European Commission

WiSHFUL

Wireless Software and Hardware platforms for Flexible and Unified radio and network controL

Year 2 Demonstration of Showcases

imec TRINITY COLLEGE DUBLIN cnit TU berlin nCENTRIC RUTGERS UFRJ SEOUL NATIONAL UNIVERSITY

This project has received funding from the European Union's H2020 Programme under grant agreement no 645274.

Table of Contents

COEXISTENCE OF IEEE 802.15.4E TSCH WITH IEEE 802.11

GOALS

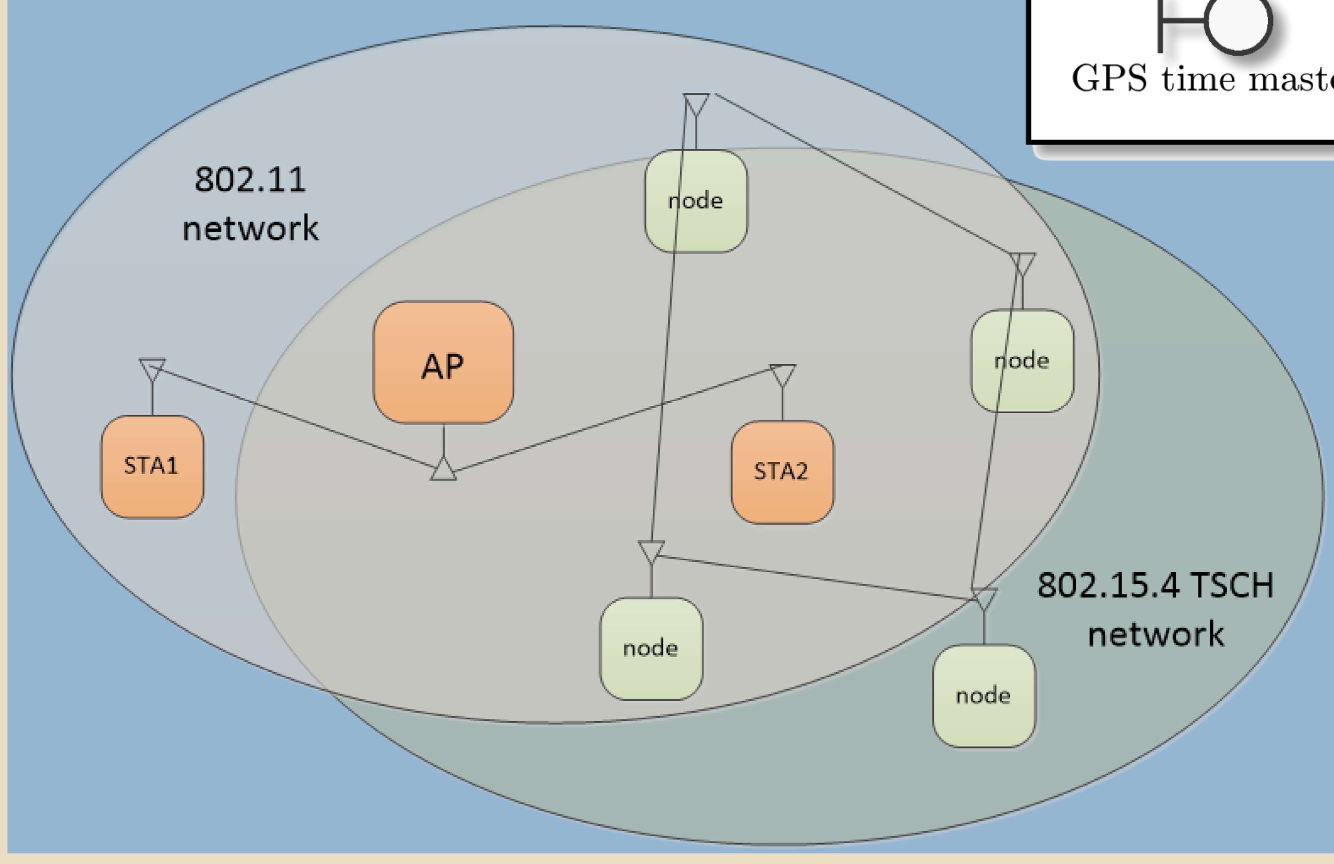
- IEEE 802.11 Wi-Fi
  - High throughput, Best effort
- IEEE 802.15.4e TSCH
  - Low latency, Low volume, High priority
- Wi-Fi will keep the channel free for the time of TSCH transmissions
- Wi-Fi regular (CSMA based) transmission in during rest of the time

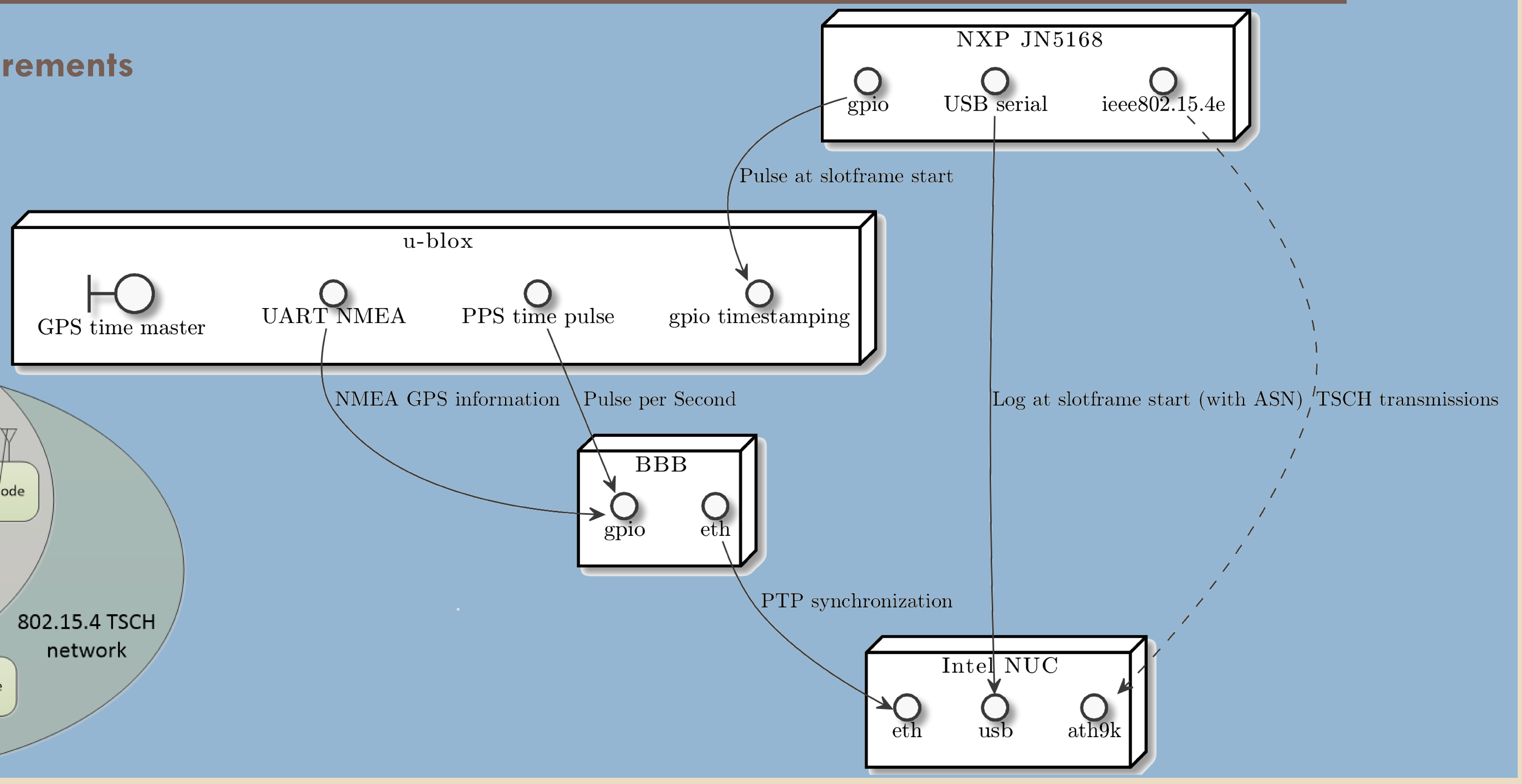
CHALLENGES

- TSCH slots might be scheduled to fulfill hard timing deadlines
- Unicast transmissions in TSCH do not incorporate Clear channel assessment (CCA)
- Wi-Fi network needs to get time reference signals from running TSCH network
  - Over the air
  - COTS Wi-Fi devices

EXPERIMENT SETUP

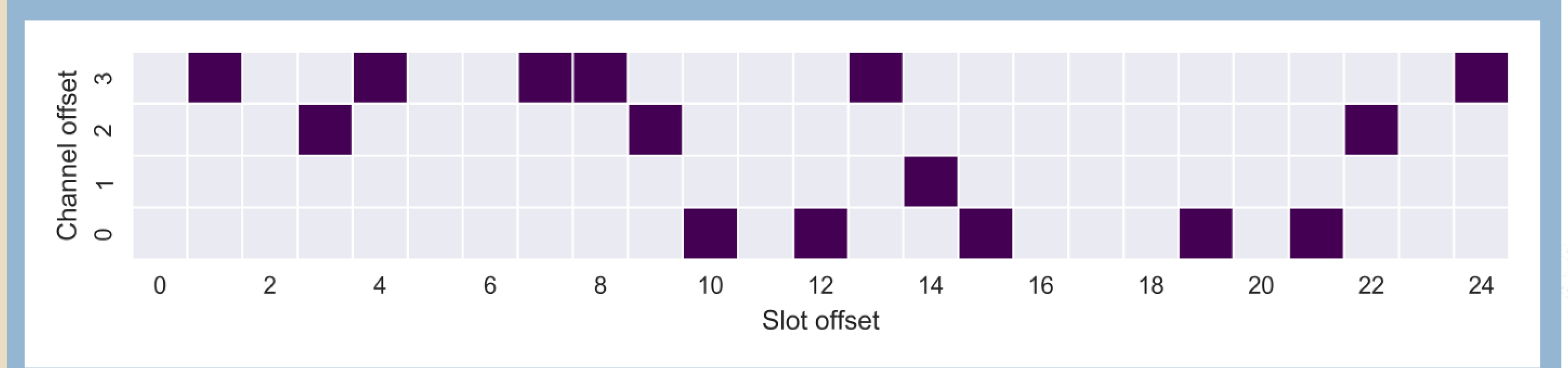
- Synchronization quality measurements
- Out-of-band GPS timing master

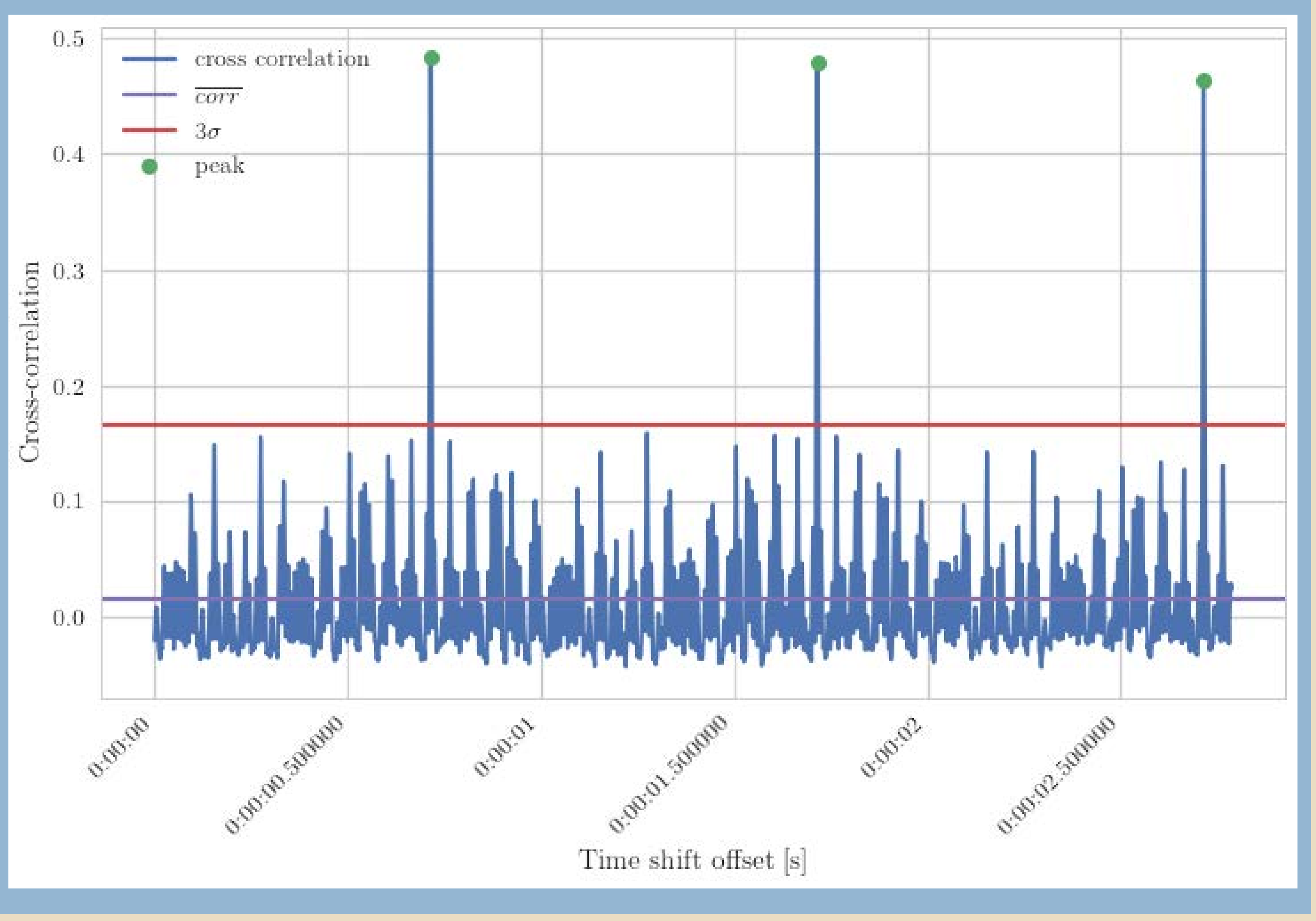




RESULTS

- Given TSCH schedule





- Achieved synchronization signal
- Timing measurements
  - Feasibility of the solution
  - Accuracy of the synchronization
  - Jitter in synchronization

CONCLUSIONS

- Over-the-air synchronization service for WiSHFUL framework
- Enabler for advanced cooperation between networks

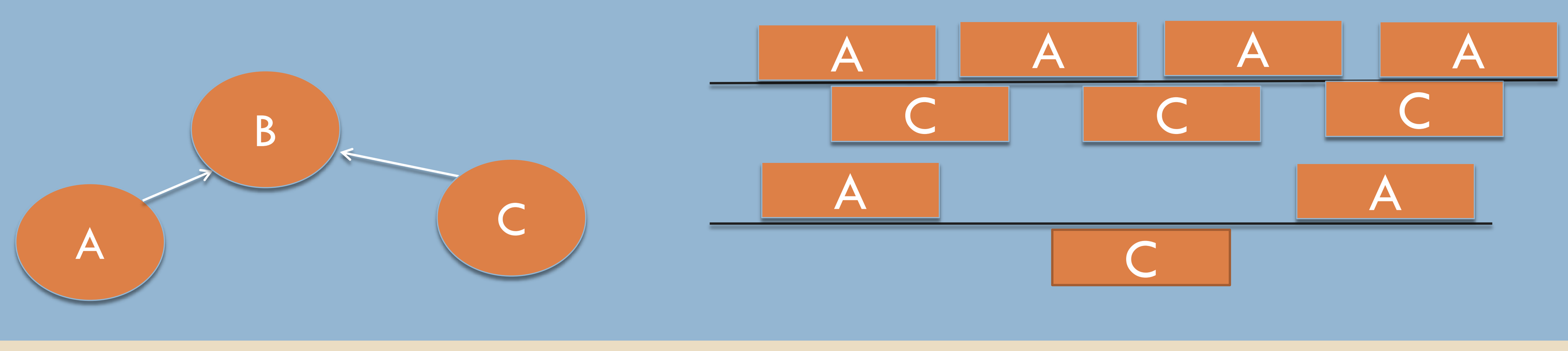
POST MORTEM

- Need for external and independent time master for timing measurements
- Usage of existing WiSHFUL capabilities to create new synchronization service

MULTIHOP LOAD-AWARE MAC ADAPTIONS

GOALS

- **Improve random access performance in multi-hop networks (reducing hidden nodes and starvation)**
  - **How? Limiting channel access attempts performed by each node to leave space to the other ones**



CHALLENGES

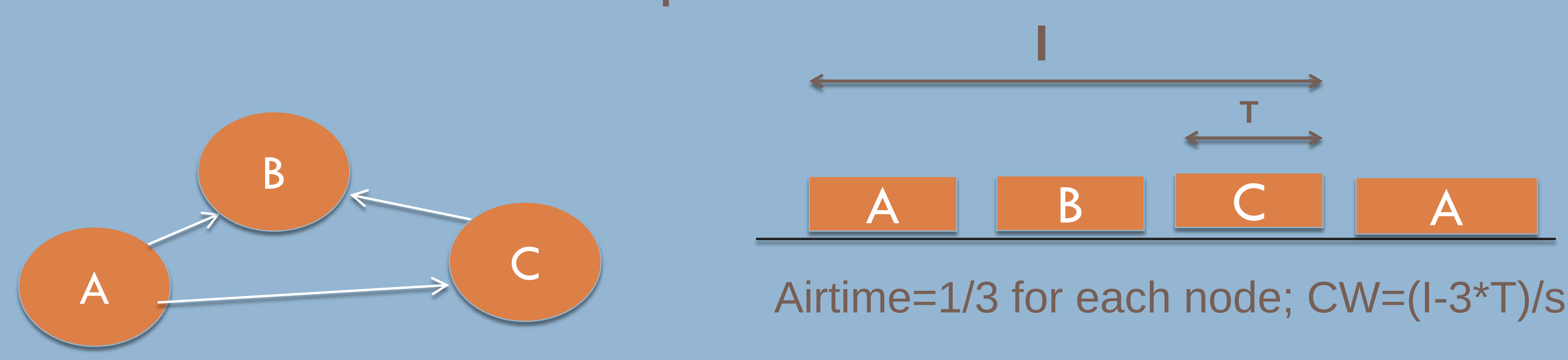
- **Make a distributed auction of resources among the nodes**
  - **Receivers offer channel 'airtimes', transmitters claim channel 'airtimes'**
  - **Resources are equally shared among competitors, no node gets more than its demand**
- **Map a desired airtime allocation to a contention window tuning**
  - **Airtimes are only probabilistically guaranteed**

Auction Mechanism Principles : REACT

- Competing nodes involve 1-hop and 2-hop neighbors, to not overload receivers!

