NTNU – Trondheim
Norwegian University of
Science and Technology

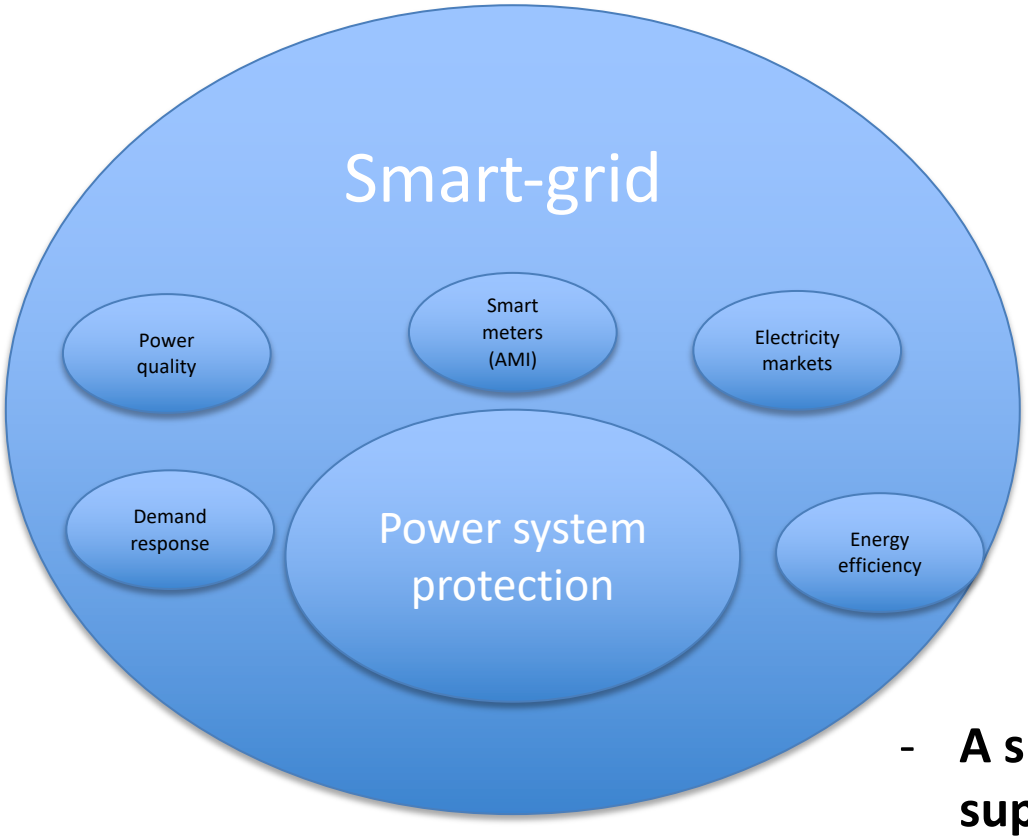
Real-Time Hardware-in-the-loop Simulation for Protection Applications

- from a smart-grid perspective

Charles M Adrah, Ph.D. Candidate
Department of Information Security and Communication Technology, NTNU
29th October, 2019

EriGrid and Cineldi: Workshop on co-simulation and hardware-in-the-loop

Motivation



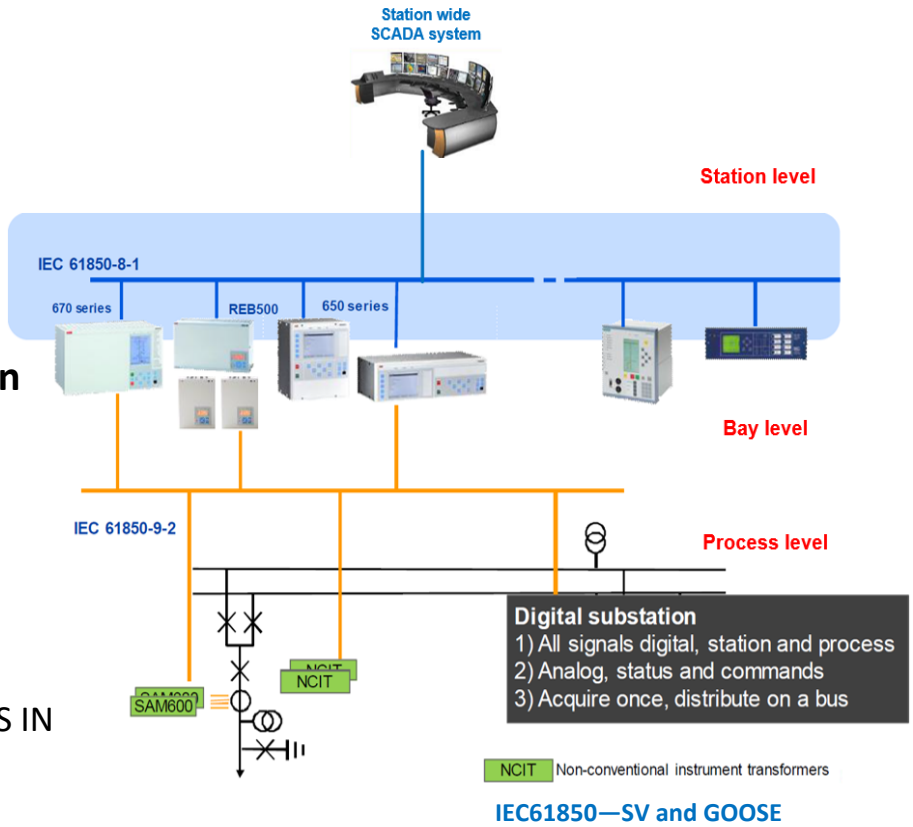
- **A smart-grid requires an ICT support system**
- **Protection applications and algorithms developed need to consider this.**

