister	Function Unregistration on FBlock
KUNS_FBlock_RUN	FBlock Execution
KUNS_ctrl_recv	Receive Data from Message Core
KUNS_ctrl_send	Send Data from Message Core

FBlock Framework API

Message Core

Single Telegram Messages

Tel ID	Tel Len	Data Byte 0 / Counter	Description
0	0 - C	Data	Single telegram

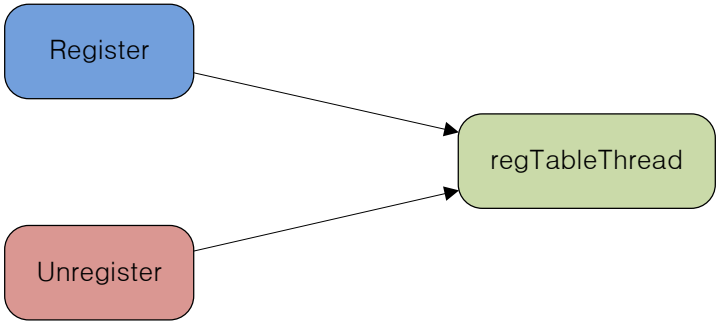


Multiple Telegram Messages

Tel ID	Tel Len	Data Byte 0 / Counter	Description
1	C	0X00	First Telegram
2	C	0x01	Subsequent Telegram
2	C	…	Subsequent Telegram
2	C	0xFF	Subsequent Telegram
2	C	0x00	Subsequent Telegram
2	C	…	Subsequent Telegram
2	C	0x0n – 1	Subsequent Telegram
3	2 - C	0x0n	Last Telegram

Modeling(FBlock Register & Unregister)

- 3 Processes
  - Register
  - Unregister
  - regTableThread
- Register
  - Request Registration of FBlock
- Unregister
  - Request Unregistration of FBlock
- regTableThread
  - Registration & Unregistration of FBlock
  - Manage FBlock Table



FBlock Register & Unregister model

```
typedef NSHandler{      byte fBlock;
                        byte instId;
                        byte status };
```

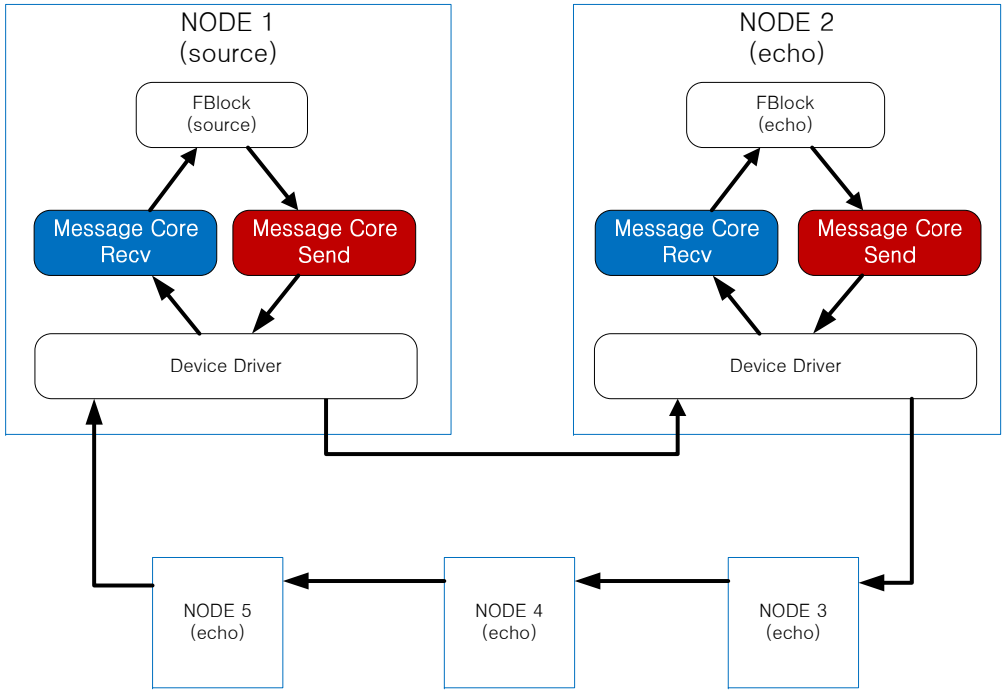
Data Structure

Modeling(Echo)

- 5 Nodes
  - 1 Source Node(4 Processes)
  - 4 Echo Nodes(4 Processes)
- FBlock
  - Source – Generate Messages
  - Echo – Return Messages
- Send / Receive
  - Handling Messages
- Device Driver

```
typedef perfect_Ctrl_Data { byte tgt_addr;
                            short ctrl_data_len;
                            byte ctrl_data[6] };
typedef ctrl_data_t {       byte tgt_addr;
                            byte ctrl_data[6] };
```

Data Structure



u-OMNiPro echo model(5 Nodes)

Verification Result(Register & Unregister of FBlock)

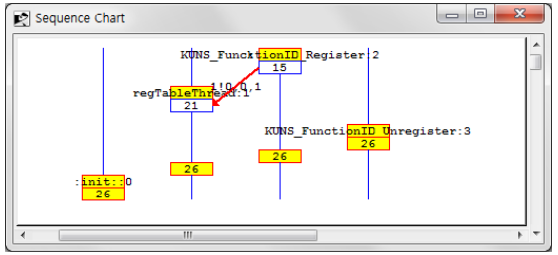
- When the registration function is called with a service function, are properties of the service function registered to the table correctly?

If
::(count < MAX_Table_Size) -> assert(pNsHandler[count-1].fBlock!=FBlock.buff[0]);
*::(count == 0) -> assert(pNsHandler[MAX_FUNCTION * MAX_INSTID - 1].fBlock != FBlock.buff[0]);*
fi;

- When the unregistration function is called with a service function, are properties of the service function unregistered from the table correctly?

assert(pNsHandler[count].fBlock != FBlock.buff[0])

  - Process is terminated by force
  - Unregistration function is not implemented

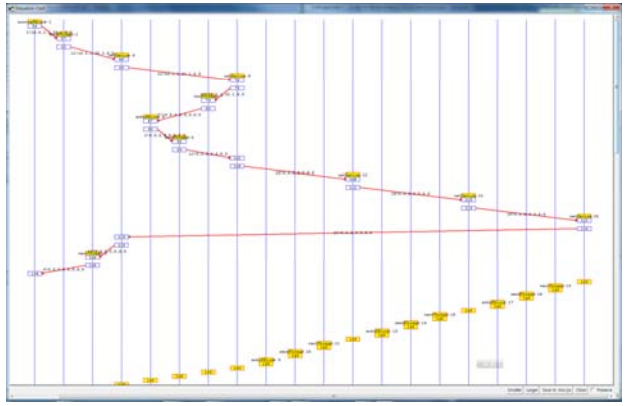


Verification Result(Echo)

- Is a message generated by the source node same with return message from a echo Node?

  assert(recvdCtrlData.ctrl_data_len != sentCtrlData.ctrl_data_llen)

  - Process is terminated by force
  - A size of single telegram is not defined



- If a first telegram was sent, is it guaranteed that a final telegram is always sent eventually?

  []((sendTelID == 1) -> <> (sendTelID == 3))

Conclusion and Future Work

- Conclusion
  - Formal Modeling & Verification for u-OMNiPro 1.0
  - We found two possible problems
    - Unimplemented problem
    - Property outside of model

- Future Work
  - Formal Modeling & Verification for k-OMNiPro (kernel-level OMNiPro)