WiSHFUL

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

Year1 Demonstration of Showcases

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

Table of Contents

PORTABLE TESTBED

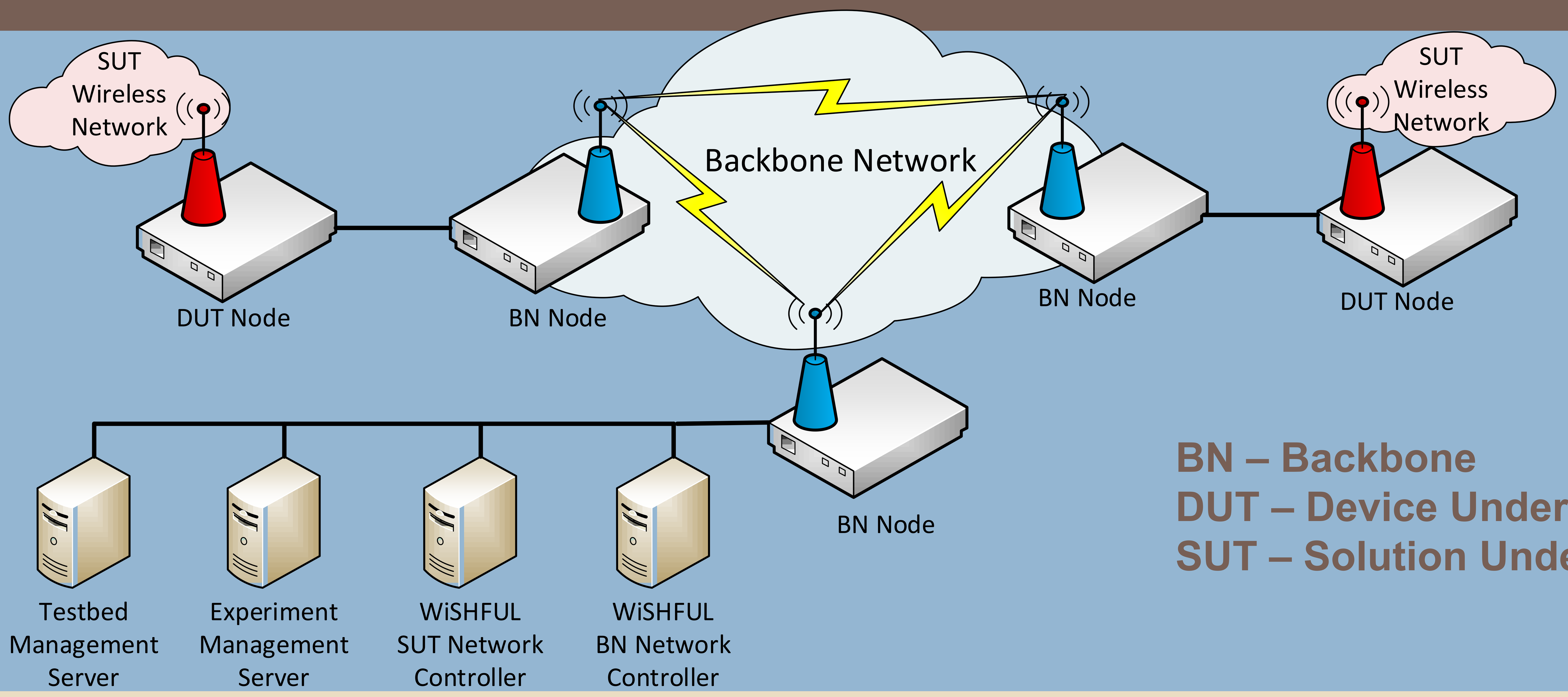
GOALS

- 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 Quality-of-Service
- Experiments in harsh environments and frequent transportation of Portable Testbed – requires robust hardware

ARCHITECTURE OVERVIEW



BN – Backbone
DUT – Device Under Test
SUT – Solution Under Test

FIRST RELEASE OF THE PORTABLE TESTBED

DUT based on Intel NUC D54250 with 3D printed antena mount

FED4FIRE compliance

jFed OMF oml

BATTERY PACKS

Available extensions:

- 802.11a/b/g/n/ac WiFi
- ZigBee sensor nodes
- Bluetooth dongles
- USRP B200mini
- Wi-Spy

WIRELESS BACKBONE

- Provided by nCentric
- Wireless Mesh Network built on 802.11 in Ad-hoc mode with OLSR routing protocol
- Self-Organization features available
- L2 tunneling between DUT nodes – transparency
- Operation of Backbone Network is supervised by WiSHFUL controller, that allows for:
  – changing channel of Backbone Network
  – traffic prioritization in NET and MAC

CONCLUSION

- First version of Portable Tesbted is offered to the research community
- Its operation was validated through showcases
- Software Toolsets are available in the public WiSHFUL code repository – the Portable Testbed can be easily replicated by experimenters

COEXISTENCE OF HETEROGENEOUS TECHNOLOGIES

GOALS

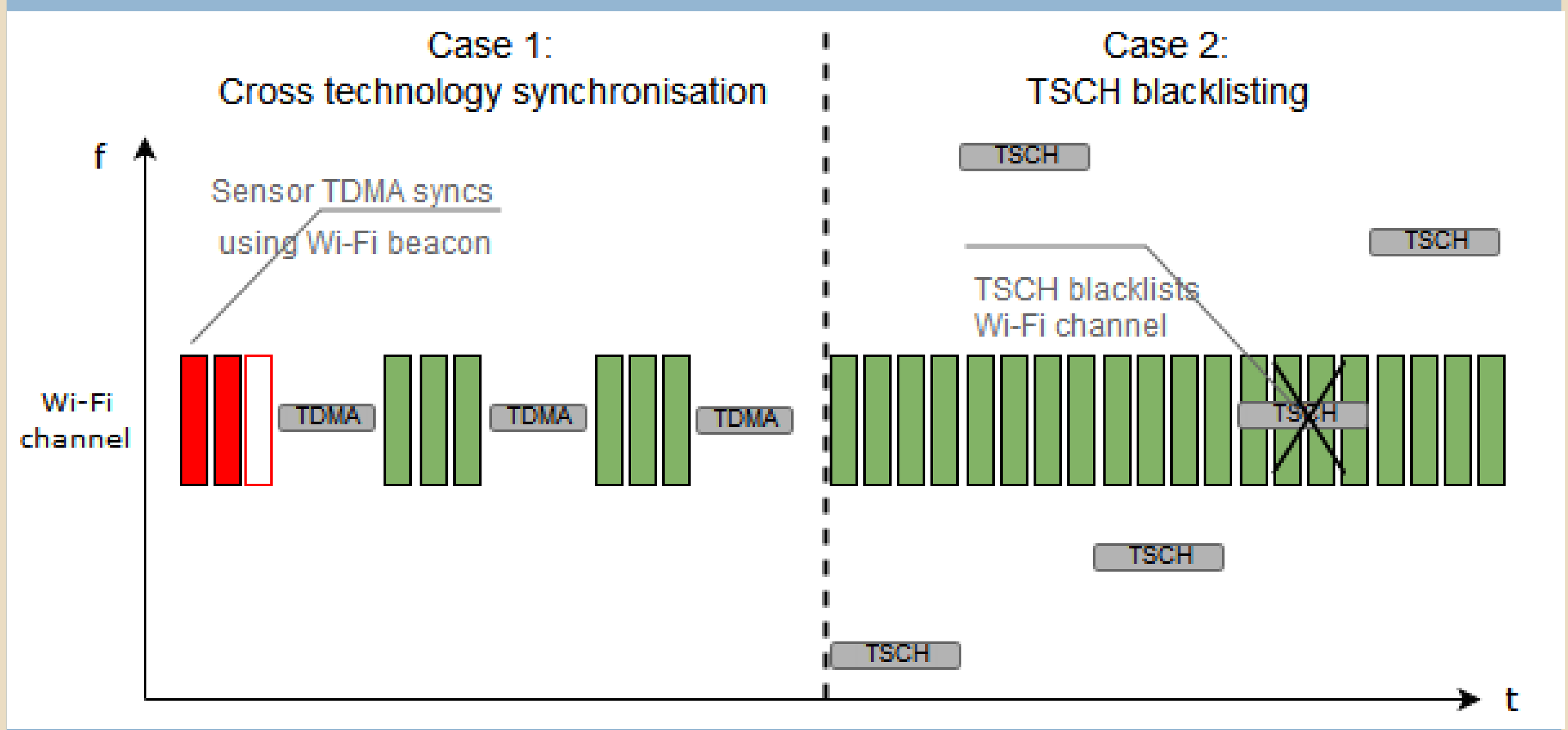
- Enable the co-existence of heterogeneous networks;
- Dynamically adapt and harmonize spectrum allocation across different wireless technologies;
- Devise different strategies for mitigating interference in heterogeneous networks.