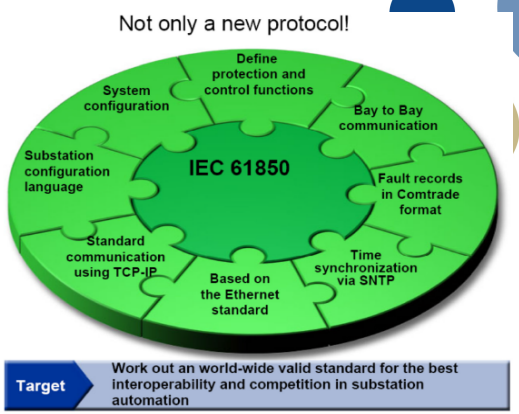
What is digital substation?

Power-system automation

Substation automation

IEC 61850 –
"COMMUNICATION NETWORKS AND SYSTEMS IN SUBSTATIONS"



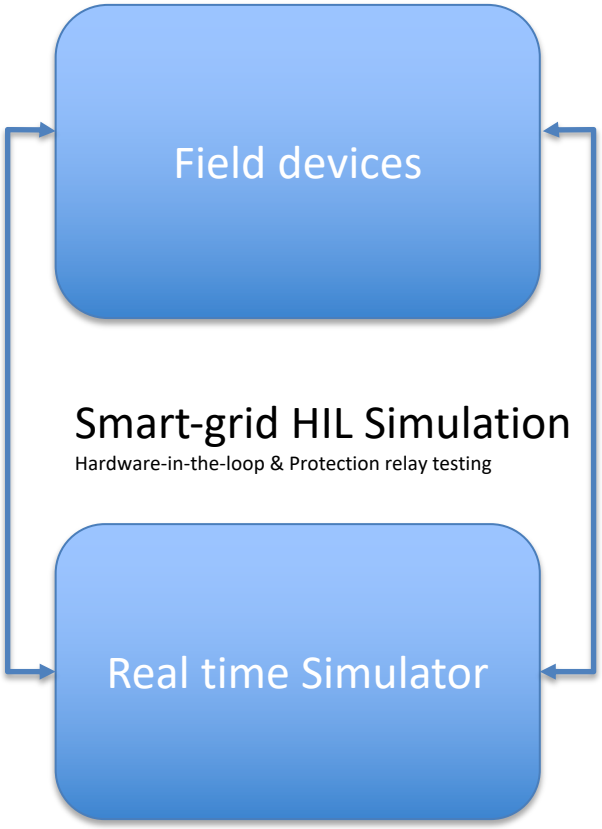


Pros and **cons** of IEC 61850 based digital substation:

- *Standard Ethernet-based communications systems*
- *No "spaghetti" mess behind*
- *Easy system expansion*
- *Increased system reliability*
- *Interoperability*
- *Increased safety for personnel*

- *More expensive implementation*
- *Training required*
- *Modifications from traditional automation system*
- *Cyber security*

Role of Communication
In smart-grid protection



Challenges;

- Deploying power system protection applications with communication technology e.g. wide area networks
- Capture network parameters that affect performance of protection traffic in intra/inter substation i.e. delay, packet loss, jitter
- OPAL-RT simulator limited in the ability to simulate communication network impairments