Channel at node B: if node A takes 1/2 of aritime, only 1/2 is available for both node B and node C transmissions!

- If N nodes are competing on the channel, no node can take more than 1/N!
- If one node needs less, equally share the spare capacity among the others.
  - CW tuning for equalizing the idle time between successive channel attempts



Airtime=1/3 for each node; CW=(I-3*T)/s

DEMO SETUP

- Two testbeds with non-fully connected topologies
  - Simple topology, in portable testbed for intuitive results
  - More complex topologies in wilab.t

Storyline: traffic flows are activated in the network under legacy 802.11 DCF

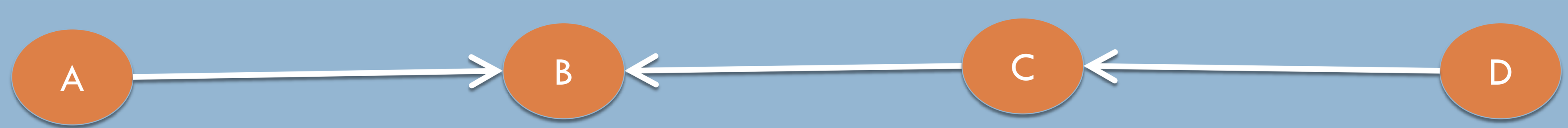
- *from hidden nodes to a common receiver*
- *from consecutive nodes in a path*

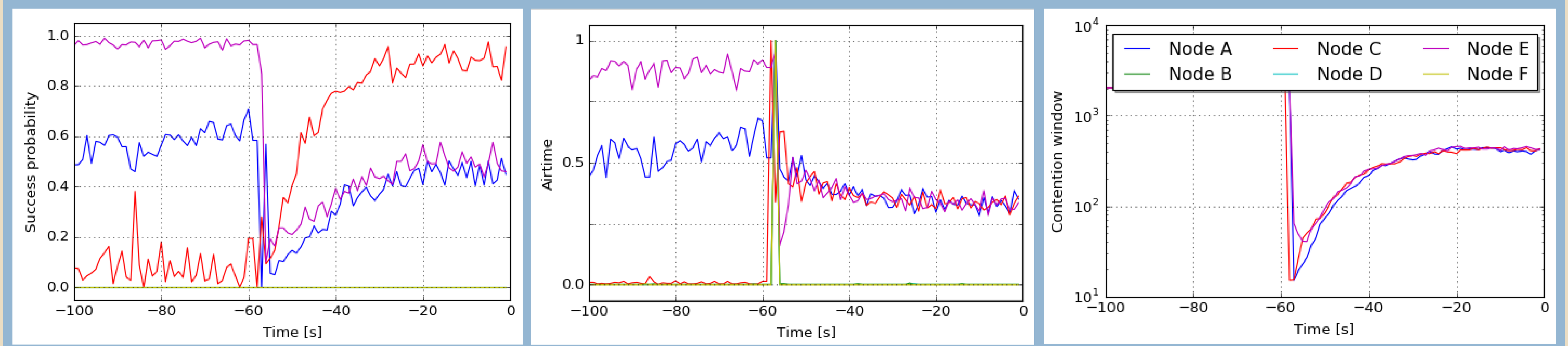
Performance are very critical, as expected;

The airtime negotiation is activated and fair transmission opportunities are achieved

- *Fair allocations generally imply also better throughput performance*

RESULTS





DCF REACT DCF REACT DCF REACT

UPI USAGE

UPI_HC	UPI_R
Send the local CP from and report statistics to controller. **start_local_control_program();** **send();** **recv();**	Frame forging for supporting network control protocol **inject_frame()** **sniff_layer2_traffic()**
	To set protocol parameters: **set_parameters**(interface, UPI_R.CW_MIN, UPI_R.CW_MAX)
	To get nodes statistics: **get_measurements**(interface, RTS, CTS, DATA , ACK, CHANNEL BUSY]

POST MORTEM

- What we want demonstrate to other experimenters?
  - How to implement network coordination mechanisms based on distributed protocols
    - Forging of customized frames
    - How to map airtime to contention window values
  - Separation of control protocols and access schemes
    - Different logics for airtime allocations can be easily implemented, e.g. per-path allocations

TAISC INTEGRATION TO SDR AND ZOLERTIA RE-MOTE

GOALS

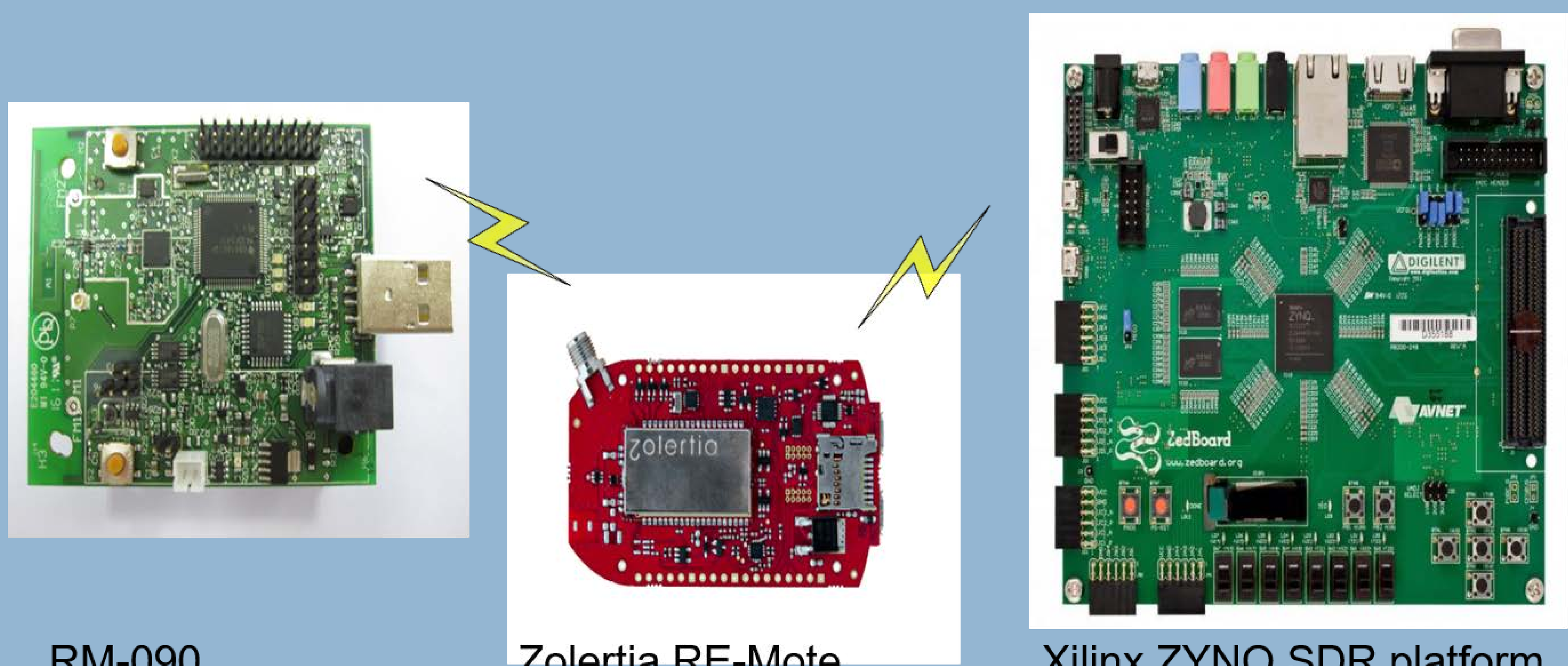
- Port TAISC to SDR and Zolertia RE-Mote
- Take advantage of the portability of the implemented MAC protocols in all platforms
- Minimize overhead of maintaining MAC implementations across different platforms

CHALLENGES

- Hide heterogeneous nature of the devices from the MAC implementer using TAISC
- Is MAC implementations based on TAISC portable between heterogeneous devices?
- Integrating various MAC implementations using TAISC to IPv6/RPL/6LowPan based stack.

EXPERIMENT SETUP

- Setting up a network comprised by different type of IEEE 802.15.4 based devices.
- TAISC MAC implementation is integrated in all platforms



RM-090
(MSP430 based)

Zolertia RE-Mote
(ARM Cortex M3 based)

Xilinx ZYNQ SDR platform
(ARM Cortex A9 based)

RESULTS

- Successful communication using UDP based CoAP between all devices proving there is reliable communication
- Able to reach all different devices through IPv6 ping.

```
19824 12316.325861000   fd00::c00a:0:0:f1   fd00::200:0:0:2   CoAP          84 ACK, MID:39936, 2.05 Content
19825 12316.326126000                                         IEEE 802.15.4  5 Ack
19826 12318.312701000   ::200:0:0:2         ::c00a:0:0:f1     CoAP          70 CON, MID:39937, POST, /toggle (text/plain)
19827 12318.313198000                                         IEEE 802.15.4  5 Ack
19828 12318.316030000   ::200:0:0:2         ::c00a:0:0:f1     CoAP          71 CON, MID:39937, POST, /toggle (text/plain)
19829 12318.316349000                                         IEEE 802.15.4  5 Ack
19830 12318.322341000   fd00::c00a:0:0:f1   fd00::200:0:0:2   CoAP          84 ACK, MID:39937, 2.05 Content
19831 12318.326087000   fd00::c00a:0:0:f1   fd00::200:0:0:2   CoAP          84 ACK, MID:39937, 2.05 Content
19832 12318.326653000                                         IEEE 802.15.4  5 Ack

Frame 19830: 84 bytes on wire (672 bits), 84 bytes captured (672 bits) on interface 0
IEEE 802.15.4 Data, Dst: TexasIns_00:06:15:a0:a2, Src: c2:0a:00:00:00:0000:f1
6LoWPAN
Internet Protocol Version 6, Src: fd00::c00a:0:0:f1 (fd00::c00a:0:0:f1), Dst: fd00::200:0:0:2 (fd00::200:0:0:2)
User Datagram Protocol, Src Port: 5683 (5683), Dst Port: 5683 (5683)
Constrained Application Protocol, Acknowledgement, 2.05 Content, MID:39937
  01.. .... = Version: 1
  ..10 .... = Type: Acknowledgement (2)
  .... 0000 = Token Length: 0
  Code: 2.05 Content (69)
  Message ID: 39937
```

```
[spilios@localhost contiki]$ ping6 fd00::c00a:0:0:f2
PING fd00::c00a:0:0:f2(fd00::c00a:0:0:f2) 56 data bytes
64 bytes from fd00::c00a:0:0:f2: icmp_seq=1 ttl=63 time=1561 ms
64 bytes from fd00::c00a:0:0:f2: icmp_seq=3 ttl=63 time=31.2 ms
64 bytes from fd00::c00a:0:0:f2: icmp_seq=4 ttl=63 time=32.0 ms
64 bytes from fd00::c00a:0:0:f2: icmp_seq=5 ttl=63 time=31.4 ms
64 bytes from fd00::c00a:0:0:f2: icmp_seq=6 ttl=63 time=32.1 ms
64 bytes from fd00::c00a:0:0:f2: icmp_seq=7 ttl=63 time=32.4 ms
^C
```

CONCLUSIONS

- Running the same MAC implementation in different platforms is feasible.
- Seamless communication
- Effort to deploy compatible MAC implementation to diferrent devices is minimized

POST MORTEM

- There is need of MAC implementation to be portable
- Timing constraints are hard to handle between heterogeneous devices
- TAISC can abstract all hardware differences and provide a unified MAC implementation environment

Radio Virtualization for Future Mobile Networks

GOALS

- **Show that radio virtualization is an enabler for base stations with multi-radio capabilities.**

Challenges

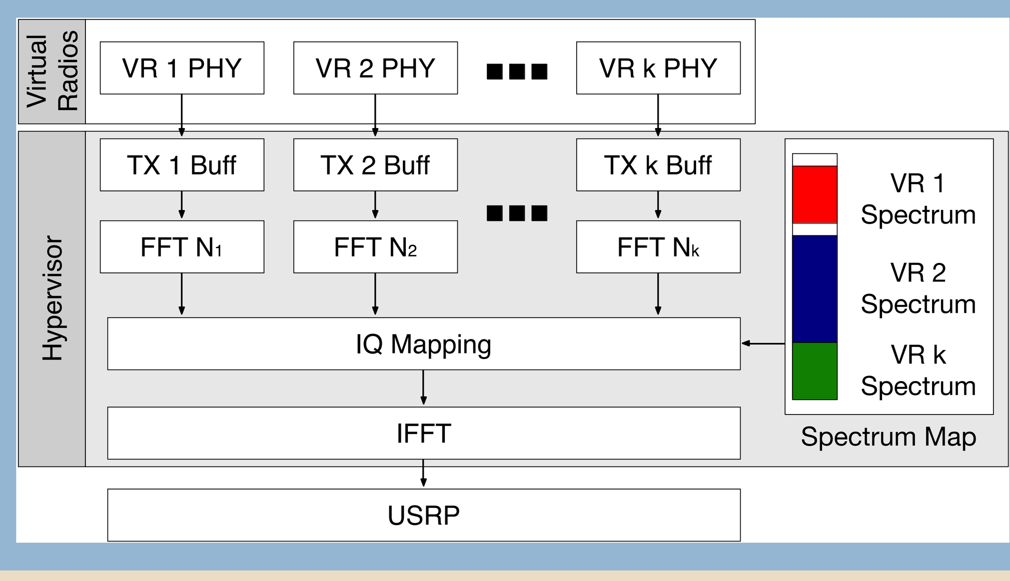
- **Design a hypervisor for Software Defined Radio (SDR) that ensures coexistence, isolation, and programmability.**
- **The hypervisor must be technology-agnostic, non-intrusive, and highly optimized.**

CONTEXT OF THE EXPERIMENT

- **5G networks is must provide connectivity services to devices with vastly different requirements, ranging from mobile subscribers with high-bandwidth services (*e.g.*, video-streaming) to IoT devices with bursty and low-bandwidth services (*e.g.*, assisted living monitoring)**
- **Providing a single "one-size-fits-all" air-interface is not desirable.**
- **Instead, future mobile networks should be flexible, providing different air-interfaces for particular users and applications.**

