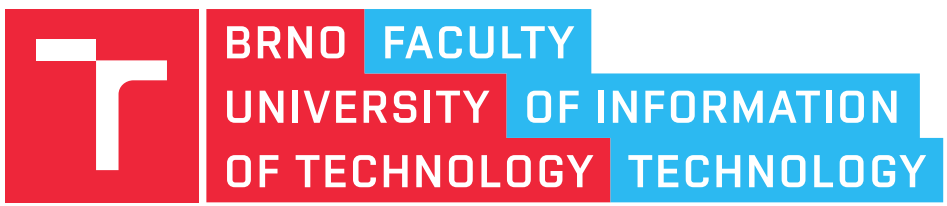
ICS-TestBed

Peter Grofčík

Brno University of Technology, Faculty of Information Technology
Božetěchova 1/2. 612 66 Brno - Královo Pole
xgrofrc00@stud.fit.vutbr.cz



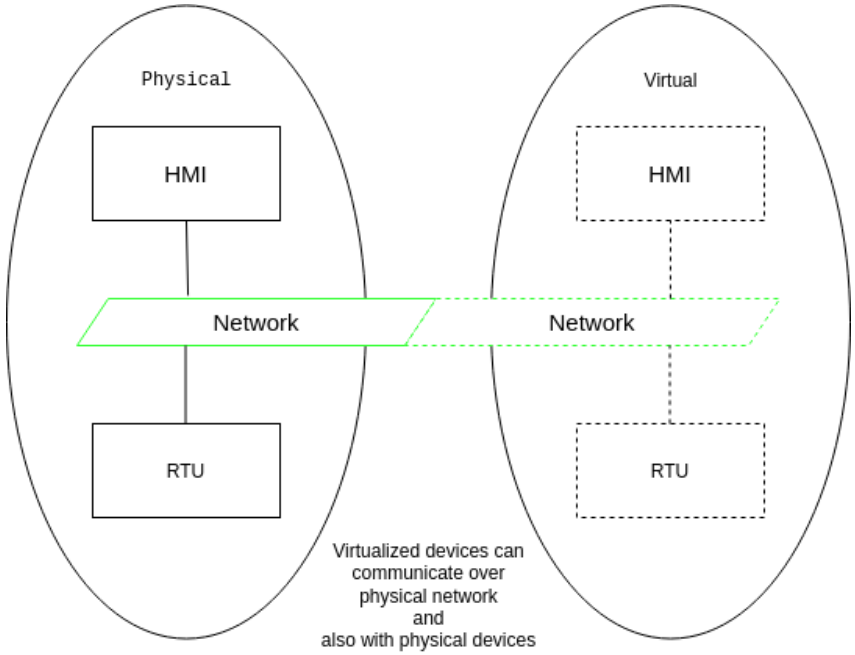
November 9, 2020

- Framework[1] itself requires JAVA 8 with maven.
- Simple cloning is not enough, because there are some missing dependencies (not up to date) in *pom.xml* file.
- Compilation needs to be processed without the execution of internal tests (-DskipTests).
- Firewall problems might occur during execution. At least port 2404 (default) needs to be enabled.

[1] https://github.com/PMaynard/ICS-TestBed-Framework

TestBed - Two types of devices (mainly)

- HMI (Human-machine interface) – periodically controlling other hosts.
- RTU (Remote Terminal Unit) – represents node for connection between sensors and HMI.