Need to factor communication effects in power system protection applications

Role of Communication
In smart-grid protection

Proposed Approach

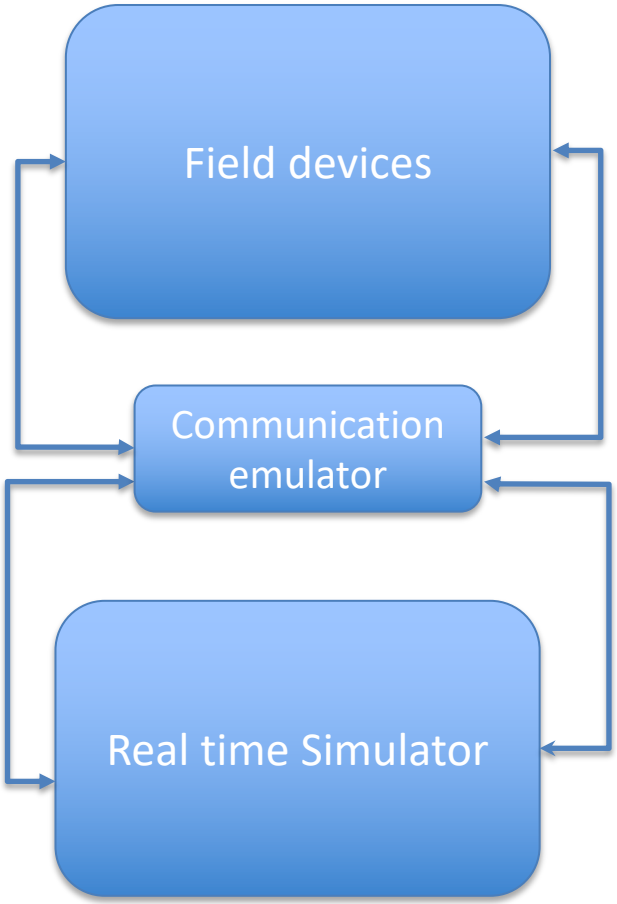
Communication network emulation;

a solution to reflect the characteristics of the ICT system and its interface with the physical power process.

Click Router as a Network Emulator Design Tool

Outline

- Network emulator design
- Architecture & Features
- Integration of emulator and HIL Test bed
- Composing new elements
- Case Studies, Results & Evaluations



Network Emulator Design & Applications

Design

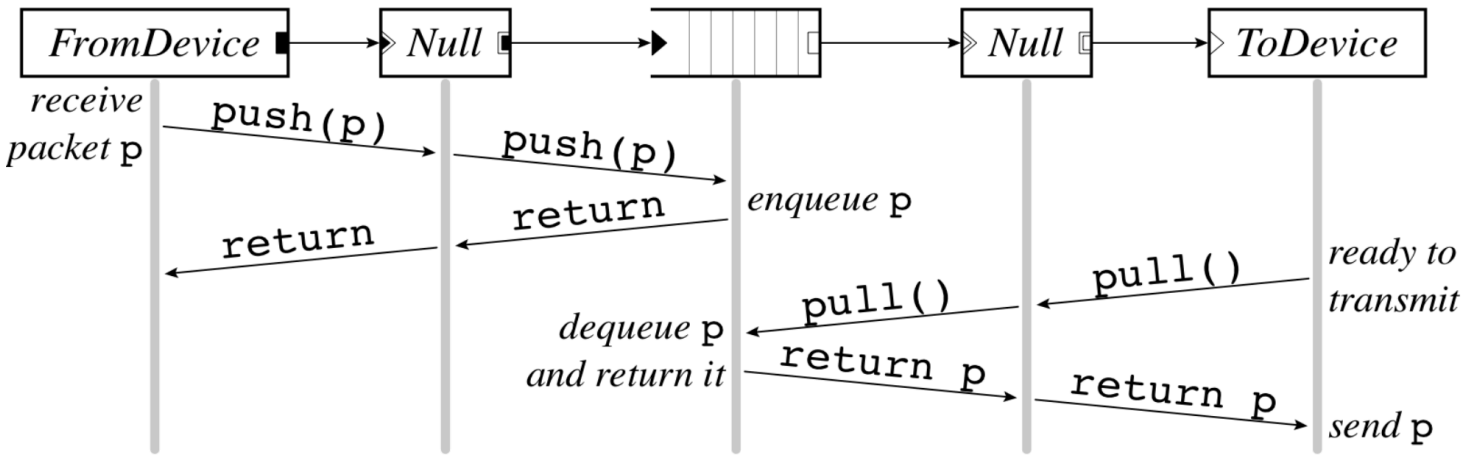
- Emulator enable us to change delay, jitter and packet corruption as a function of time.
- Control communication properties between multiple source relays and destination relays
- Impairing specific subsets of the network traffic
- Vary the properties of communication parameters in real time using - using handlers
- Bandwidth restriction
- Emulate different queueing schemes and traffic priorities