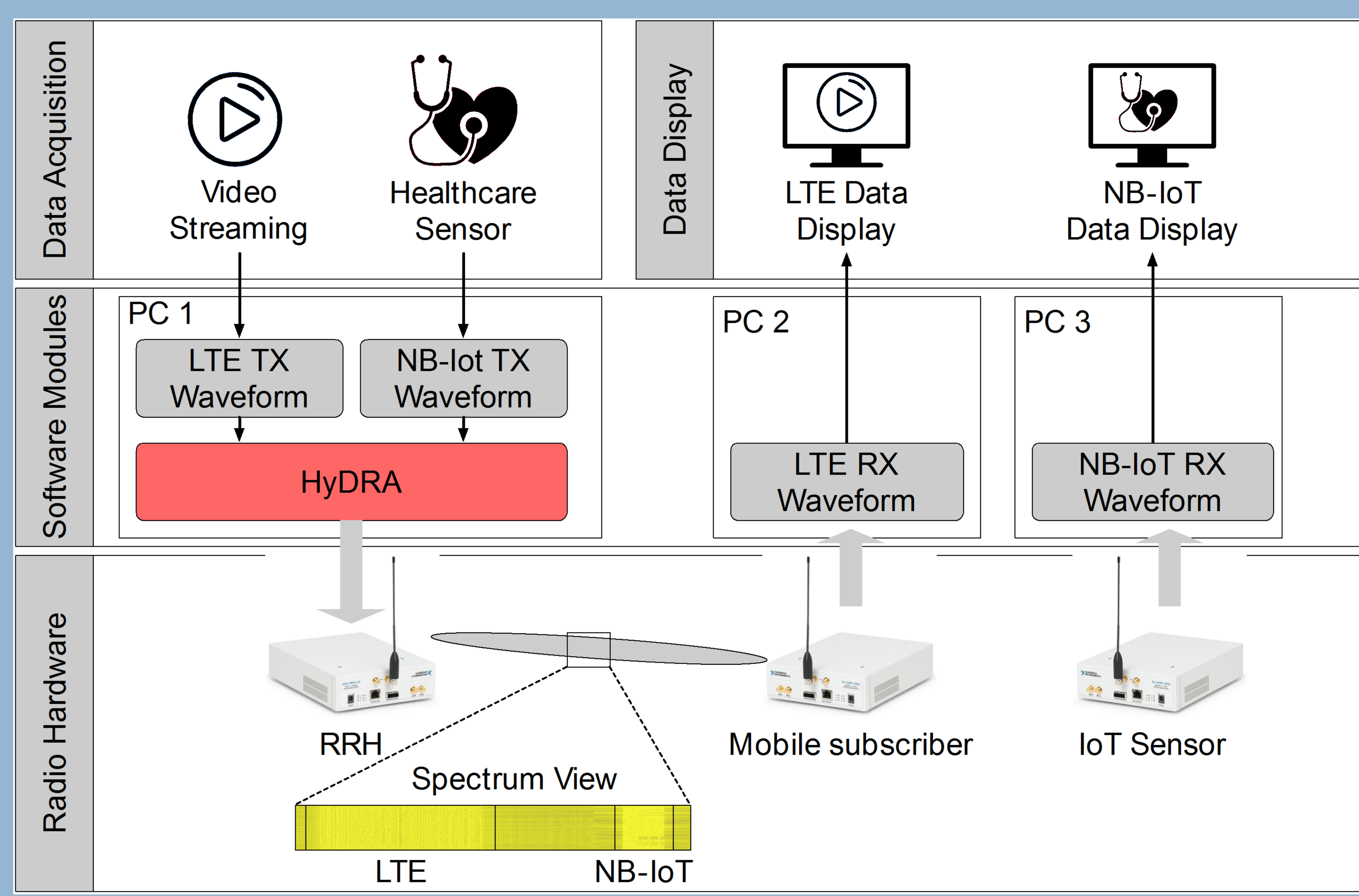
IMPLEMENTATION

- **Hypervisor for Software Defined Radios (HyDRA)**
- **Manageable through Wishful UPIS;**
  - Create virtual radios;
  - Change configuration of virtual radios;
  - Obtain performance metrics from virtual. radios



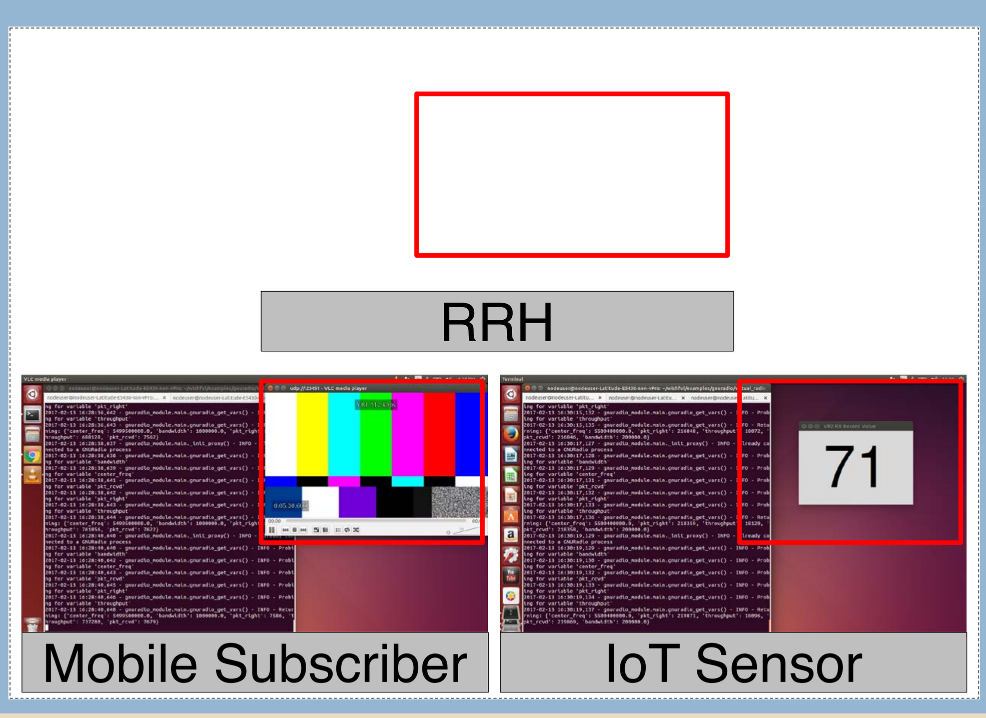
DEMO SETUP

- **1 USRP transmitting two different radio access technologies;**
  - LTE: Video-streaming;
  - NB-IoT: Healthcare sensor.
- **1 USRP acting as a mobile subscriber;**
  - Shows the video being received.
- **1 USRP acting as a NB-IoT receiver;**
  - Shows the data from the heartbeat sensor.



RESULTS

- **Screen capture from the transmitter, mobile subscriber and IoT sensor receiver:**



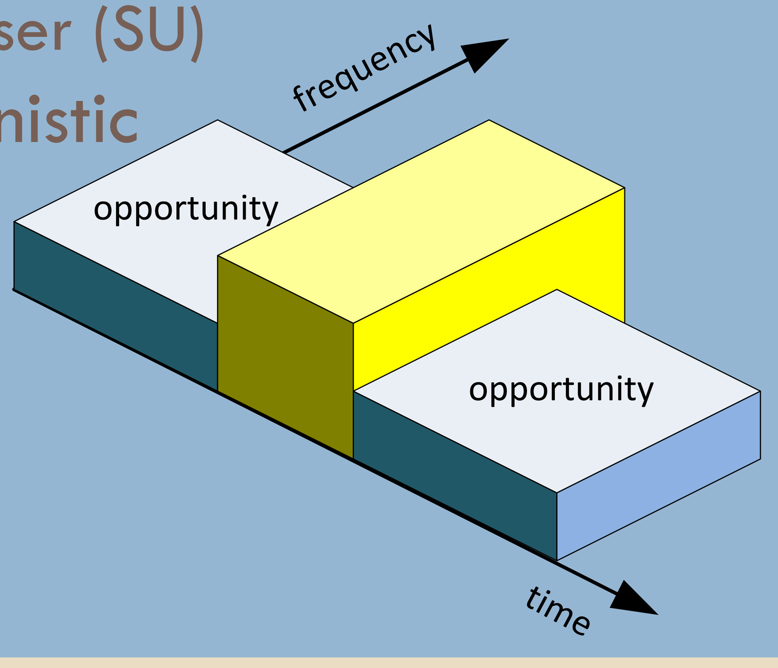
CONCLUSIONS

- **We demonstrate radio virtualization as a mechanism to provide versatile connectivity services in 5G mobile networks.**
- **To this end, we designed a hypervisor capable that multiplex IQ samples from different virtual radios into a single signal.**
- **Our demonstration is aligned with standardizations efforts made by 3GPP that consider a base station executing LTE and NB-IoT virtual radios.**

Enabling LTE-U/Wi-Fi Coexistence Through Cognitive Radio

MOTIVATION

- **Rapid growth** in the use of wireless devices and appearance of novel applications.
- **WiFi** is the **dominant access technology** and there is strong trend towards WiFi meshing & densification.
- **5 GHz ISM band** is being used by current and future 802.11standards.
- **"LTE in Unlicensed"** (LTE-U) constitutes a new source of interference with strong impact on WiFi in 5 GHz spectrum.
- **Cognitive Radio** is a promising approach to overcome spectrum crunch for WiFi:
  - LTE-U as primary user (PU),
  - WiFi as secondary user (SU)
- WiFi will perform opportunistic spectrum access to avoid collisions with LTE-U ON phases



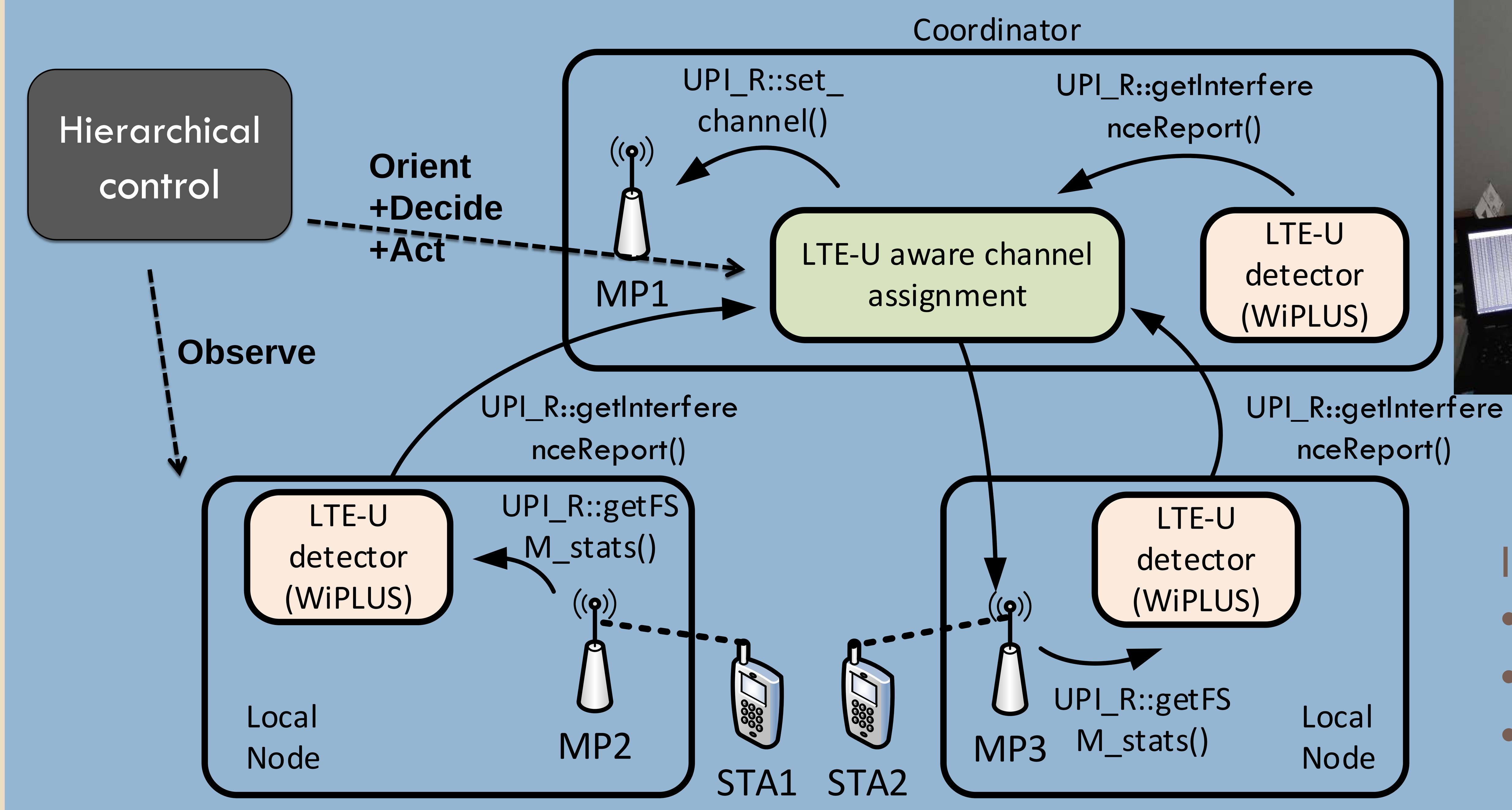
CHALLENGES

- An experimenter would like to easily **prototype** own **LTE-U/Wi-Fi coexistence schemes** using Cognitive Radio approach,
- There are three **main challenges**:
  - LTE-U detection (duty cycle, timing, …),
  - Synchronization of WiFi with LTE-U,
  - Execution of coexistence schemes.
- Further, there is a need to find proper **abstraction** for the WiSHFUL UPI functions.

DEMO

- WiFi interference-aware **channel selection**, i.e. abandon channel suffering from LTE-U interference,
- **Cognitive loop**:
  - Observe = WiPLUS, Orient = Threshold, Decide = Best Channel, Act = Retune
- Challenges:
  - Data fusion from multiple mesh nodes,
  - Seamless and time synchronized global channel switching in the whole 802.11mesh

IMPLEMENTATION

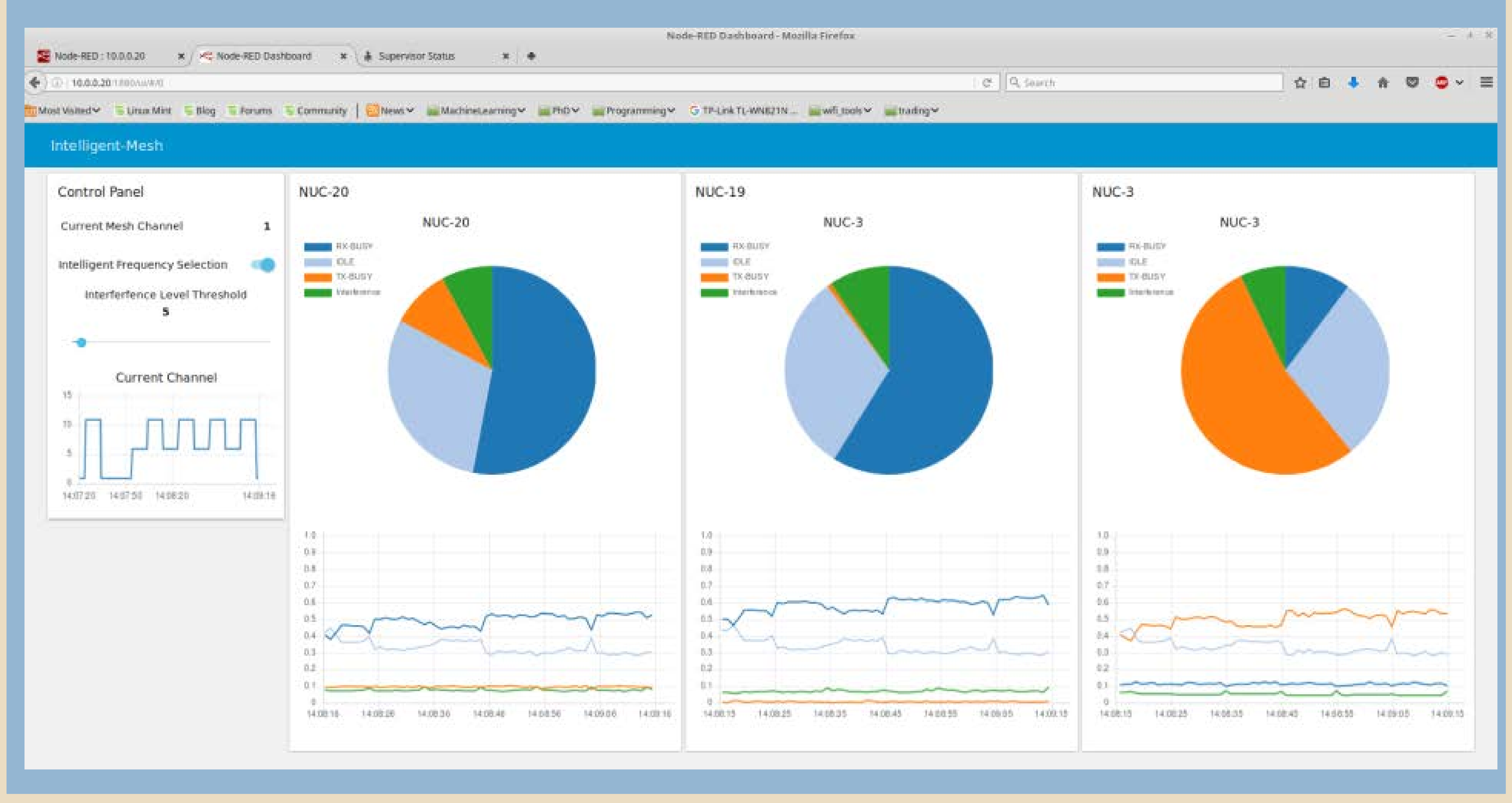


Demo setup.