CHALLENGES

- Share medium access information (channel usage, time slot schedule, …);
- Blacklist TSCH channels that collide with Wi-Fi channels;
- Provide a mechanism for synchronizing different technologies using cross-technology beacons.

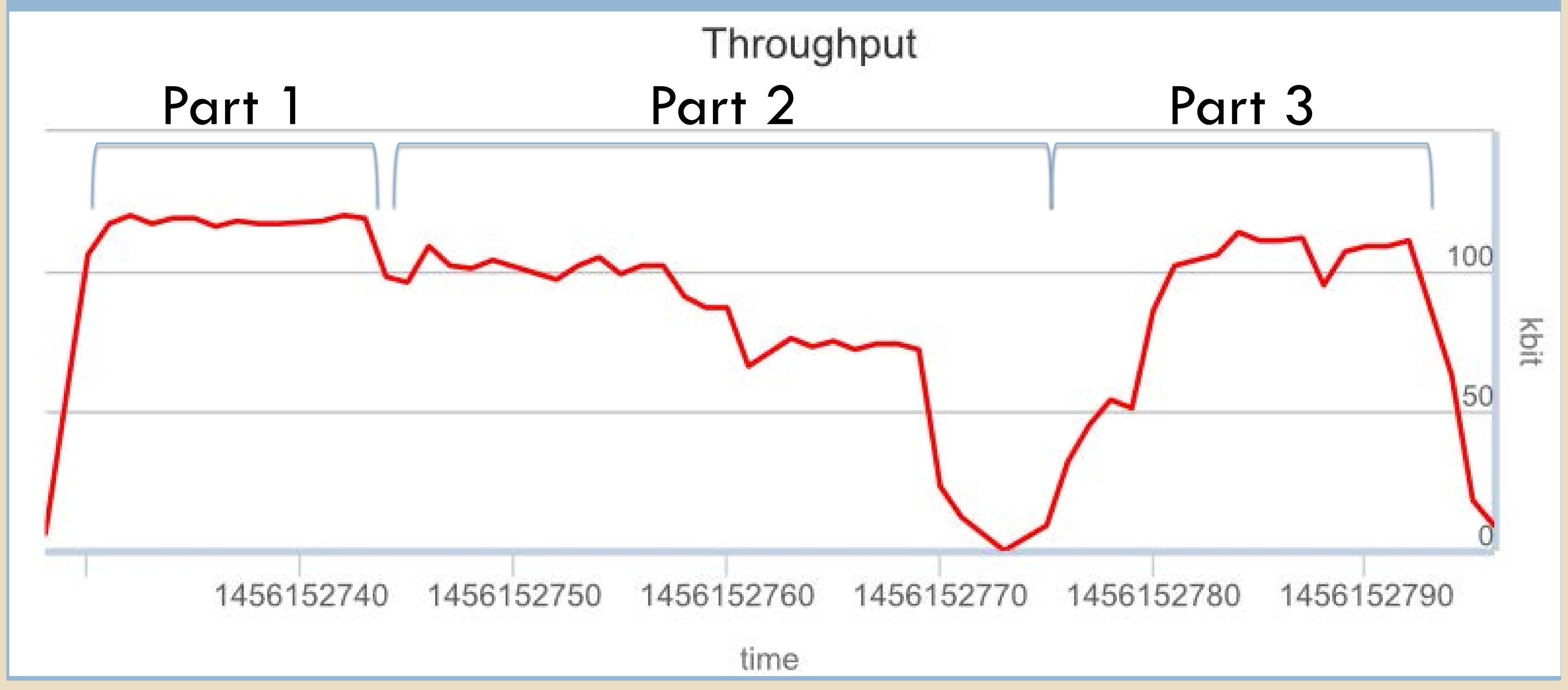
DEMO SETUP

- Part 1: Cross-technology synchronization
- Part 2: TSCH blacklisting

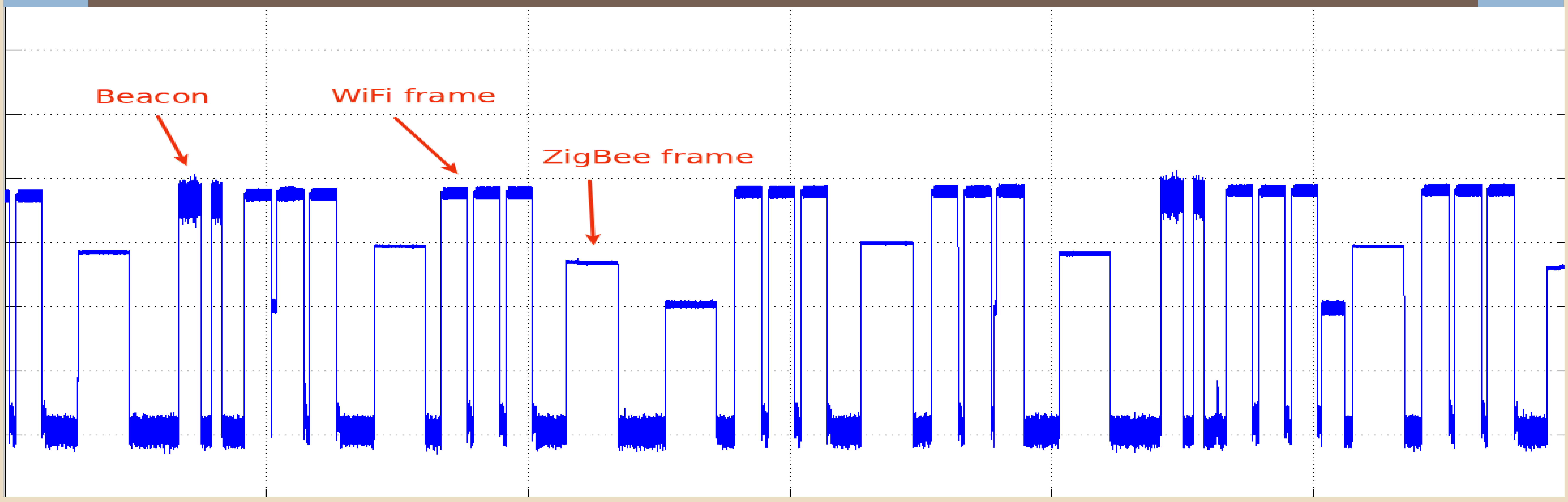


RESULTS

- Part 1: Before WiFi Traffic, using all TSCH channels
- Part 2: During Wi-Fi Traffic, using all TSCH channels
- Part 3: During Wi-Fi Traffic, with overlapping TSCH channels blacklisted



USRP PLOT CROSS-TECHNOLOGY SYNCHRONIZATION BEACON (Y-AXIS = POWER, X-AXIS = TIME)



CONCLUSIONS

- Coherent example of coexistence between IEEE-802.11 and IEEE-802.15.4;
- Cross-technology synchronization succeeded with micro-second level accuracy (+- 5 micro-seconds per superframe);
- Manage to deliver equal amount of bandwidth when Wi-Fi is present and interfered IEEE-802.15.4 channels were blacklisted

POST MORTEM

- Channel hopping enabled Wi-Fi and TSCH experiments using cross-technology synchronization
- Active detection of external interference on IEEE-802.15.4 channels coming from non-WiSHFUL nodes for TSCH blacklisting would be great
- Create a model to detect when the cross-technology synchronization will deteriorate.