Motivation for choice of tool: Click Modular Router

Click Router is a software framework for building flexible and configurable routers

- Flexibility
  - Adding new features to enable experimentation
- Openness
  - Allow users to build and extend
- Modularity
  - Simplify the composition of existing features & addition of new features

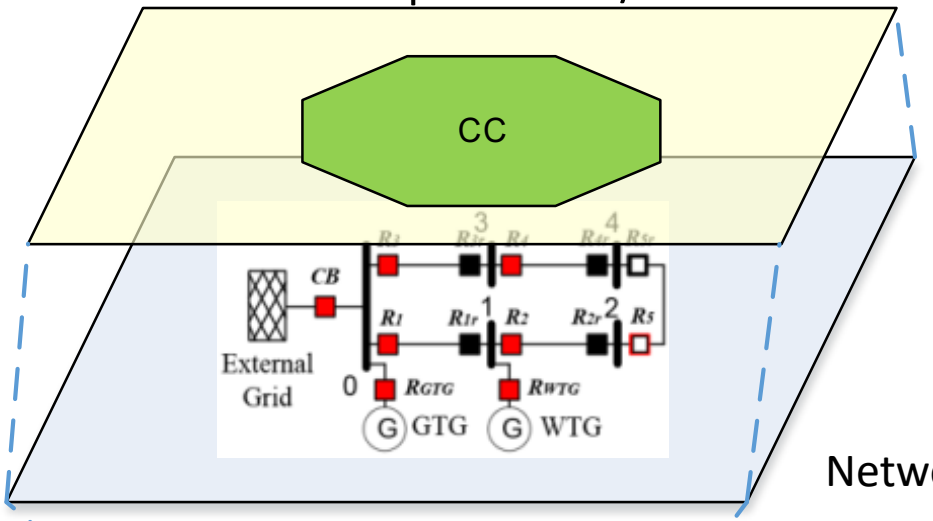
Router as a Graph of Elements & Push and Pull Connections



- Push connection
  - Source pushes packets downstream
  - Triggered by event, such as packet arrival
  - Denoted by filled square or triangle
- Pull connection
  - Destination pulls packets from upstream
  - Packet transmission or scheduling
  - Denoted by empty square or triangle
- Agnostic connection
  - Becomes push or pull depending on peer
  - Denoted by double outline