Execution

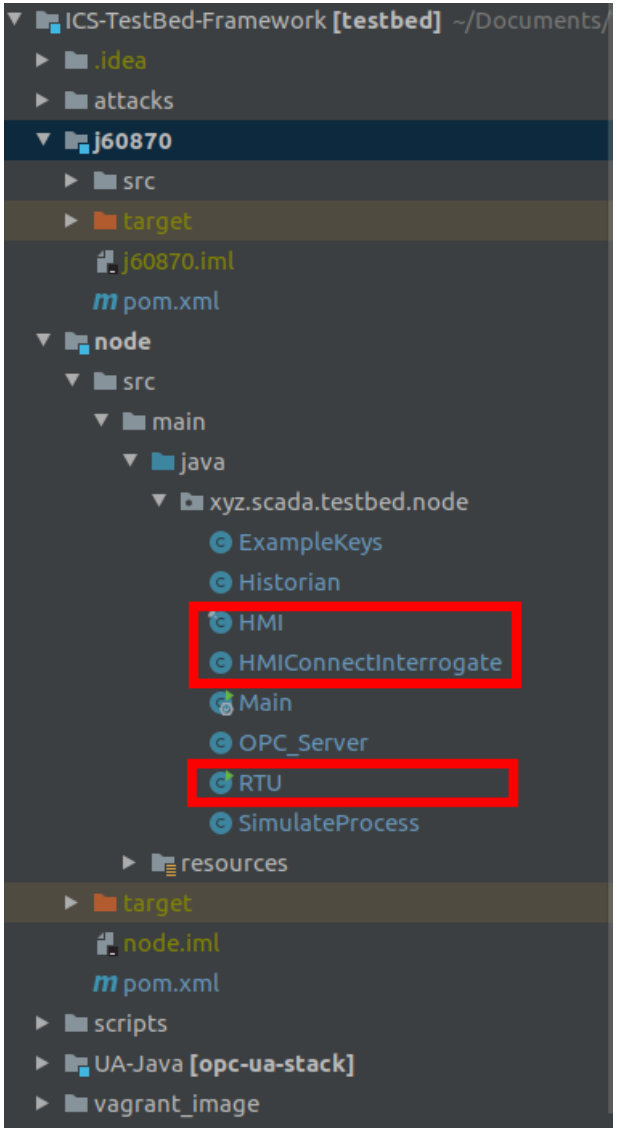
- Compilation creates *jar* archive, which will open a shell command line after execution.
- Any desired IP information is to be set in this stage before the execution of a single device.
- Default settings work for local-host communication.

```
HMI
        hmi: Start HMI.
        hmi-common-address: Set IEC104 Common Address to query.
        hmi-iec104port: Set IEC105 port.
        hmi-interval: Set interval query (MS).
        hmi-show: Show current HMI configuration.
        remote-hosts: Set hosts to connect to.
```

```
RTU
        enable-opcua: Enable OPC-UA
        rtu: Start RTU.
        rtu-iec104port: Set IEC104 Port.
        rtu-listen: Set listen interface.
```

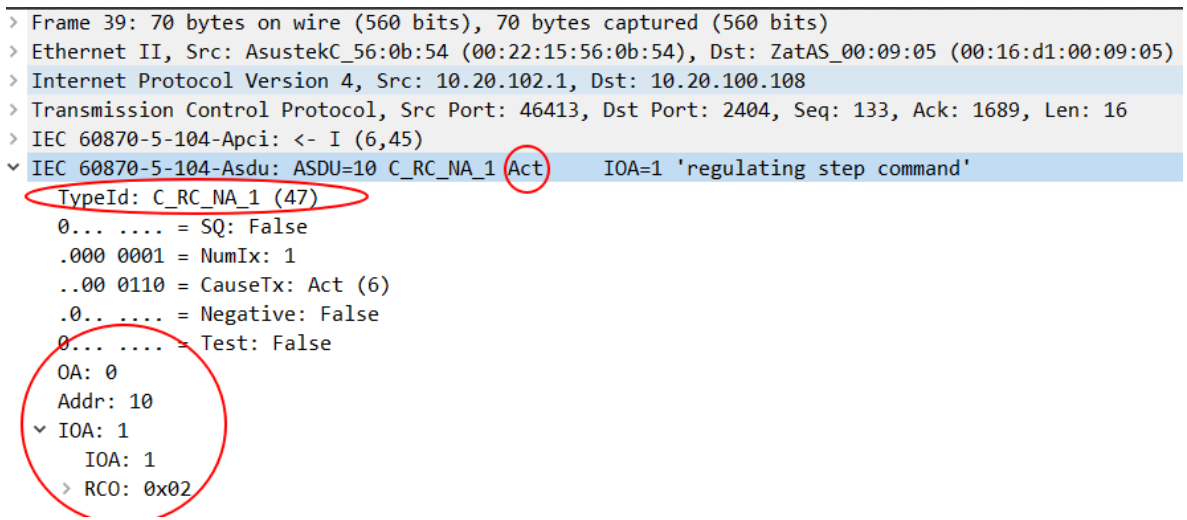
Building virtual devices I.