Involved Wishful UPIs:

- *UPI_R::set_channel(),*
- *UPI_R::getFSM_stats(),*
- *UPI_R::getInterferenceReport()*

VISUALIZATION



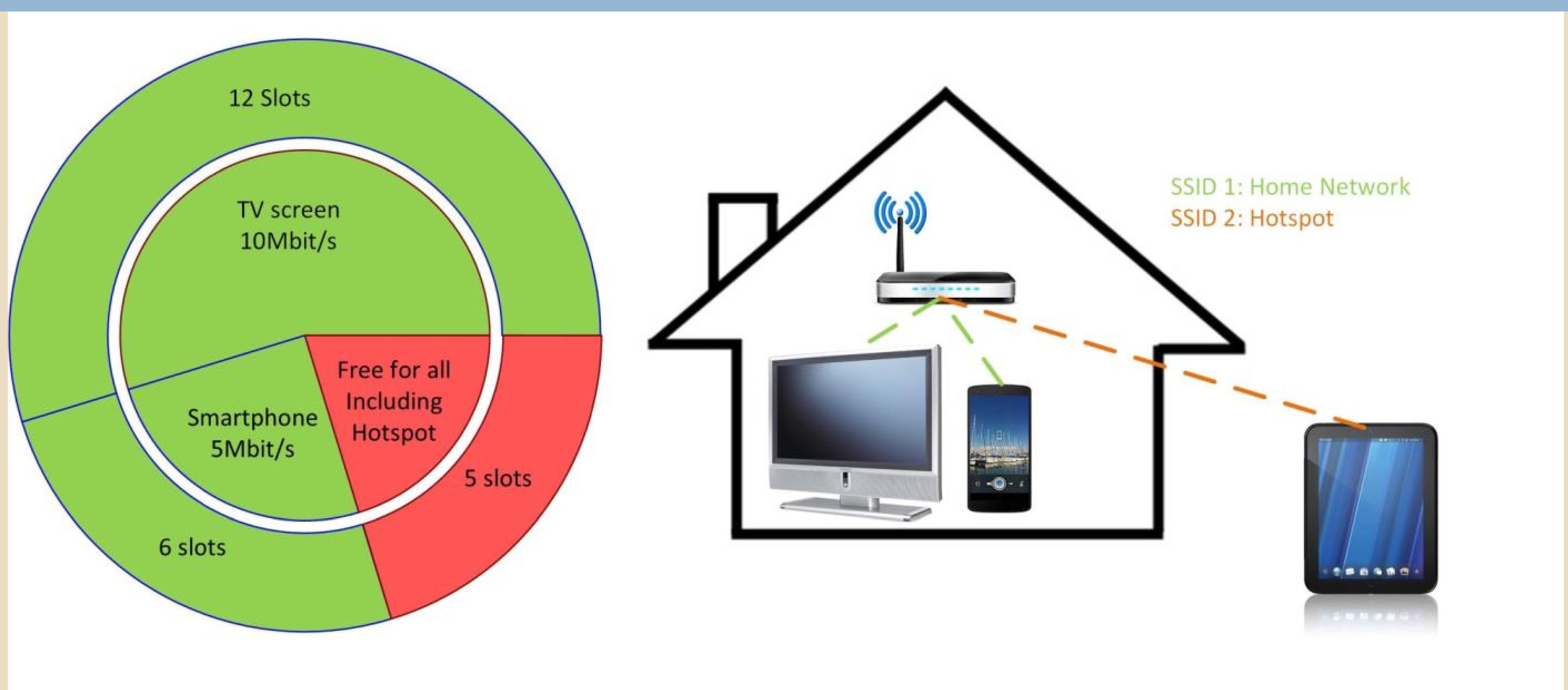
POST MORTEM

- Using Wishful an experimenter can easily prototype own cognitive radio solutions enabling WiFi/LTE-U co-existence
- Showcase demonstrated cognitive channel selection, i.e. fast abandoning of channel suffering from LTE-U interference,
- Wishful UPI functions hide complexity, i.e. global time synchronized channel switching, interference detection, etc.

Radio Slicing for virtualized Home WiFi Access Points

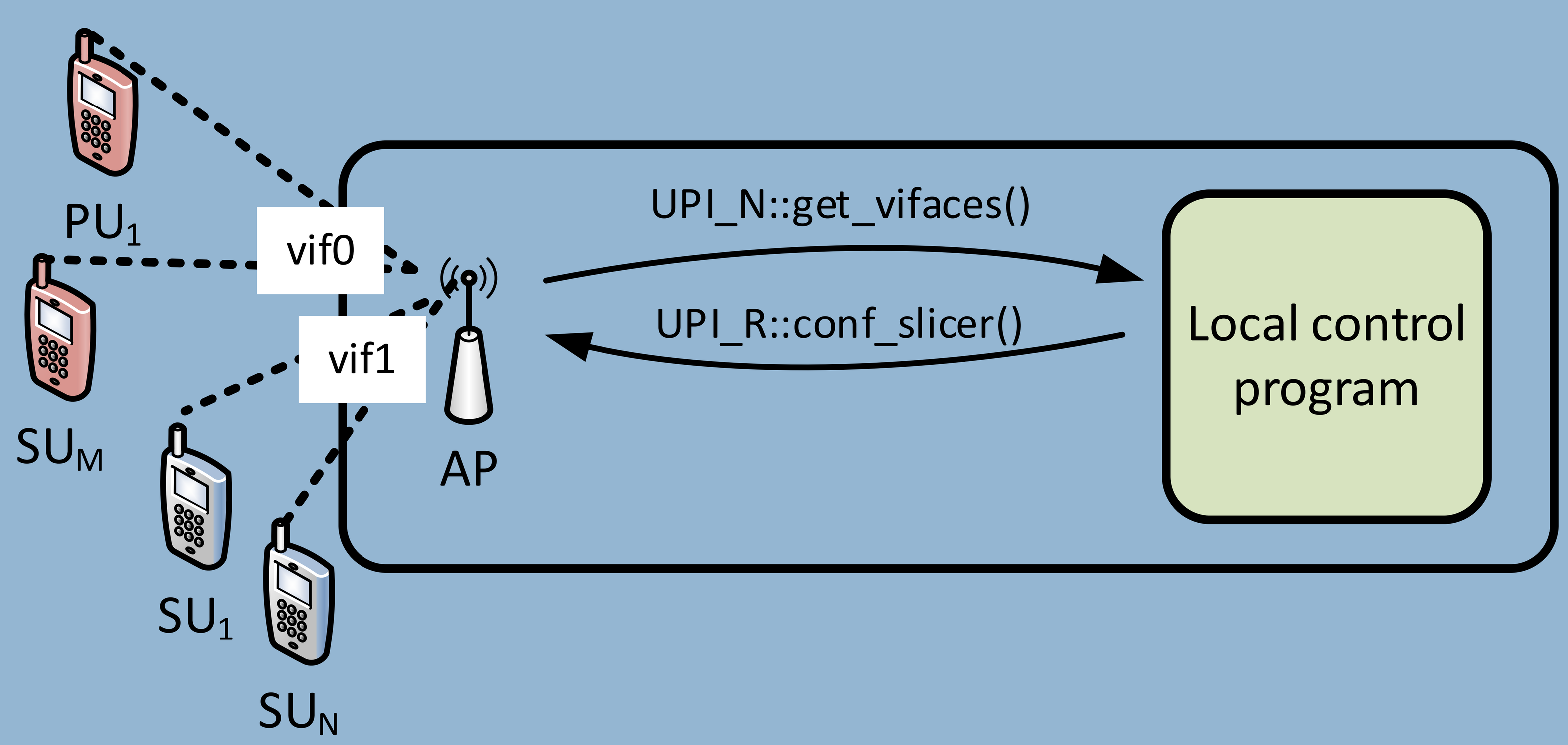
MOTIVATION

- **Virtualized WiFi Networks** are very common in nowadays home WiFi networks,
- One physical AP hosts **multiple virtual WiFi networks** identified through different SSIDs and BSSIDs,
- Examples are Hotspot Networks installed by cellular providers or „Guest Networks“ installed by the resident.

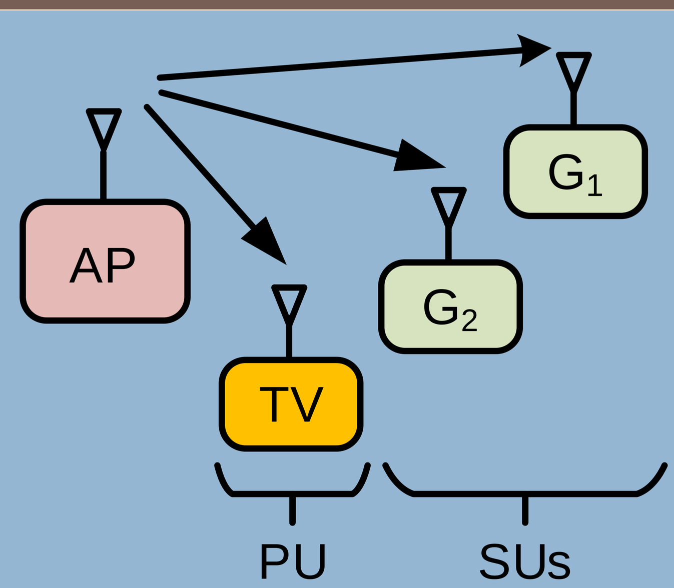


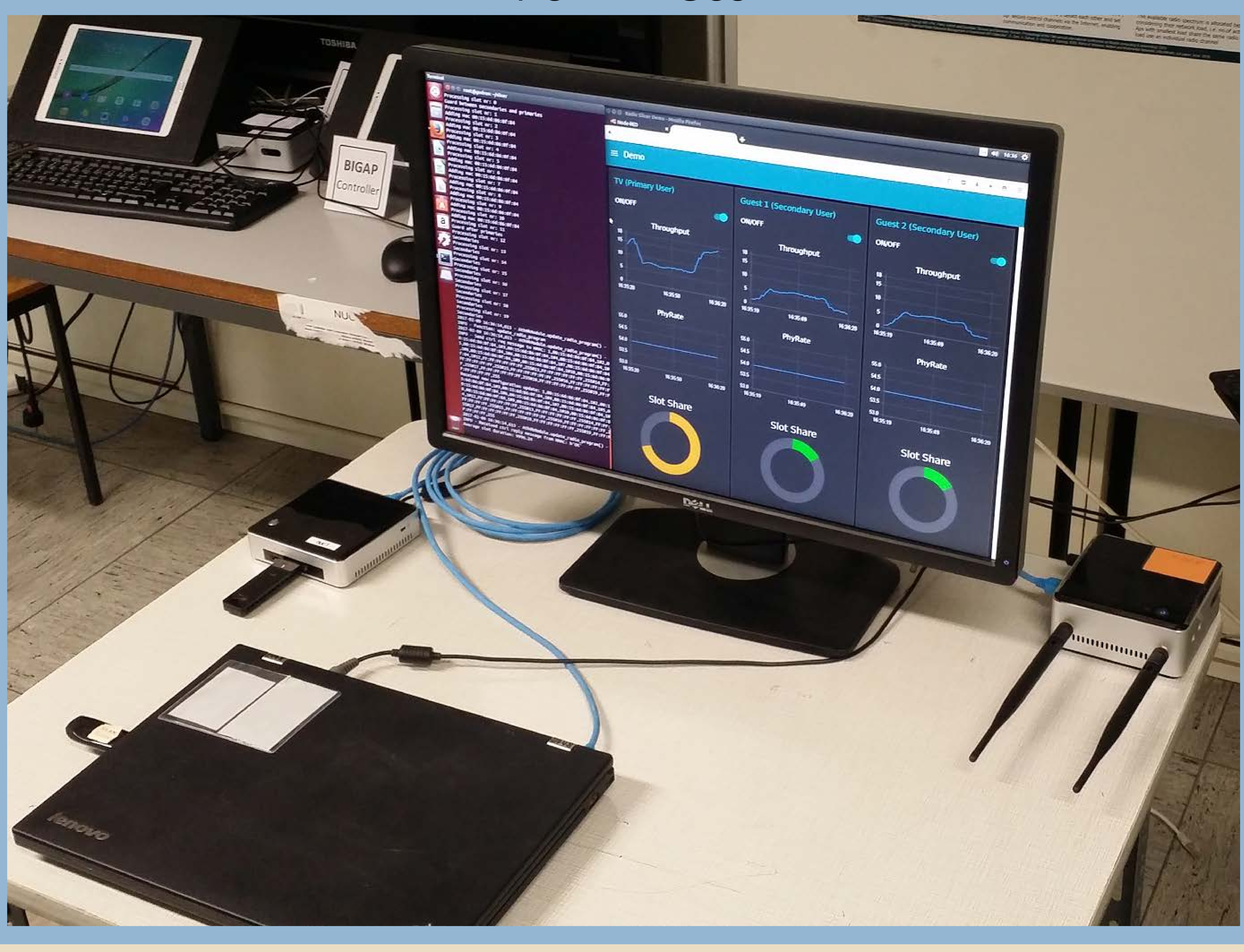
IMPLEMENTATION

- Fully implemented as local Wishful control program running on WiFi AP



DEMO SET-UP





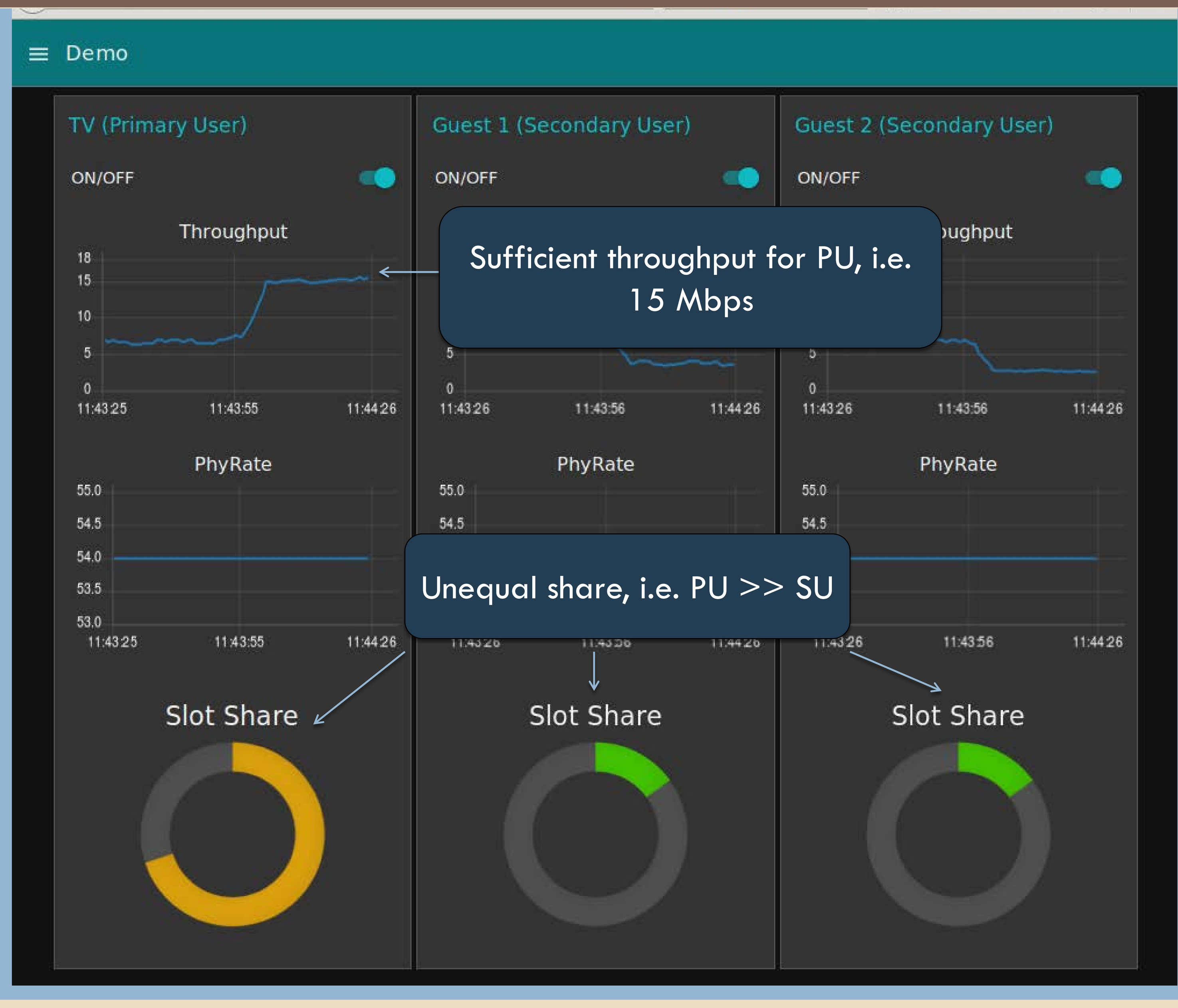
CHALLENGES

- The Primary User (PU, AP owner) should not even be aware of the fact his resources are shared with Secondary Users (SU) → **strict isolation**
- Current approaches, e.g. meSDN, cannot guarantee strict isolation of PU.
- **Our approach** - MAC layer level slicing to guarantee bandwidth and traffic isolation for PU:
  - End-users are able to set fixed bandwidth guarantees for their PU devices,
  - Radio slicer takes care of slice assignment by taking into account the quality of the radio channel (CQI, PER)