Integration of Emulator and HIL : Setup

Test power system network with CC in Opal RT-Lab/Matlab



Practical protective relays

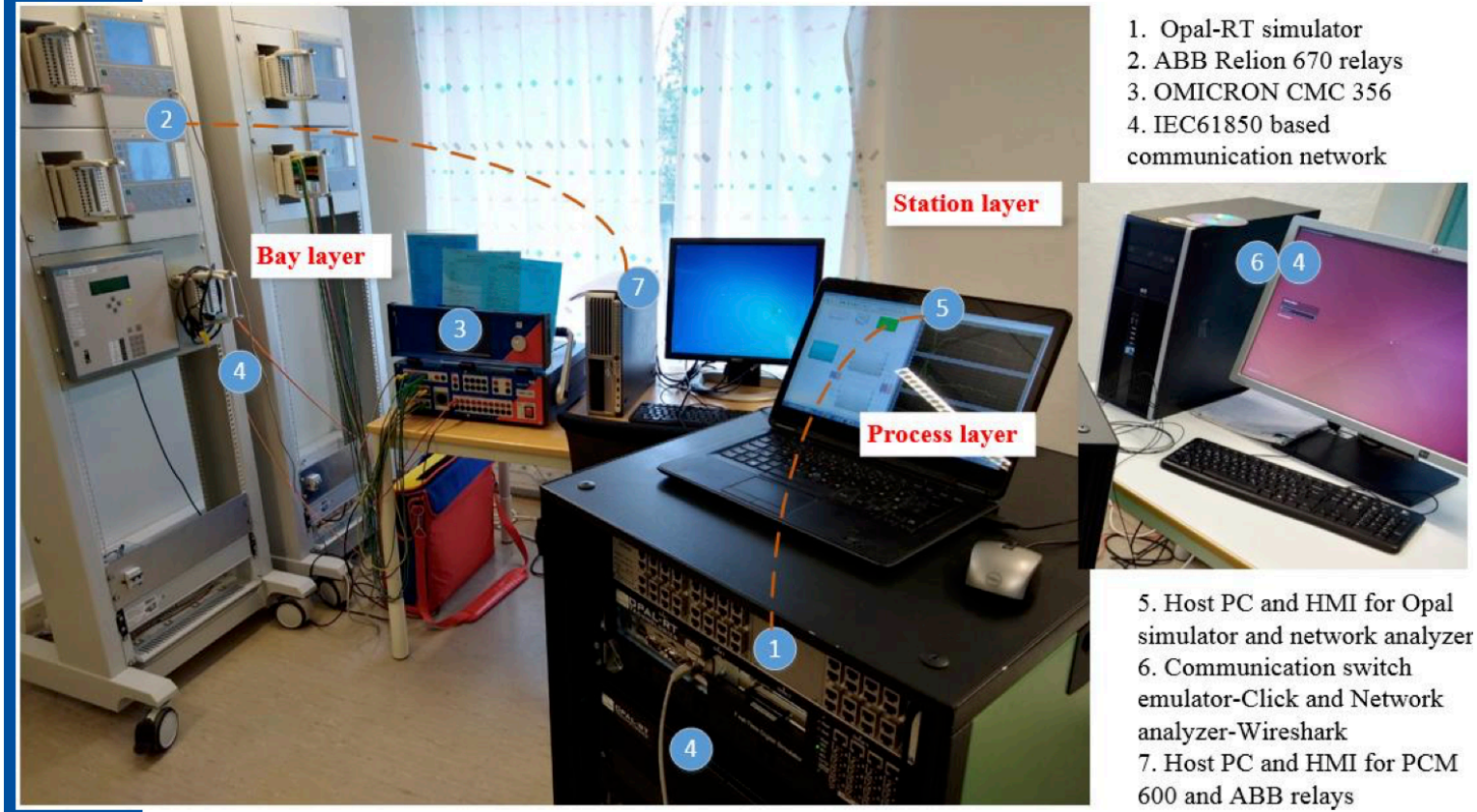
Network Emulator

SV,
GOOSE
-status of CBs and SGs to CC
-SG switch order from CC

OPAL simulator OP5600

SV,
GOOSE
-status of CBs and SGs from relay
-SG switch order to relay

Real time Hardware in the loop (HIL) protection lab at NTNU

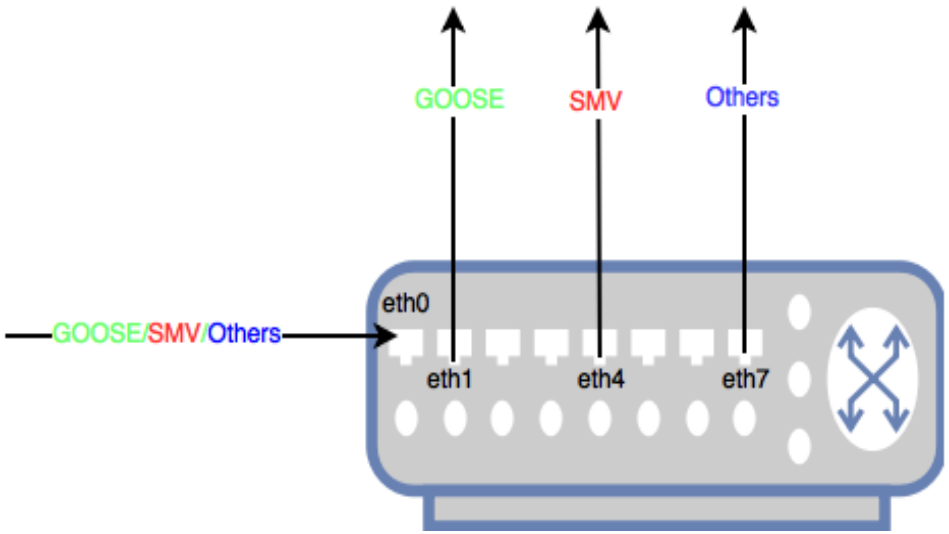


1. Opal-RT simulator
2. ABB Relion 670 relays
3. OMICRON CMC 356
4. IEC61850 based communication network
5. Host PC and HMI for Opal simulator and network analyzer
6. Communication switch emulator-Click and Network analyzer-Wireshark
7. Host PC and HMI for PCM 600 and ABB relays

Composing new elements: Emulator with IEC 61850 Capabilities

– Classifying IEC 61850 GOOSE & SMV packets

```
require(clicklocal);


        FromDevice(eth1)-> ic :: IECClassifier(GOOSE, SMV, -)

        ic[0]-> Queue(10000)-> ToDevice(eth0);

        ic[1]-> Discard;

        ic[2]-> Discard;

ClickyInfo(STYLE @import test-clicky.ccss);
```