- The behavior of each device needs to be hard-coded in the corresponding class (does not require any IP information)



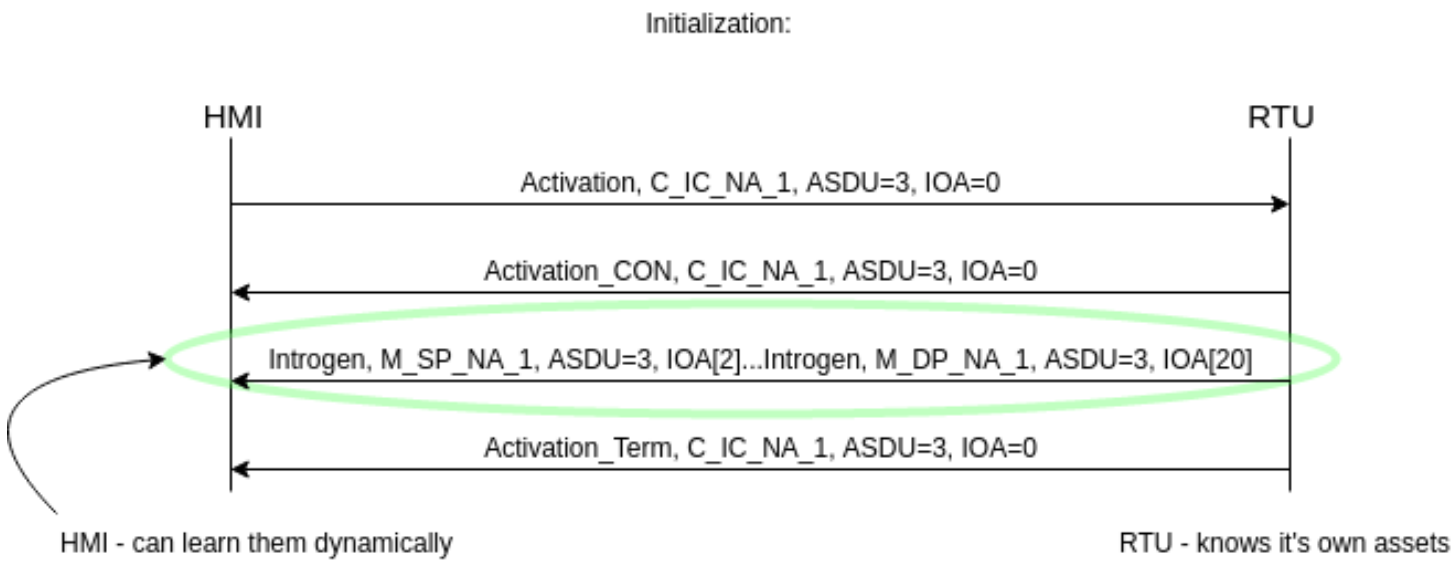
Building standard behavior from captured traffic requires important values such as:

- Asdu
- Type of command
- TypeID of command
- OA – Address of originate
- IOA – Object(s) of interest for command



Building virtual devices III.

- From data collected this way it's possible to generate behavior for each device to be virtualized.
- RTU device is represented by a finite automaton, that can provide information about itss assets any time it's asked for.
- HMI device is an initiator of communication and can learn about assets that can be provided by RTU dynamically during INIT state (does not need to be hard-coded as for RTU).



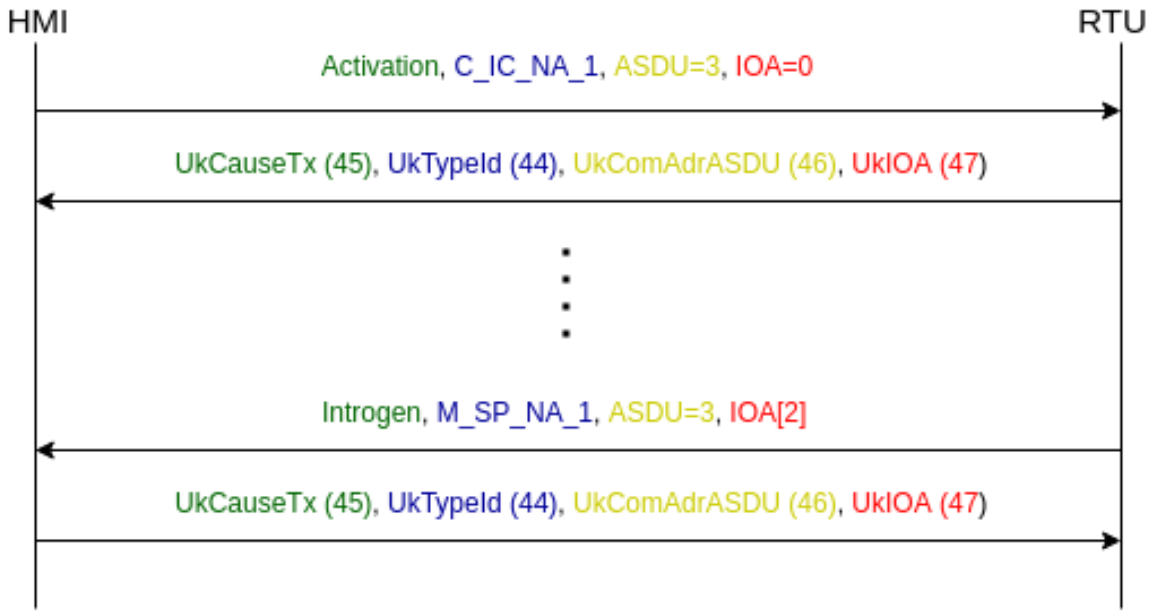
Building virtual devices IV.