WISHFUL FUNCTIONALITY

- **Virtual interfaces** - support of configuration and management of virtual interfaces for 802.11,
- **Link monitoring** - advanced link monitoring, i.e. bitrate, PER, airtime utilization,
- **Radio slicer** - configuration of radio slicer by setting target bitrate (MAC layer) for each virtual interface (→ PU/SU)

VISUALIZATION



POST MORTEM

- Using the Wishful framework an experimenter can **easily prototype** own algorithms and policies for **WiFi radio slicing**,
- Showcase demonstrated WiFi radio slicing for isolation of primary (PU) and secondary users (SU)
  - Exclusive time slots are assigned to PU,
  - Remaining time slots are shared by PU and all SUs
- Wishful UPI functions hide complexity, i.e. computation of radio slice size (which depends on PHY rate, PER).

PORTABLE TESTBED

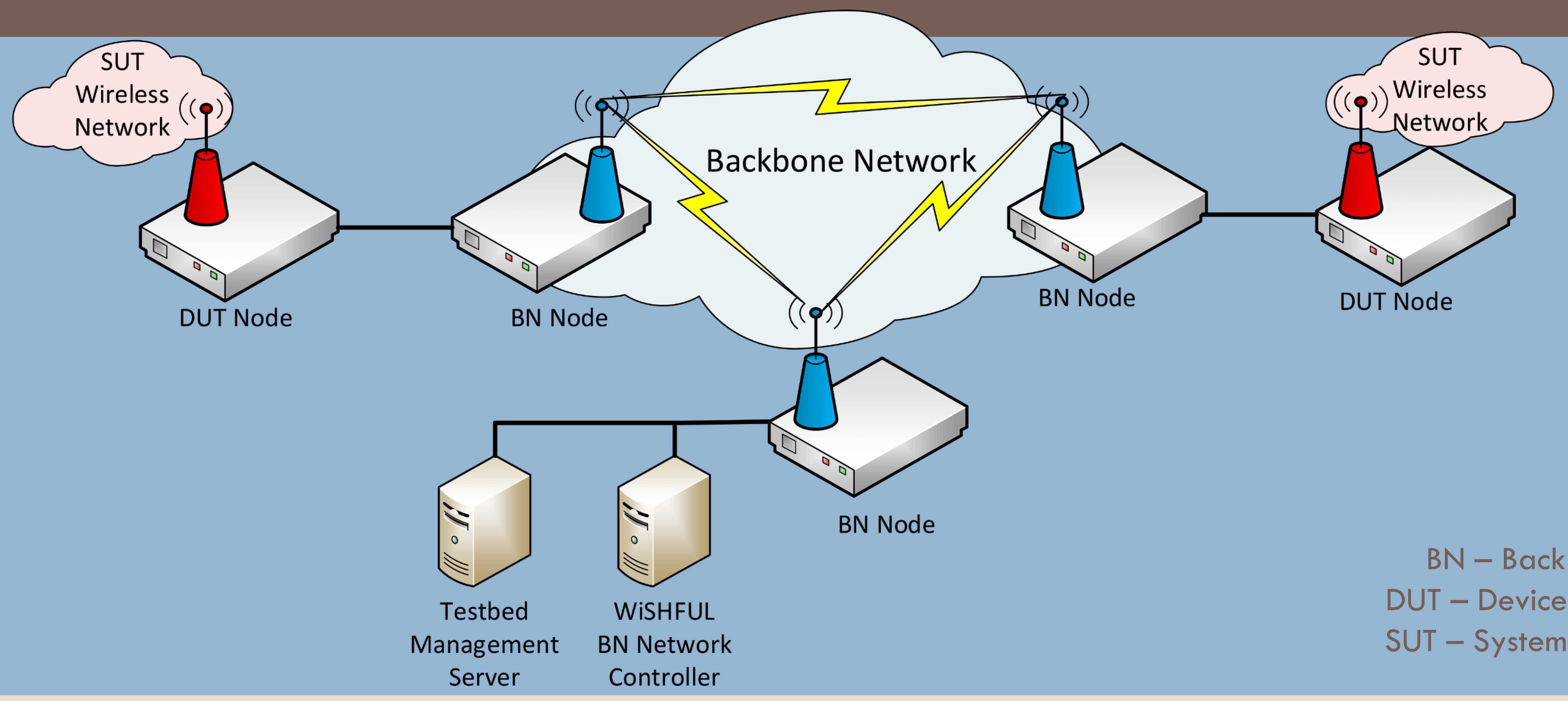
GOALS

- **Help researchers to increase realism of their experiments and examine their prototypes in heterogeneous environments**
- **Design and develop a new testbed platform that supports portability and facilitates execution of wireless network experiments in real world scenarios**

CHALLENGES

- **Fed4FIRE compliance** – an experimenter should be able to use the same tools as in fixed testbed
- **Wireless Backbone Network** - eliminate configuration overhead, reduce the impact of interference, provide QoS
- **Experiments in harsh** environments - frequent transport of Portable Testbed requires robust hardware

ARCHITECTURE



BN – Backbone Node
DUT – Device Under Test
SUT – System Under Test

RESULTS

Controller in flight case
Easy packing and transport

DUT based on Intel NUC
Custom antenna mounts
PoE powered from controller

Backbone node
Custom mesh network
Control using UPIs

Battery powered
Easy testing on the go

Fed4FIRE compliant

WIRELESS BACKBONE

- **Wireless mesh network based 802.11n ad-hoc**
- **OLSR based routing protocol**
- **Self-healing and configuring**
- **L2 tunnelling between DUT nodes for full transparency**
- **Backbone Network supervised by WiSHFUL controller:**
  - **seamless channel change**
  - **traffic prioritization in NET and MAC**
  - **channel occupation measurements**
  - **airtime fairness**
  - **mesh visualisation**

CONCLUSION

- **Independent WiSHFUL showcases running on Portable Testbed with wireless backbone enabling flexible topologies**
- **Portable Testbed is offered to the research community**
- **Toolset available in the public WiSHFUL repository – the Portable Testbed can be replicated by experimenters**

LINK ESTIMATOR SELECTION IN LARGE SCALE SENSOR NETWORKS

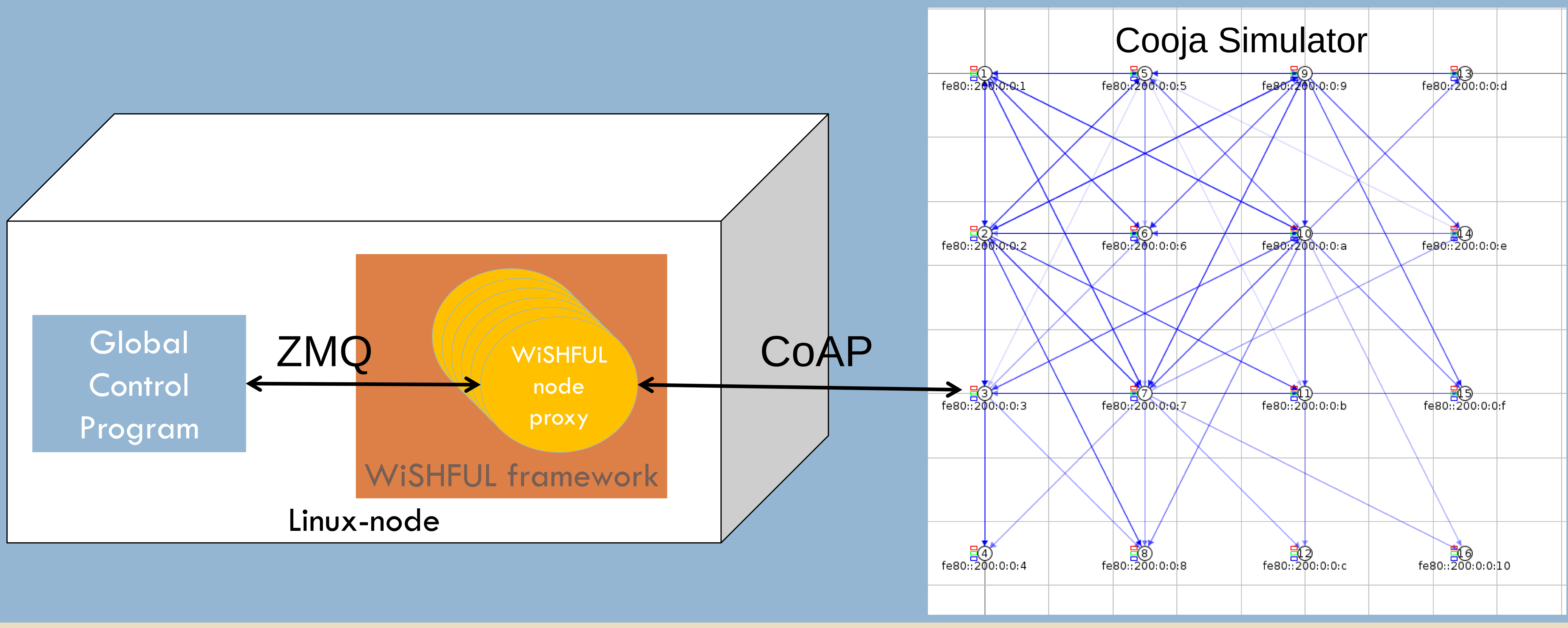
GOALS

- UPI enabled Control of Contiki IPv6 mesh network.
  - Monitoring IP-UDP-TCP-RPL statistics.
  - Configuring RPL routing protocol.
- Y2 extensions to WiSHFUL.
  - Integration of Cooja network simulator.
  - UPI control flows over CoAP.
  - Refactored local monitoring and configuration engine in Contiki.

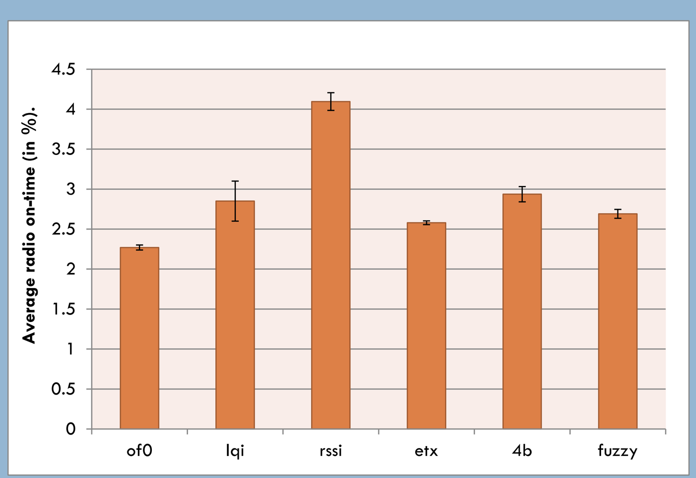
CHALLENGES

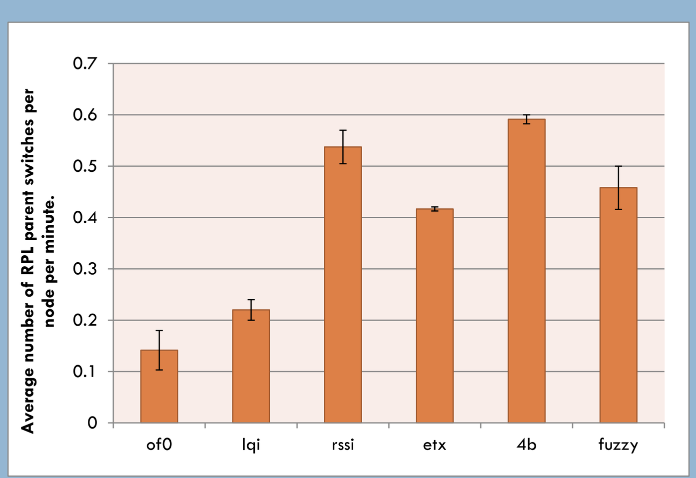
- Enabling in-band and out-of-band control over CoAP.
- Switching link estimator on-the-fly.
- WiSHFUL services on constrained devices (RPC, node discovery, bootstrapping).
- Large scale sensor network experiments.