LOAD AND INTERFERENCE AWARE MAC ADAPTATION

GOALS

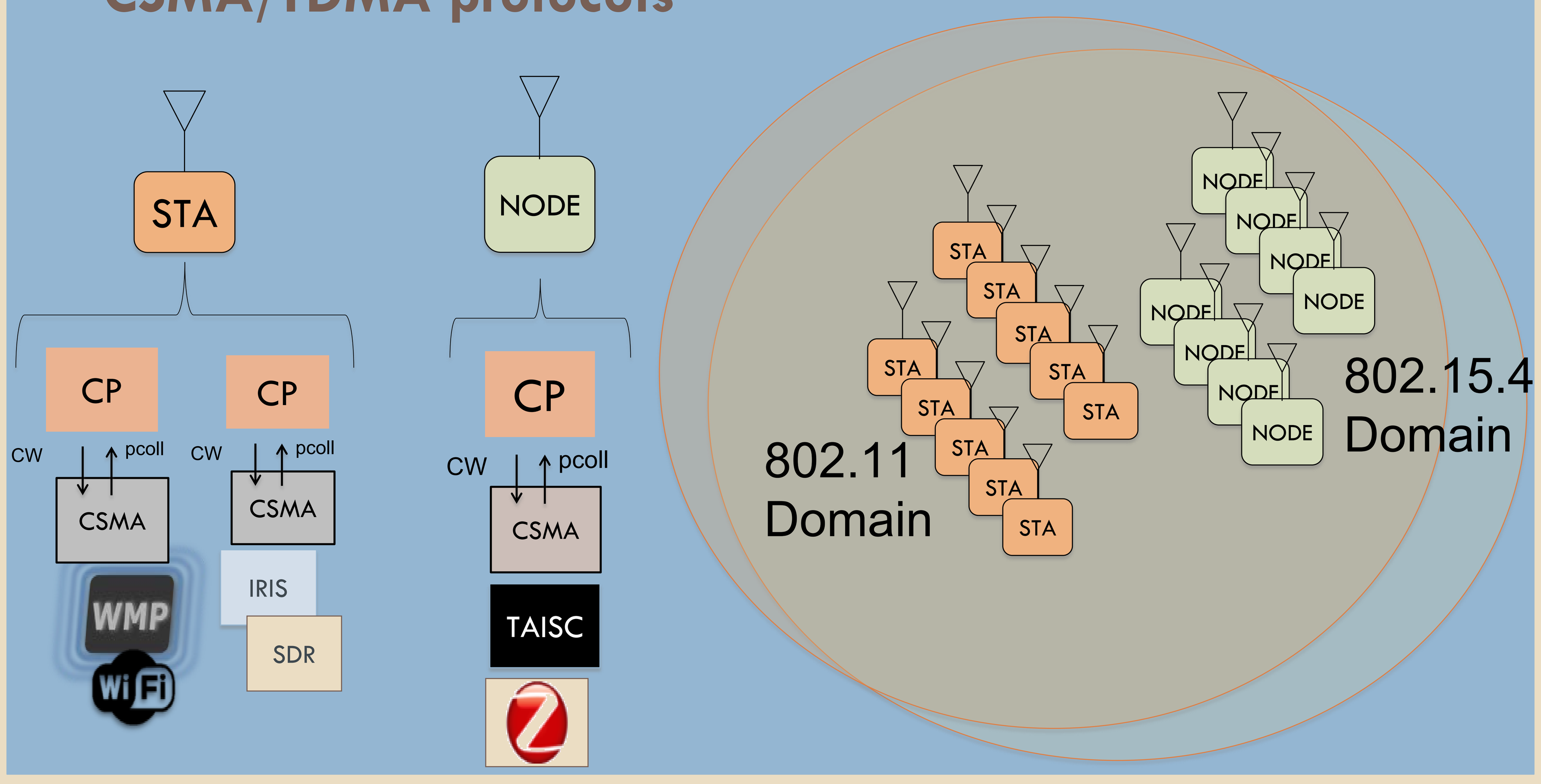
- **According to the network scenario, adapt dynamically the MAC protocol for improving the network performance:**
  - **CSMA** in case of bursty unpredicatble load
  - **TDMA** in case of greedy deterministic load