Composing new elements : GOOSE Classifier

```
require(clicklocal);


        FromDevice(eth0)

            -> ic :: IECClassifier(GOOSE, SMV, -)

       ic[0]-> Strip(14)

            -> CheckGOOSEHeader()

            -> gc :: GOOSEClassifier(GOOSE_TRIP1, GOOSE_OV2PTOV, -)

       gc[0]-> ToDevice(eth1);

       gc[1]-> ToDevice(eth2);

      gc[2] -> Discard;

       ic[1]->Discard;
       ic[2]->Discard;
```

```
CLICK_DECLS


GOOSEClassifier::GOOSEClassifier() { };
GOOSEClassifier::~GOOSEClassifier() { };


int
GOOSEClassifier::initialize(ErrorHandler *errh)
{

    return 0;
}


void GOOSEClassifier::push(int,Packet *p) {
      unsigned  _textlength = TEXTLENGTH;

      int out_port = 2; // Default output for others

         const unsigned char *data = (p->data()+_textlength);
         unsigned  goCBRefLength=0;
         //  sa<<((*(data-1)) & 0xff);
         if((*data & 0xff) == goCBRefTag){
           data++;
          goCBRefLength = ((*data) & 0xff);
          //  unsigned char goCBRefData[goCBRefLength];
          /* for (unsigned i = 0; i < goCBRefLength ; i++){
             data++;
           goCBRefData[i] = { (*data)};
           }*/
            }

        const unsigned char *data2 = (p->data()+_textlength+2+goCBRefLength);
        unsigned talLength=0;
         if((*data2 & 0xff) == talTag){
           data2++;
           talLength = ((*data2) & 0xff);
         }

       const unsigned char *data3 = (p->data()+_textlength+4+goCBRefLength+talLength);
       unsigned datsetLength=0;
         if((*data3 & 0xff) == datsetTag){
           data3++;
           datsetLength = ((*data3) & 0xff);
         }

       const unsigned char *data4 = (p->data()+_textlength+6+goCBRefLength+talLength+datsetLength);
```

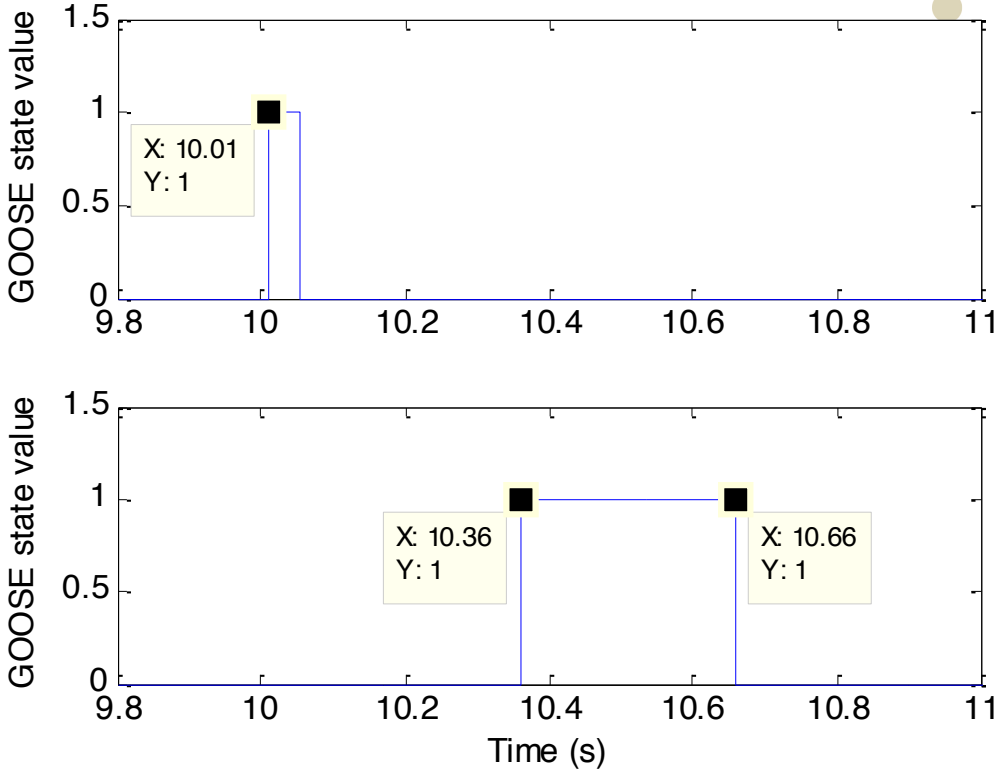
Composing new elements : Random Delay Element