DEMO SETUP

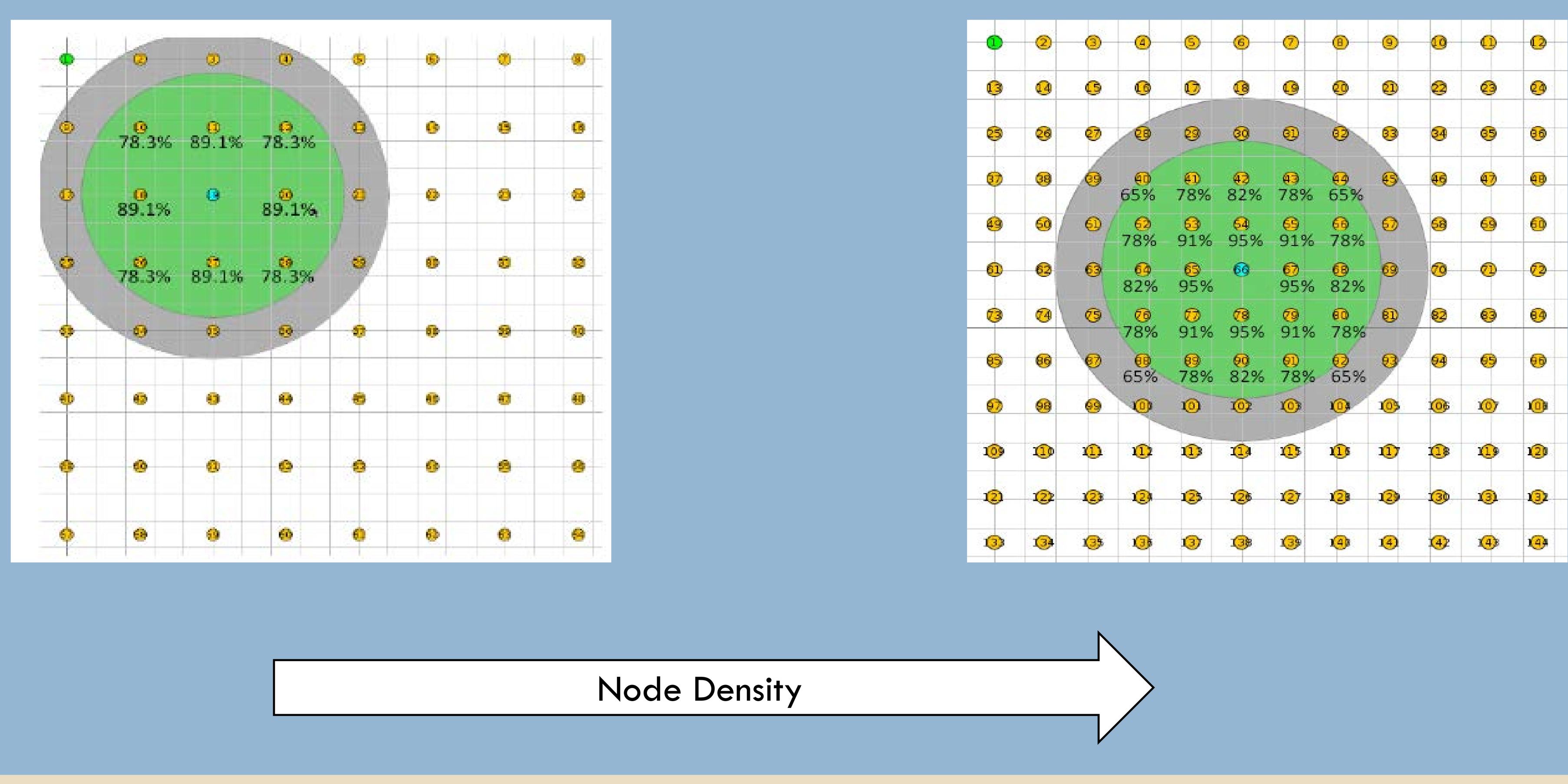


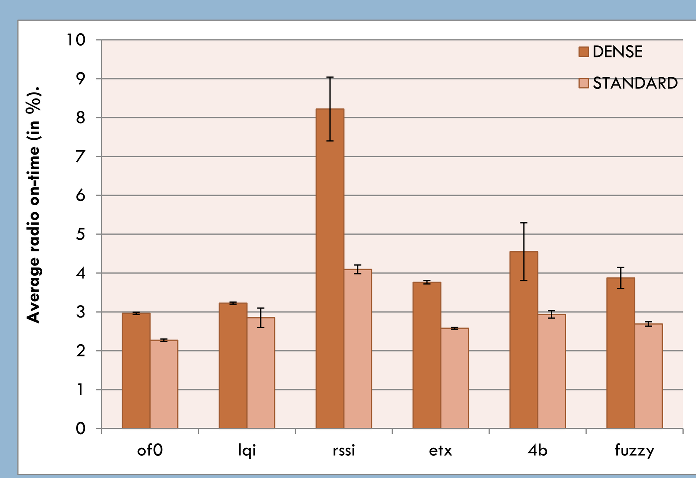
RESULTS

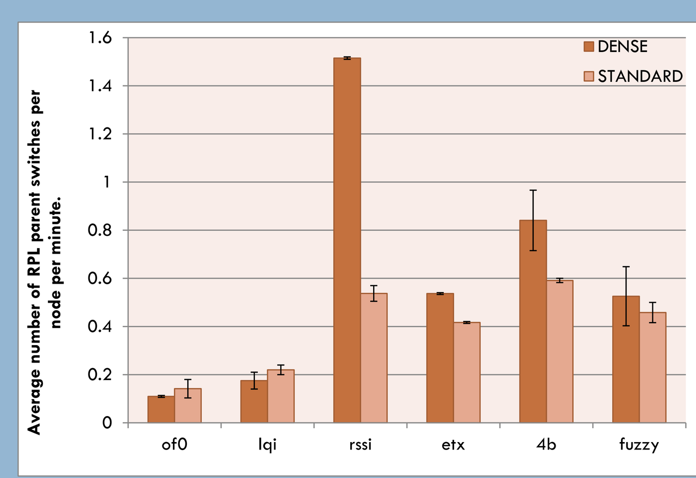




Impact of node density on routing protocol performance for different link estimators







CONCLUSIONS

- Node density has a serious impact on the performance of the routing protocol.
- Cooja combined with WiSHFUL drastically reduces the time to develop and test large scale sensor network experiments.
- It is possible to provide WiSHFUL services on constrained devices as it is on more powerful devices.
- In-band control flows have a significant impact on the network performance.

POST MORTEM

- In-band global control is quite slow.
  - ➢ Multicast would decrease delays.
  - ➢ Time-scheduling should be implemented.
- The impact of in-band control flows should be further analyzed and reduced.
  - ➢ Local control programs in Contiki can reduce the overhead drastically.

STRALE: MOBILITY-AWARE PHY RATE AND FRAME AGGREGATION LENGTH ADAPTATION IN WLANS

GOALS

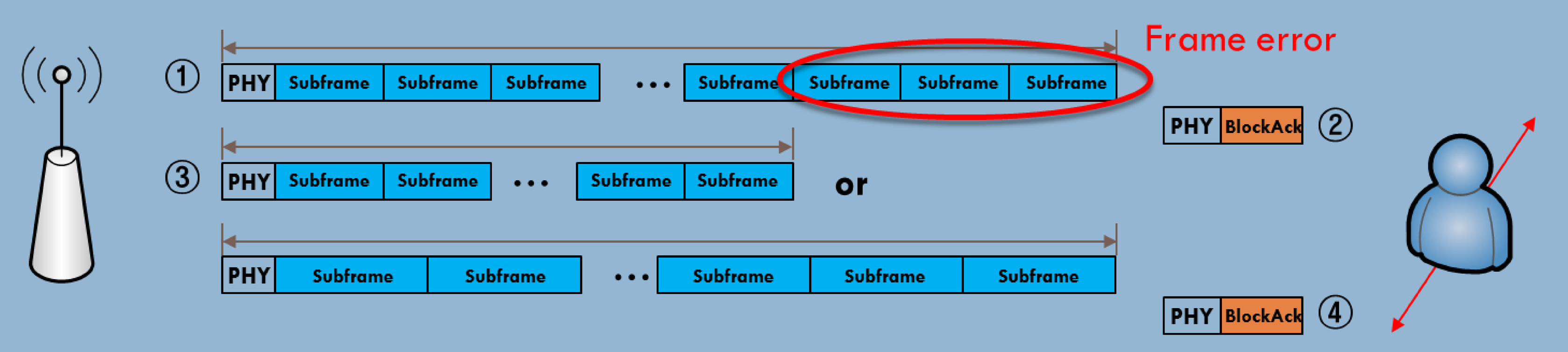
- **STRALE:** Standard-compliant and mobility-aware PHY rate and frame aggregation length adaptation in WiFi networks
- **Implementation of STRALE**
  - ✓ WiSHFUL framework
  - ✓ Local controller

CHALLENGES

- **UPI functions to implement**
  - ✓ MCS selection and A-MPDU length setting
  - ✓ Enabling/disabling STRALE
    - Statistics management
    - PHY rate and A-MPDU length decision
    - Run-time adaptation dynamically

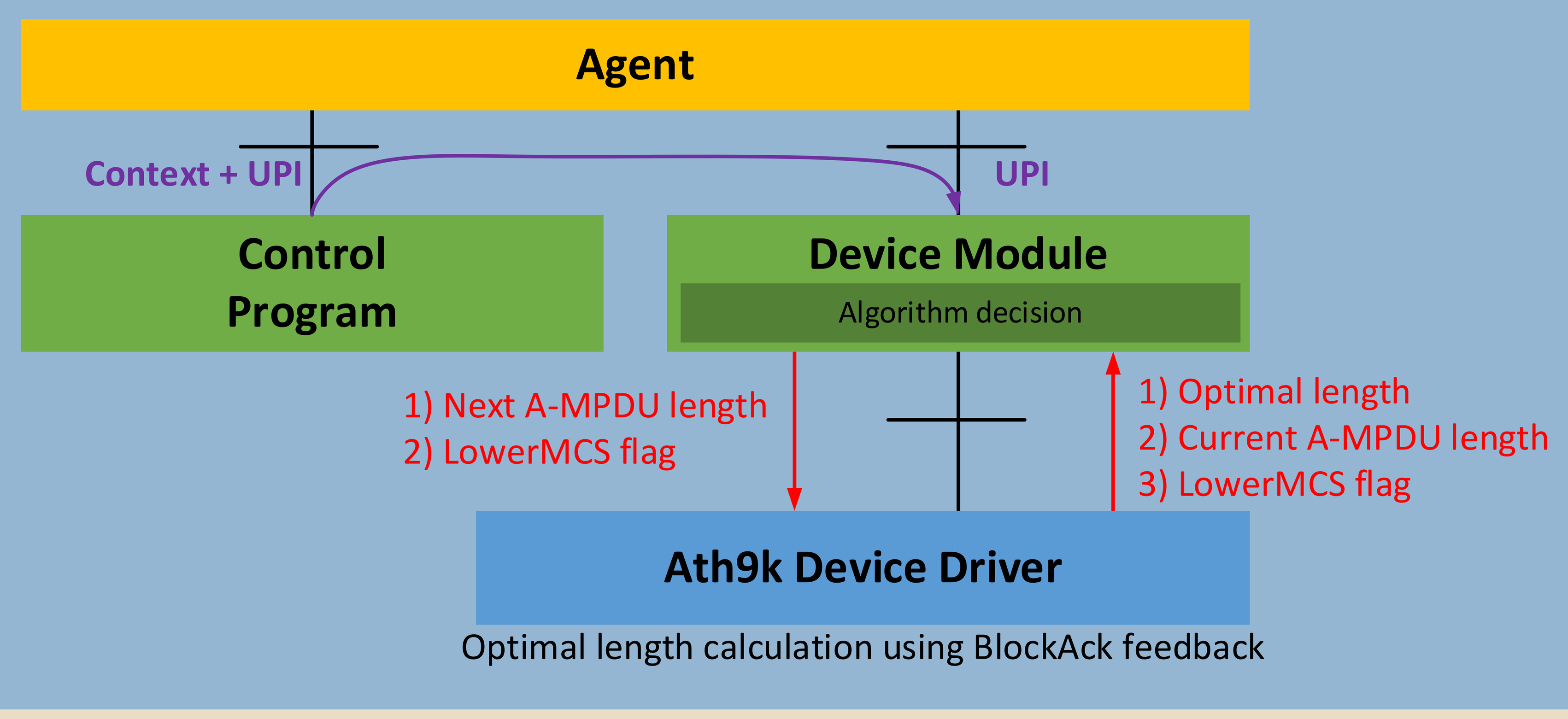
STRALE ON WiSHFUL FRAMEWORK

- **STRALE**
  - ✓ Determining the degree of the user mobility
  - ✓ Dynamically adapting PHY rate and A-MPDU length simultaneously during runtime



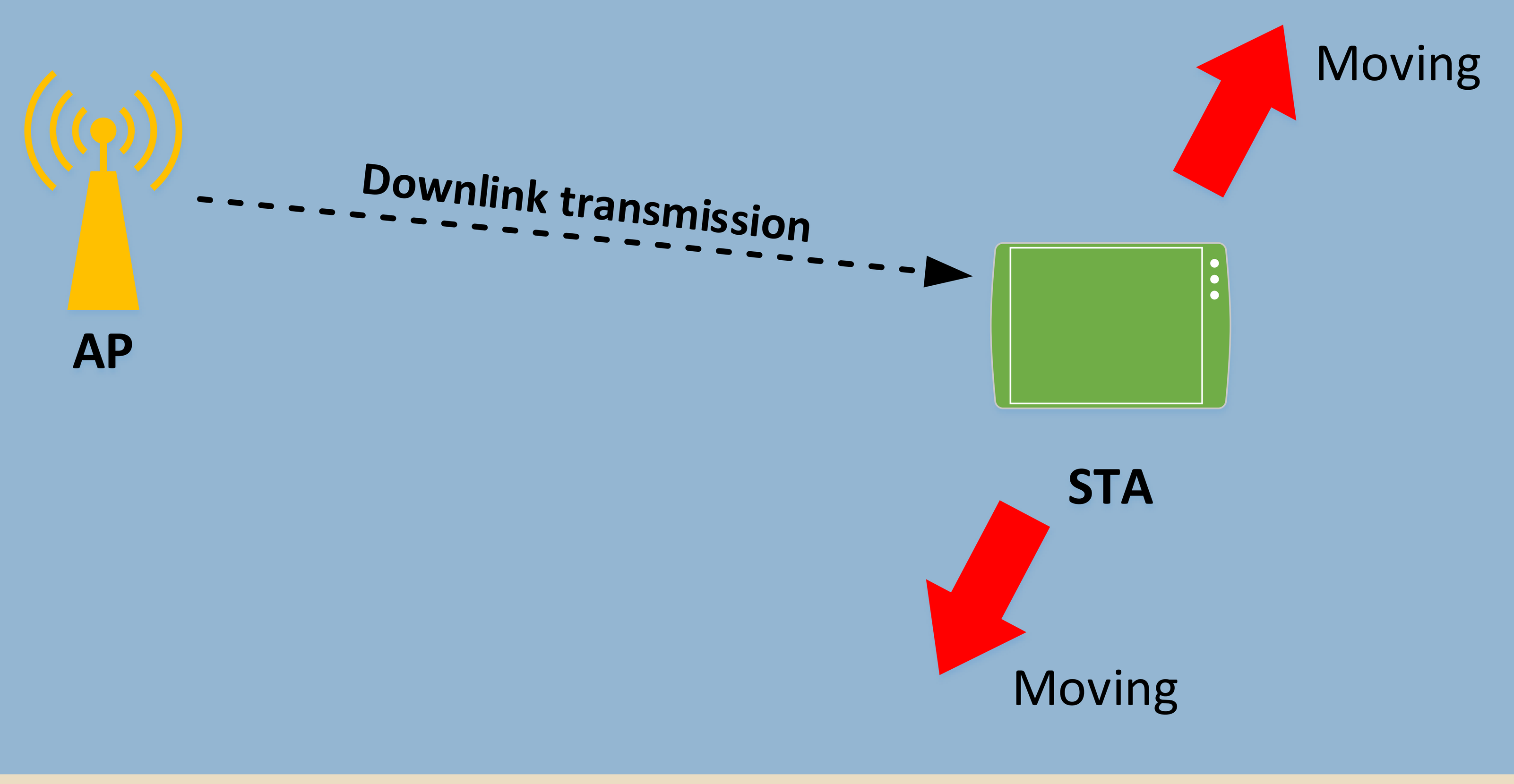
- **UPIs on local controller**

toggle_strale(phy_dev, strale_onoff)
ampdu_length(phy_dev, length)
fixed_rate(phy_dev, rate)
strale(phy_dev)