- As initiator HMI is responsible for cyclic communication, so that needs to be hard-coded on the HMI part.
- It means that HMI consists of sequence(s) of Activation commands based on captured traffic.
- HMI can be an initiator for more RTUs of multiple types (meaning that HMI can handle more parallel sequences).

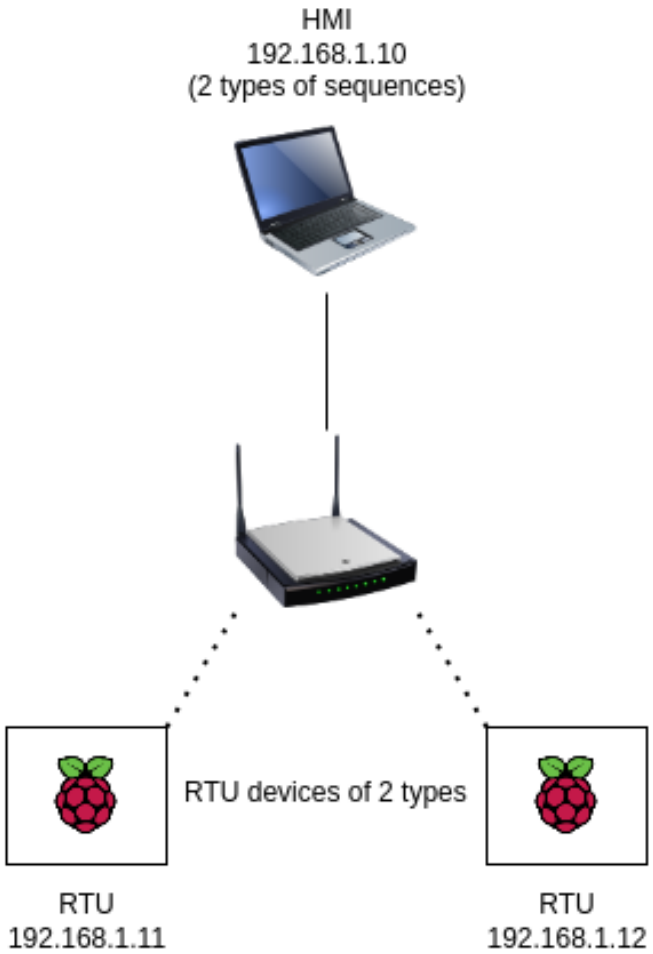
```
41 10.029175444  192.168.1.10    192.168.1.12    104asdu   82 <- I (2,31) ASDU=10 C_SC_NA_1 Act     IOA=2
47 10.033351119  192.168.1.10    192.168.1.12    104apci   72 <- S (34)
51 20.029494077  192.168.1.10    192.168.1.12    104asdu   82 <- I (3,34) ASDU=10 C_SC_NA_1 Act     IOA=13
57 20.039124781  192.168.1.10    192.168.1.12    104apci   72 <- S (37)
59 30.031989301  192.168.1.10    192.168.1.12    104asdu   82 <- I (4,37) ASDU=10 C_DC_NA_1 Act     IOA=1
70 30.081254621  192.168.1.10    192.168.1.12    104apci   72 <- S (40)
73 40.032341905  192.168.1.10    192.168.1.12    104asdu   82 <- I (5,40) ASDU=10 C_DC_NA_1 Act     IOA=14
```

Building virtual devices V. Processing of unknown data

- Framework itself did not consist of any useful functions for processing data that is not corresponding in the scenario.
- From IEC104 specifications: Each device should be able to check incoming messages for unknown data obtained.
- My addition needed to be made to *Connection* class so that both types of devices can check its incoming message and react with a correct negative response if needed.



Simulated topology



	Source no.1	Source no.2	Output (HMI cap)
Packets	105 (2 minutes)	174 (3 minutes)	19542 (2 days)
IEC104	57%	62.5%	50.5%

Next step

Start with emulation of attacks

- Packet drop-outsown
- Man-in-the-middle
- DDoS
- Scanning

Thank You For Your Attention !