Challenges

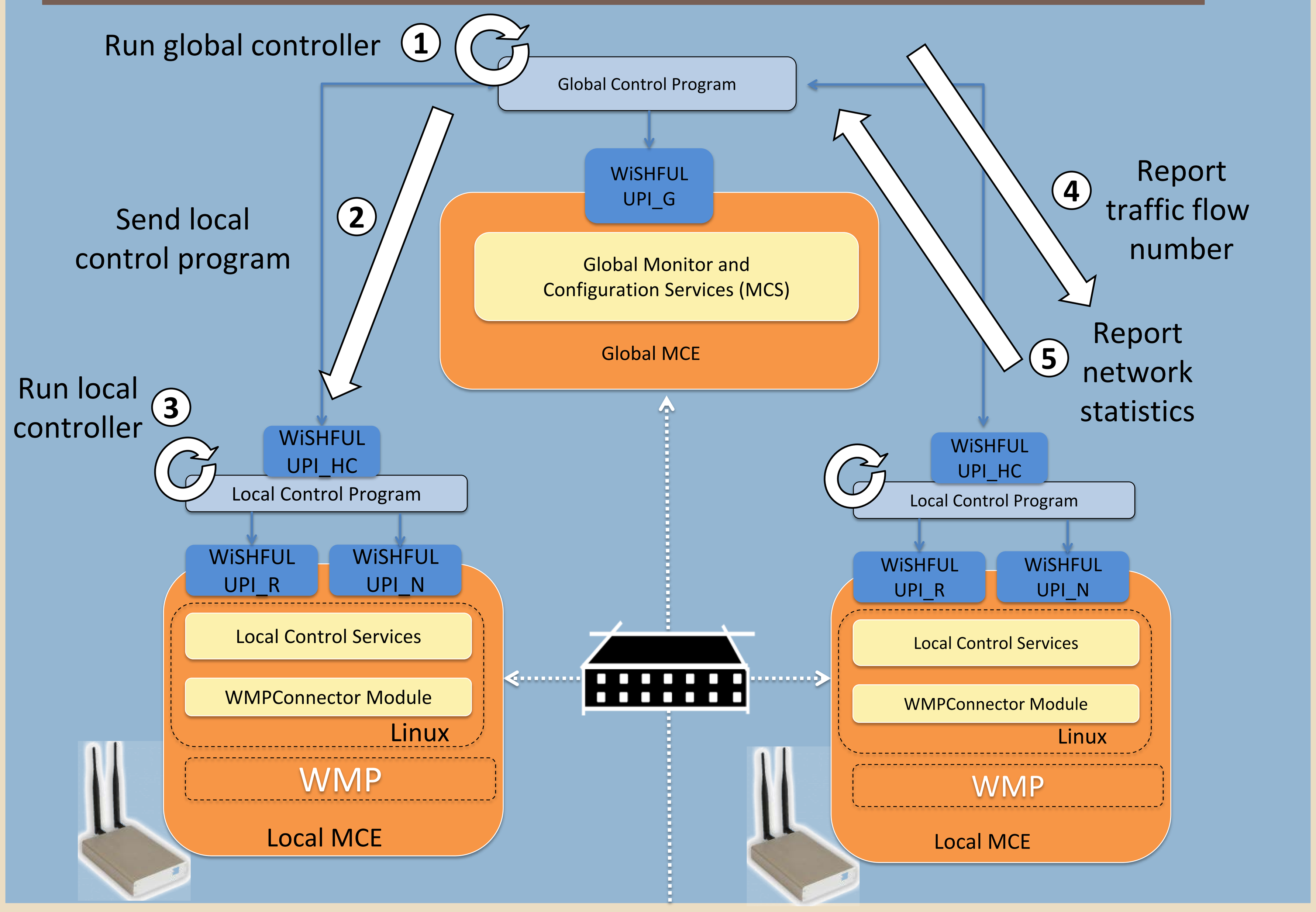
- **Define a flexible and platform-independent control logic, decoupled from protocol operations by exploiting WiSHFUL**
- **Provide an aggregated estimate of the network state**
- **Switch from a protocol to another**