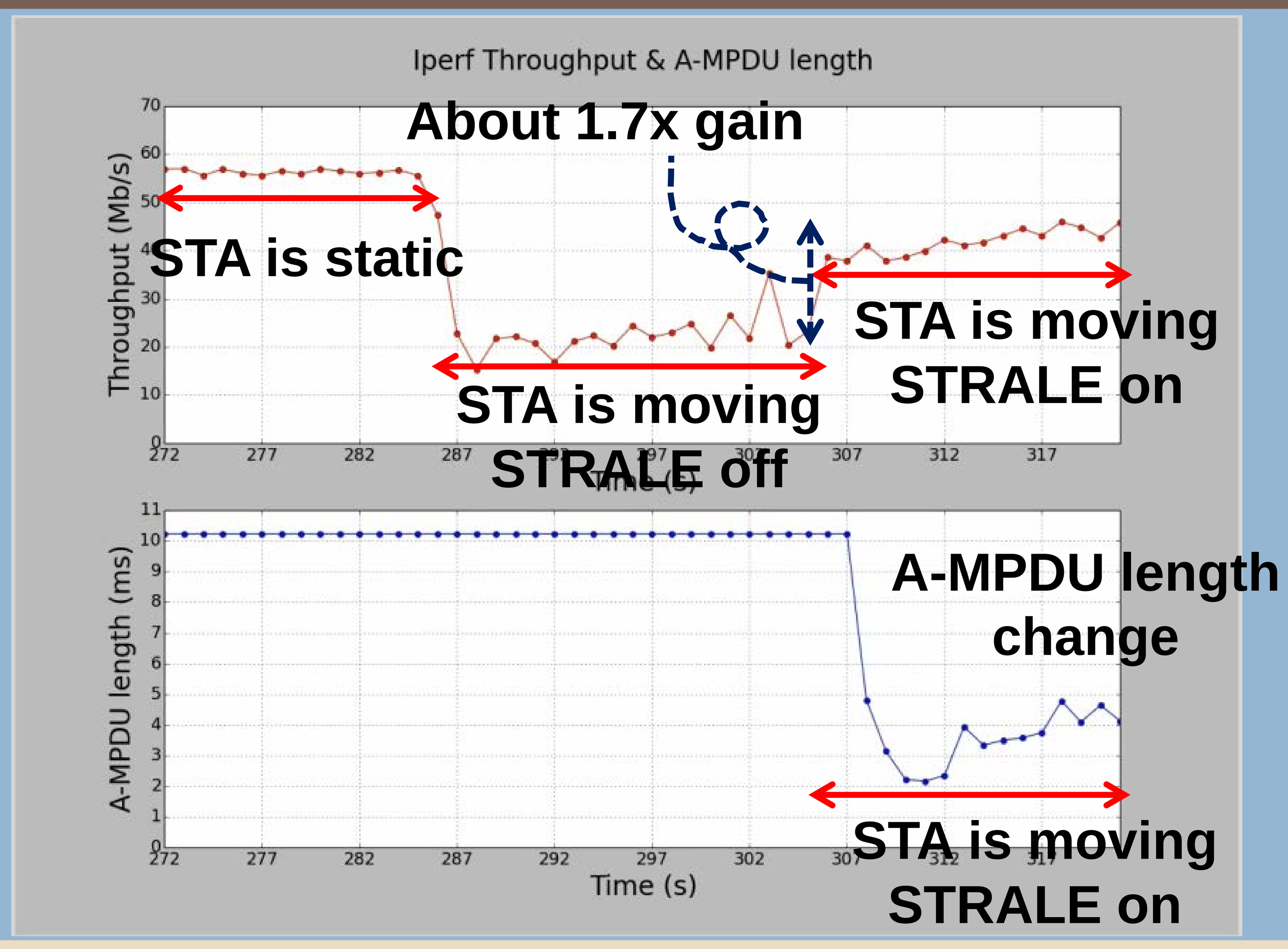


DEMO SETUP

- **Demo environment**
  - ✓ STRALE on AP-side
  - ✓ STA is moving around AP



PERFORMANCE EVALUATION



INNOVATIONS

- **Run-time (per A-MPDU) WiFi performance enhancement in mobile scenarios**
  - ✓ Dynamic PHY rate and A-MPDU length adaptation algorithm
- **Successful implementation of STRALE on WiSHFUL framework**
  - ✓ Modification of device/control module in local controller
  - ✓ Verification of easy prototype

CONCLUSIONS

- **WiFi performance enhancement: Dynamic adaptation of PHY rate and A-MPDU length during run time**
- **Implementation of additional UPI functions**
  - ✓ MCS selection and A-MPDU length setting
  - ✓ Enabling/disabling STRALE
  - ✓ Enable communication between WiSHFUL device module and device driver

Offline MAC performance prediction

GOALS

- To predict network performance using an offline trained model
- To enable online MAC selection based on multiple MAC prediction models.

Challenges

- Generate big dataset from a series of experiments
- Which features to select for extraction?
- How to evaluate the trained model's prediction accuracy?

CONTEXT OF THE EXPERIMENT

Process flow:

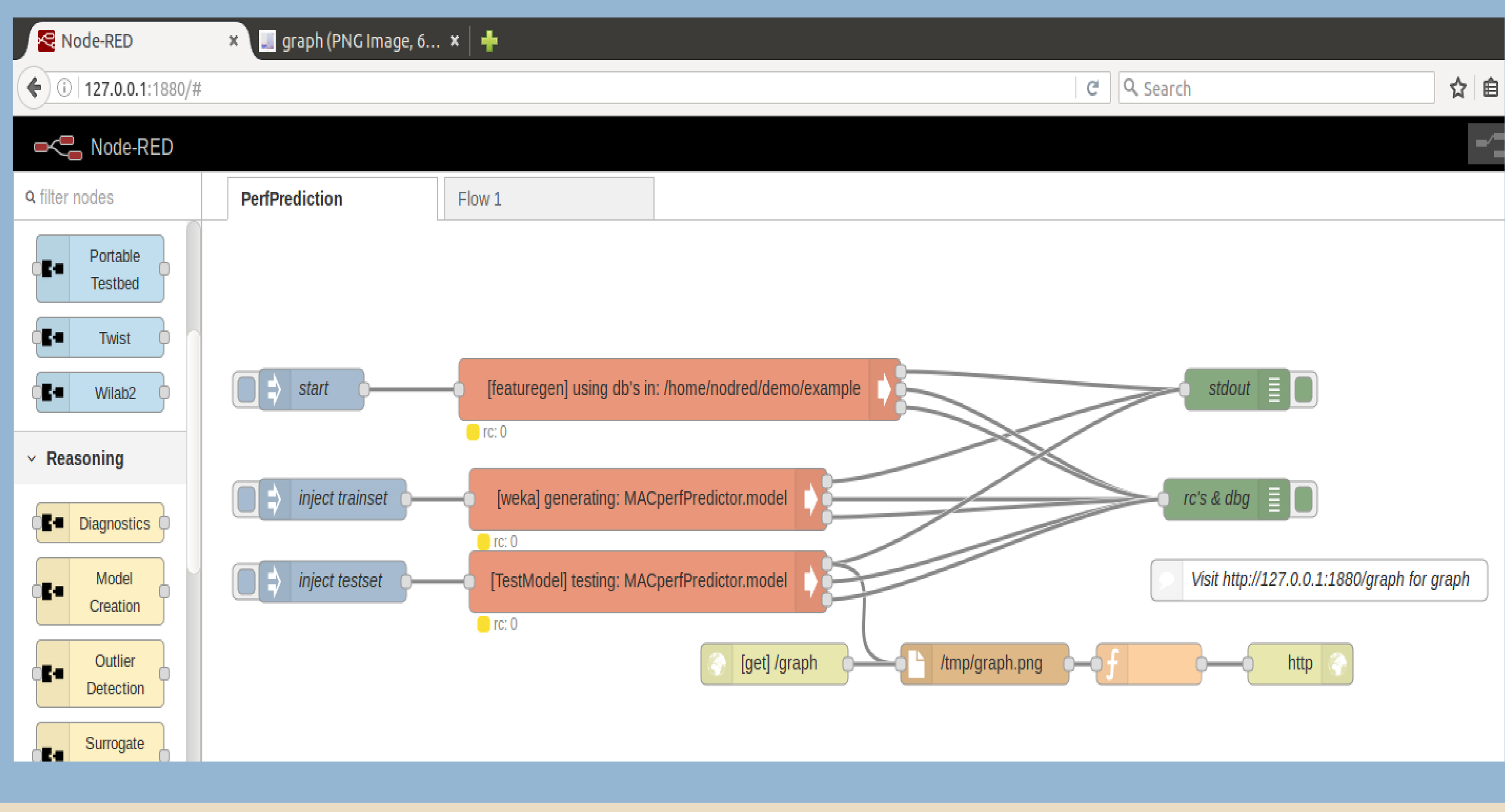
1. Construct a big dataset with raw data from a series of experiments by using WiSHFUL UPIs to gather the data.
2. Extract features from data set
3. Trained and validated the prediction model using a N fold cross-validation approach
4. Evaluate prediction accuracy of the model

Intelligence framework:

- Node-RED was used as a visualization and execution environment.

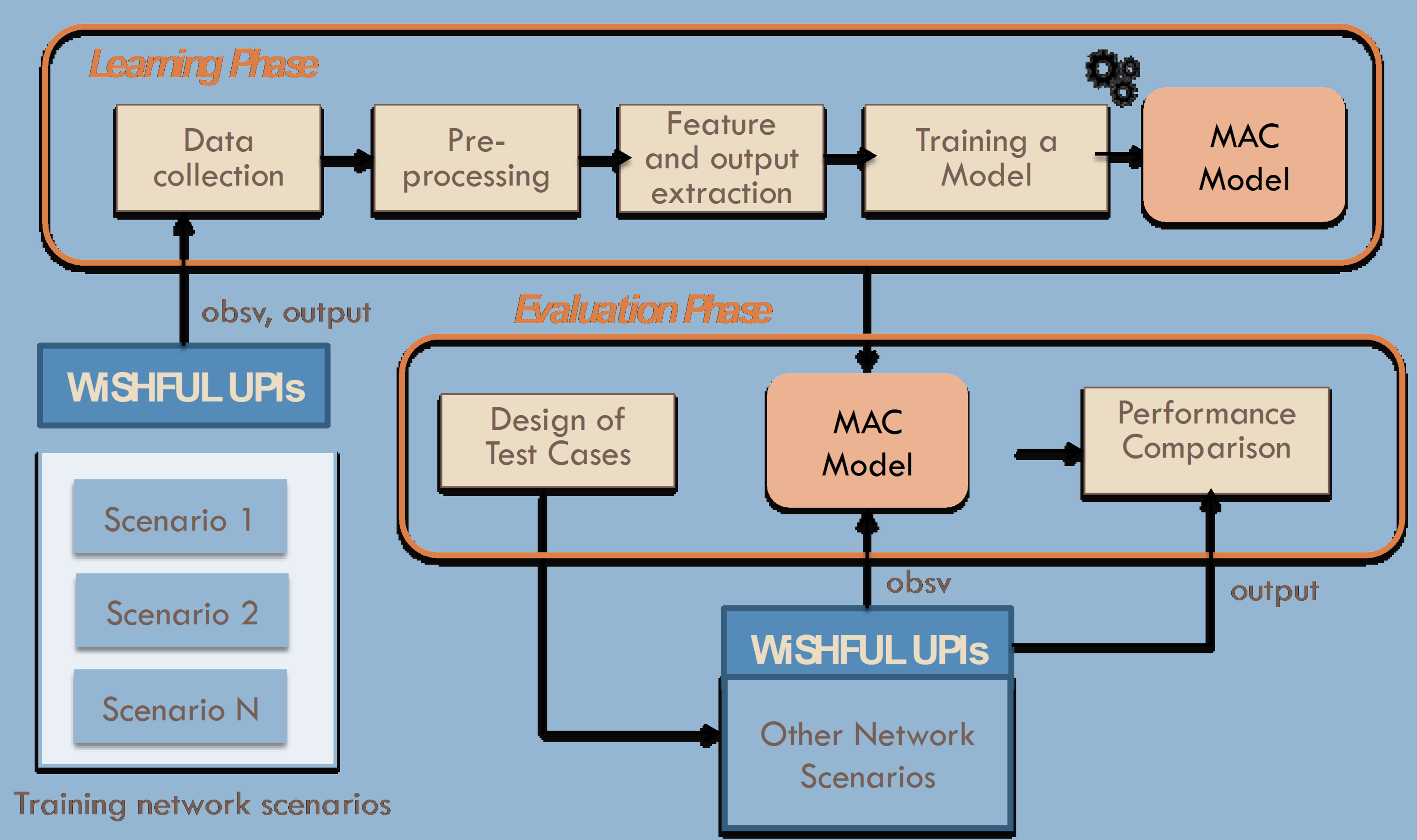
DEMO SETUP

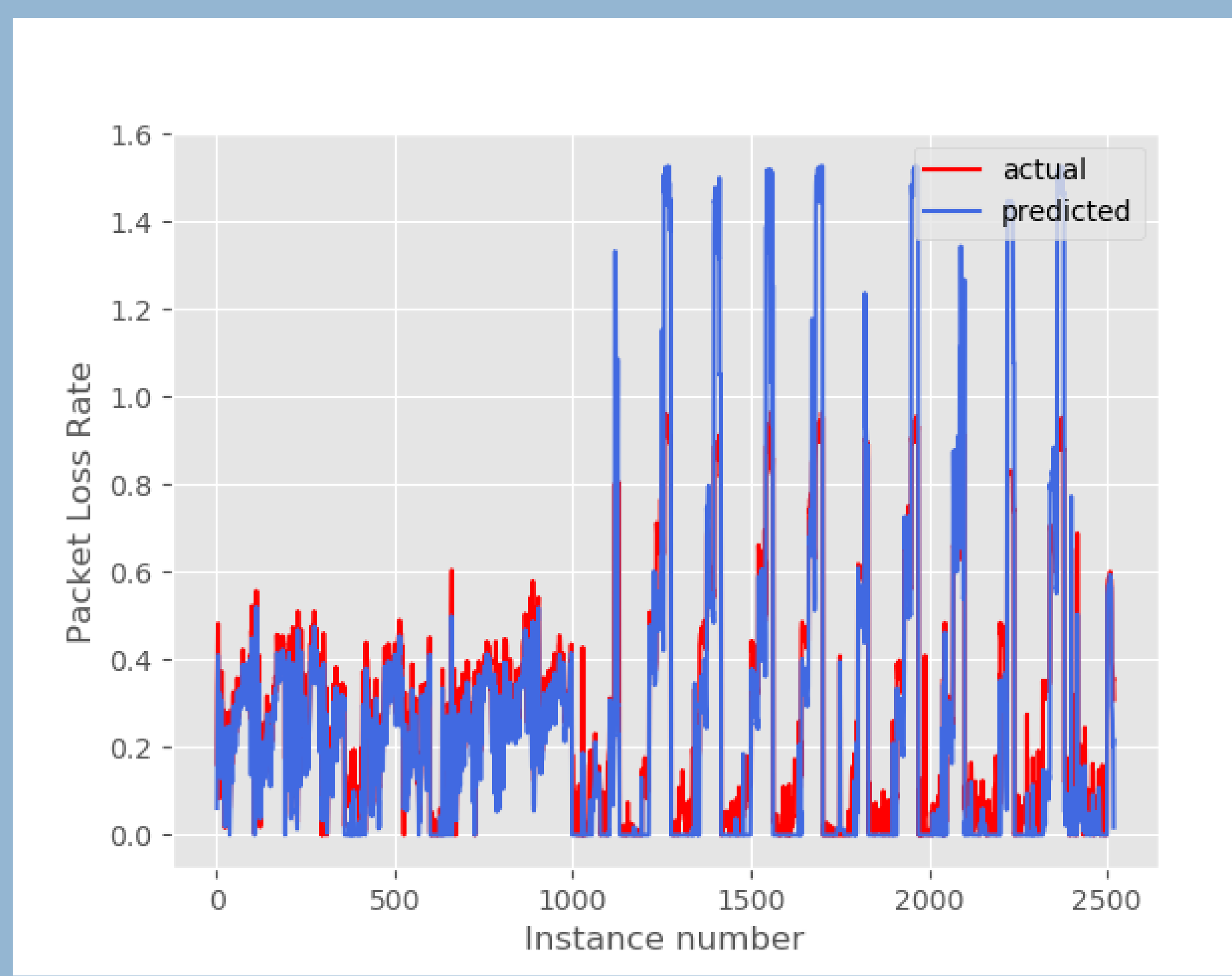
- In Node-Red various components are implemented to support the training and evaluation of a offline model.



RESULTS

The trained model is predicting the Packe Loss Rate of the network quite nicely





CONCLUSIONS

- Machine learning can play an important role in predicting Network performance and thus allowing the network to adjust accordingly
- Models for different MACs can be trained, and then used for online performance comparison leading to intelligent selection of best MAC protocol

POST MORTEM

- The first step towards an online MAC selection component is concluded.
- More trained MAC models are needed in order to enable multi-MAC performance comparison.