- An element that applies random delay time on packets passing the router
- Assumes uniform distribution

```
require(clicklocal);

     FromDump(/home/charles/Documents/tracedumps/udpsample1.pcap, STOP true)
    -> Strip(14)-> CheckIPHeader()

    -> ip :: IPClassifier(udp,tcp,icmp,-)
    ip[0] -> tp :: Tee(2)
    tp[0] ->  q1::Queue(1000);


    tp[1] -> Queue(1000) -> RandomDelay(1,5)-> q1;

       q1 -> Unstrip(14) -> ToDump(/home/charles/Documents/tracedumps/drop.pcap);
    ip[1] -> Discard;
    ip[2] -> Discard;
    ip[3] -> Discard;

ClickyInfo(STYLE @import test-clicky.ccss);
```

HIL-POTT Applicatiion

- **Case Study : HIL POTT application**
  - Emulator introduces delay of 0.35 seconds on GOOSE packets
  - ABB Relion 670 relay GOOSE publisher
  - Relay in OPAL-RT GOOSE subscriber
- **Zone 2 backup protection to clear fault, 0.3 seconds**

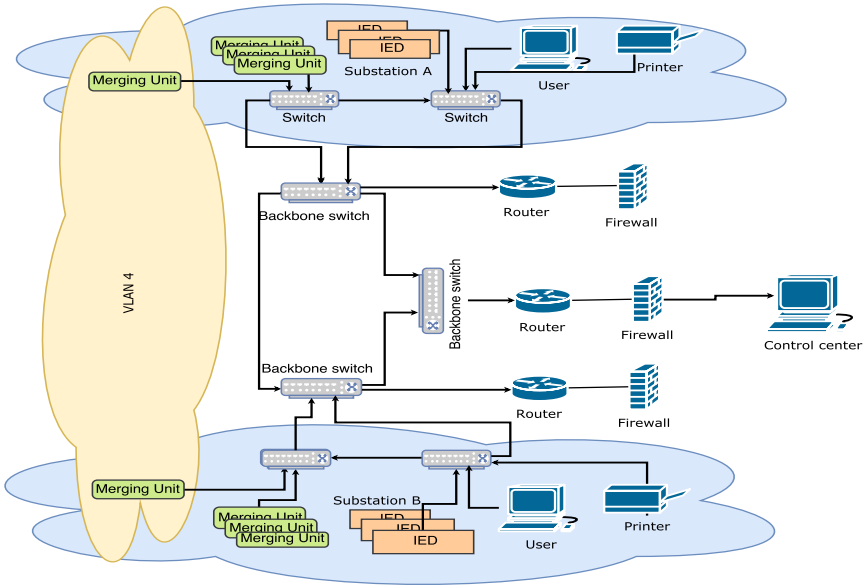


GOOSE data signals recorded by OPAL-RT

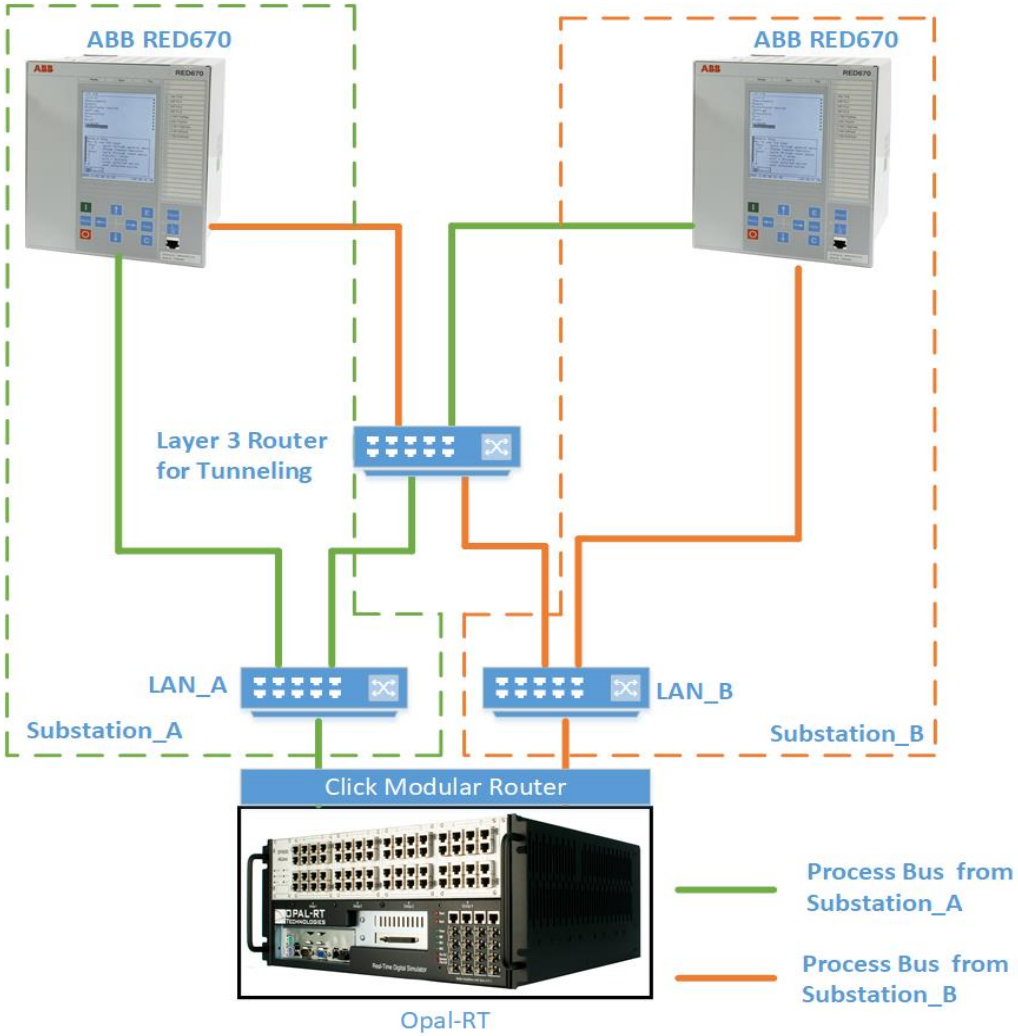
- Normal operation: detected fault cleared by the permissive trip signal
- Abnormal operation(delay introduced); POTT failed to clear fault in time
  - Backup protection was triggered at 0.3s to clear fault

Sample Values in Wide Area – VLAN Architecture for state estimation

- VLAN; a distinct Ethernet LAN sharing the same physical medium with other VLANs
- Method involves assigning a unique VLAN tag to SV data to routed outside the substation
- Enables connecting process busses from different substation
- Investigating the performance of the process bus network, i.e. ***characterizing latency and jitter for situations such as heavily loaded networks.***
- Feasible in a dedicated and isolated network for the utility - limitations



Routable SV in Wide Area for State Estimation; Laboratory Set up



Conclusion