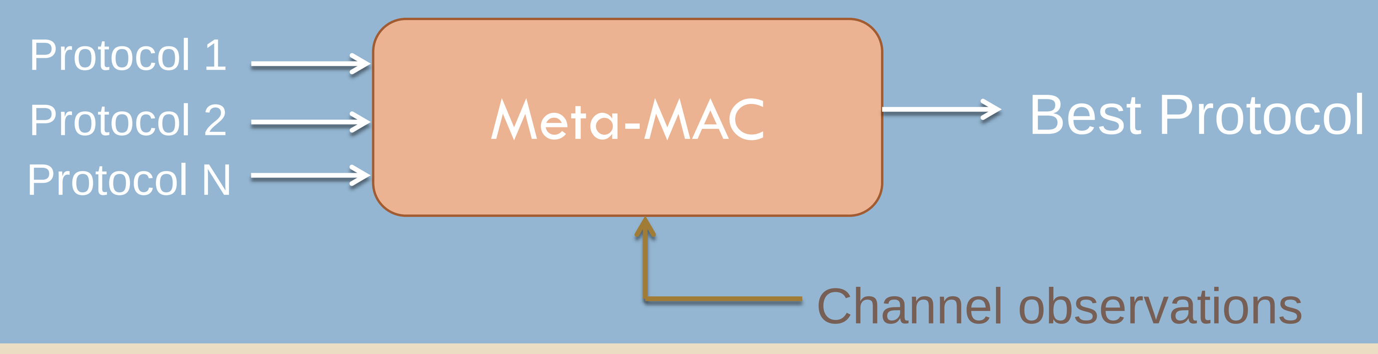
META-MAC

GOALS

Show how to learn the best possible MAC protocol

- **Slotted Aloha vs. TDMA**

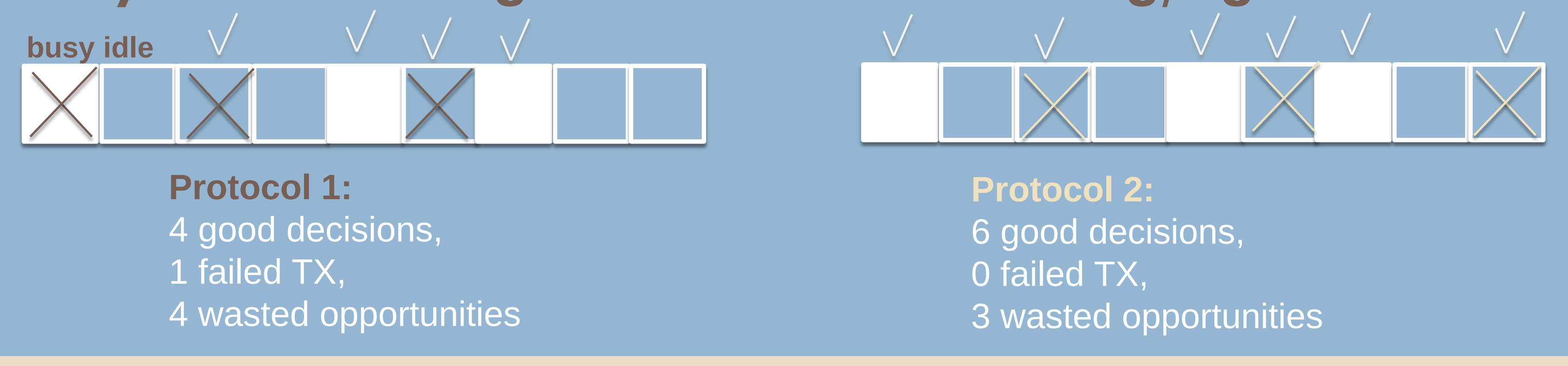
From rule-based decisions to automatic decisions taken by means of channel observations



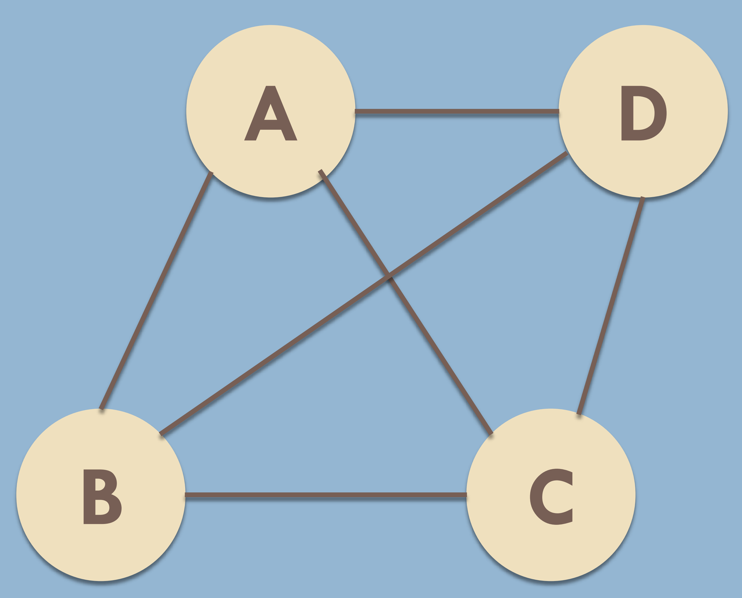
CHALLENGES

How to predict performance of protocols not currently running on the nodes?

- **Virtual execution of protocols on channel traces**
- **Dynamic ranking as a function of wrong/right decisions**



DEMO SETUP

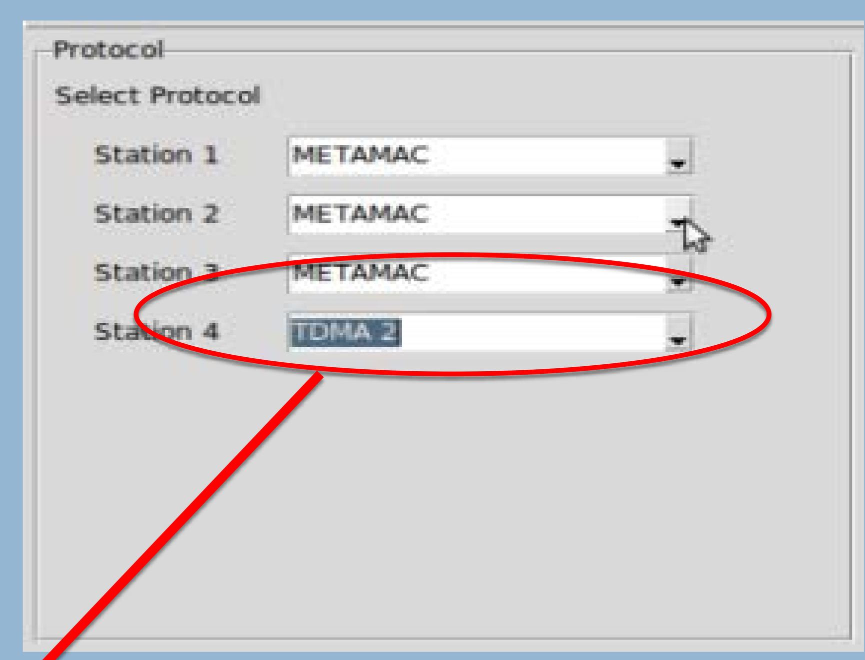


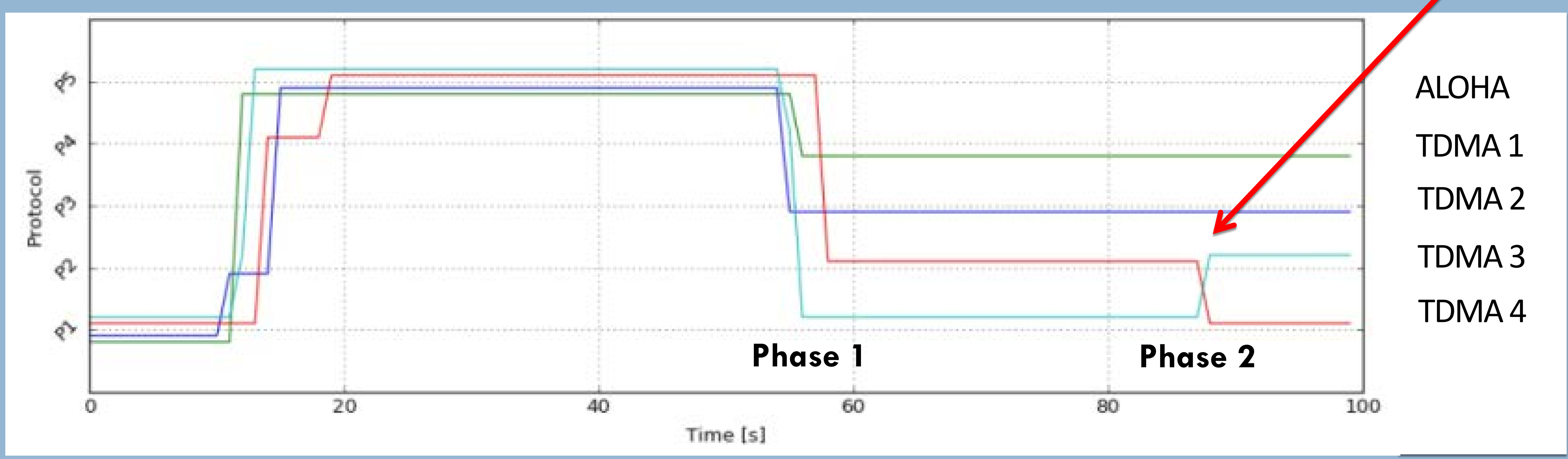
Fully connected simple network;

Protocols : 1 ALOHA and 4 TDMA;

Storyline 1 (Phase1) :

Varying traffic demand and forcing some nodes to a given protocol, we observe decisions at each node;



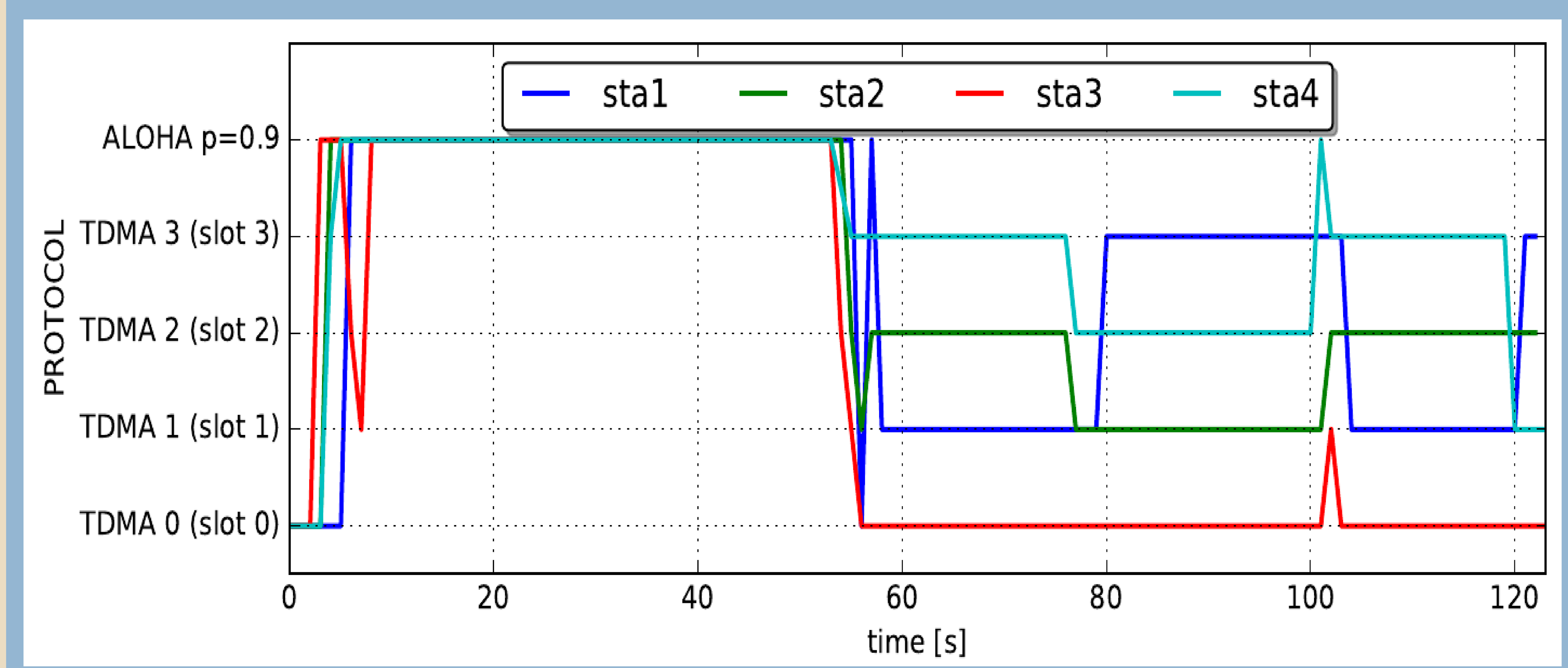


Storyline2 (Phase2) :

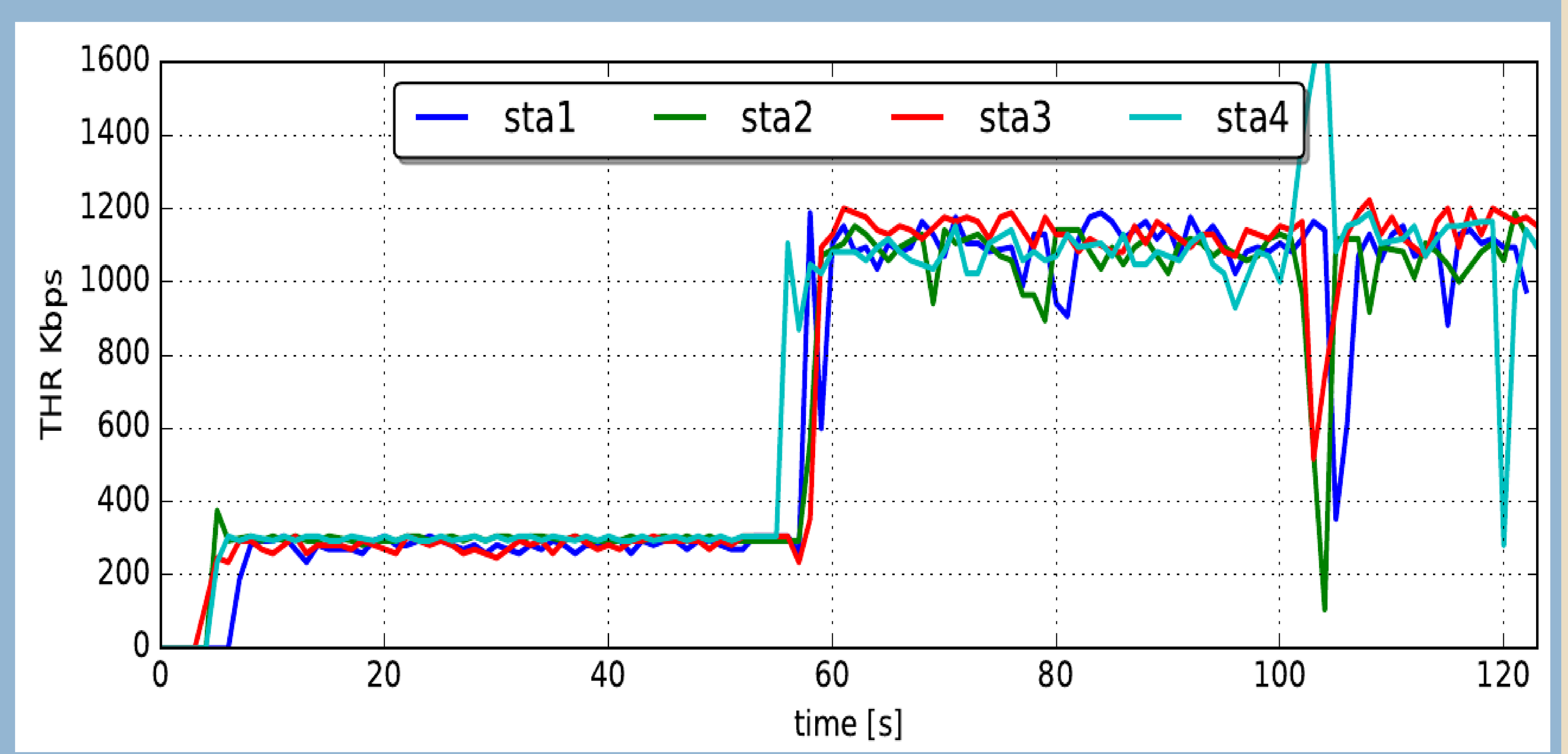
When a node is configured with a fixed slot assignment, the network reacts for finding a new non-conflicting;

RESULTS

Dynamic adaptations in case of traffic and topology changes



Throughput or access delay optimizations



UPI USAGE

UPI_HC	UPI_R
Send the local CP from and report statistics to controller. **start_local_control_program();** **send();** **recv();**	To active a radio program **activate_radio_program**('radio_program_name')
	To set protocol parameters: **set_parameters**(interface, UPI_R.TDMA_ALLOCATED_SLOT)
	To get nodes statistics: **get_measurements_periodic**(params) **Packet queue, Transmitted, Transmitted Success, Transmit Other, Bad Reception, Busy slot**

POST MORTEM

What we demonstrate to other experimenters?

- **A complete cognitive loop for MAC adaptations**
  - **Towards self-adapting networks!**
- **How to use UPI for gathering low-level channel traces**
- **Emulation of MAC protocols based on their formal definition**

PROJECT DATA

Start Date: 01/01/2015; Duration: 36 M
EU Funding: 5.171 M€

Contact:

Ingrid Moerman, imec, Belgium
Email: ingrid.moerman@ugent.be
Web: http://www.wishful-project.eu/