- Emulator can be used to ***change communication properties of delay, jitter, packet loss, packet corruption and bandwidth*** for WAN power system applications
- Smart-grid HIL simulation approach can be used to ***estimate communication network constraints*** to be considered in protection application settings to allow for transient communication QoS parameters.
  - Investigate effects of network congestion and packet loss on different protection algorithms
  - Model time delays and data loss to study effects of relay performance compromised by distributed generation and non-zero fault impedances.

References

References;

- *Adrah, C. M., Kure, O., Liu, Z., & Hoidalen, H. K. (2017). Communication network modeling for real-time HIL power system protection test bench. In Proceedings - 2017 IEEE PES-IAS PowerAfrica Conference: Harnessing Energy, Information and Communications Technology (ICT) for Affordable Electrification of Africa, PowerAfrica 2017. https://doi.org/10.1109/PowerAfrica.2017.7991240*
- *Pandakov, K., Adrah, C. M., Liu, Z., Høidalen, H. K., & Kure, Ø. (2018). Hardware-in-the-loop testing of impedance protection with compensation of fault impedance and DG infeed current. The Journal of Engineering, 2018(15), 1018–1022. https://doi.org/10.1049/joe.2018.0189*
- *Adrah, C. M., Kure, O., Yellajosula, J. R. A. K., Paudyal, S., & Mork, B. (2018). A Methodology to Implement and Investigate Performance of Sampled Values for Wide-Area Protection. 2018 2nd International Conference on Smart Grid and Smart Cities (ICSGSC), 84–90. https://doi.org/10.1109/ICSGSC.2018.8541290*
- *Pandakov, K., Adrah, C. M., Hoidalen, H. K., & Kure, O. (2019). Experimental validation of a new impedance based protection for networks with distributed generation using co-simulation test platform. IEEE Transactions on Power Delivery, 1–1. https://doi.org/10.1109/tpwrd.2019.2935834*

Thanks for the attention