CONTEXT OF THE EXPERIMENT

- **Several nodes work on the same environment and are activated sequentially**
- **All nodes use the same control program for selecting CSMA/TDMA protocols**

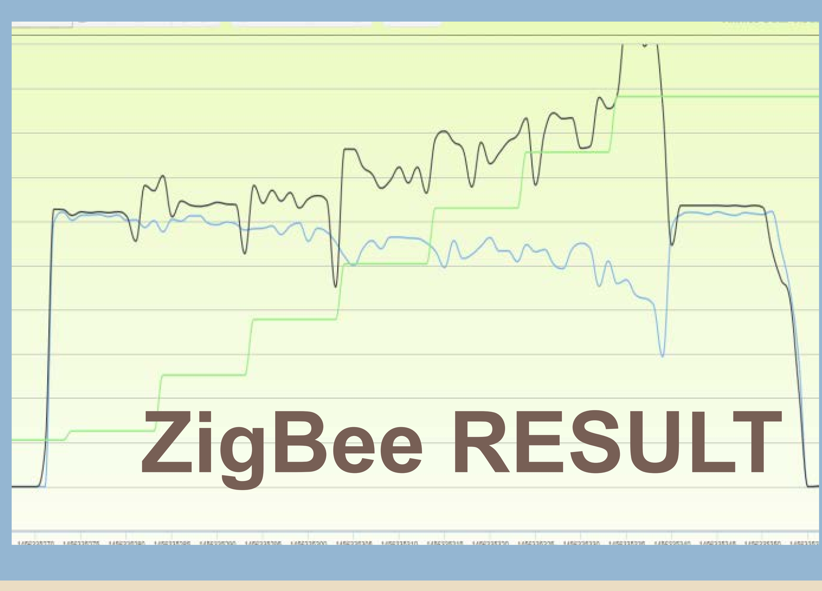


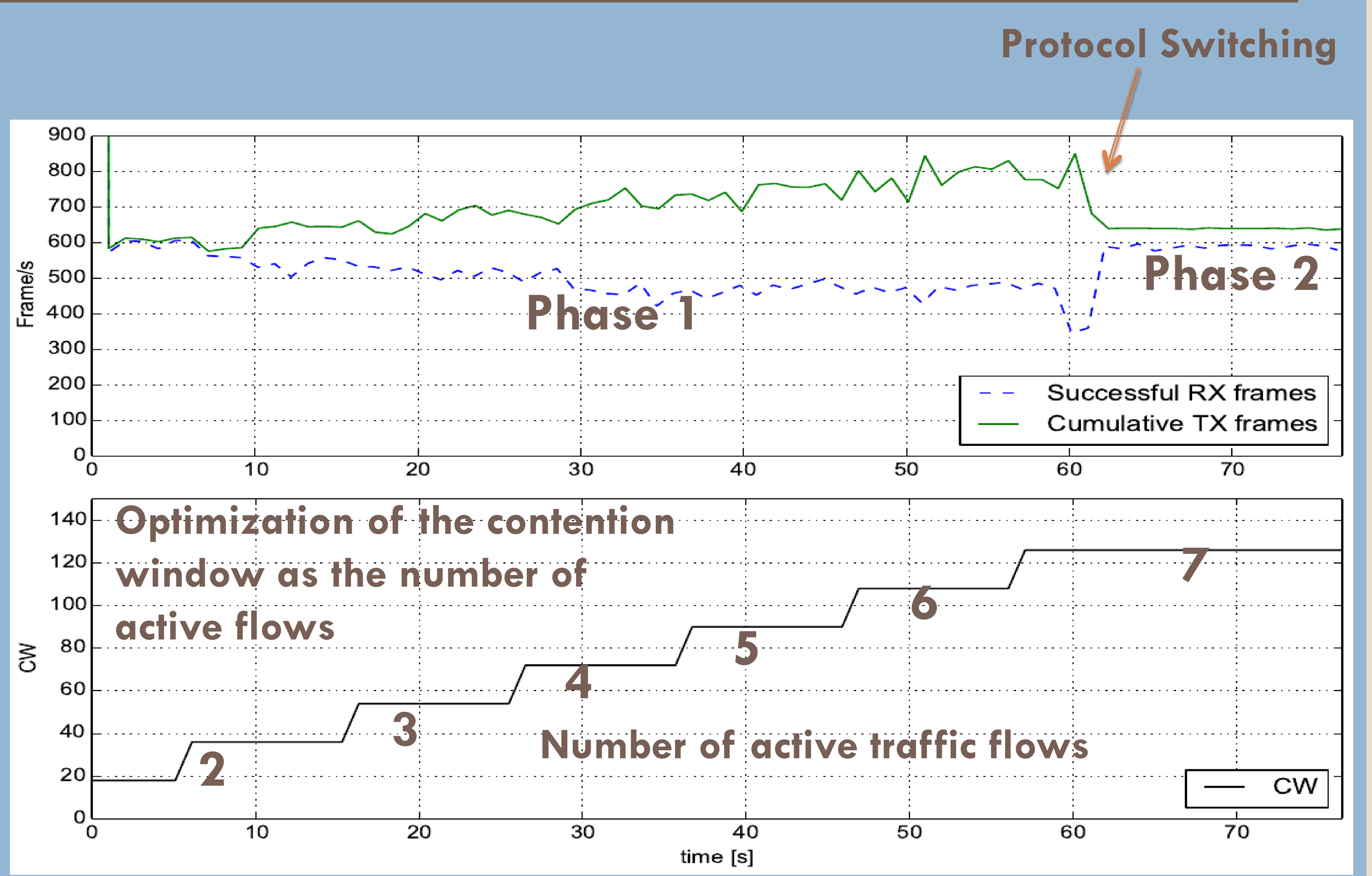
DEMO SETUP



RESULTS

- **Phase 1:** Protocol Adaptation – working on the configuration of the Radio Program (LOCAL)
  - All the nodes run a CSMA protocol, whose contention windows are updated as a function of the statistics gathered from the platform
- **Phase 2**: Protocol Switching – working on the activation of a novel Radio Program (GLOBAL)
  - When the number of traffic flows overcome a given threshold, all the nodes switch to TDMA in a coordinated manner





DETAILED RESULTS

UPI USAGE

UPI_HC	UPI_R
The below functions are used to send the local CP from Global MCE to Local MCE. **runAt();** **ctrlMsgCollector();** **msgFromController();** **transmitCtrlMsgUpstream();** **getMsgFromController();**	To active a radio program on platform **setActive**('radio_program_name' : 'TDMA') To set protocol parameters: **setParameterLowerLayer**('interface' : interface, UPI_RN.CSMA_CW : CWMIN, UPI_RN.CSMA_CW_MIN : CWMIN, UPI_RN.CSMA_CW_MAX : CWMAX) To get nodes statistics: **getMonitor**('interface' : interface, 'measurements' : [UPI_RN.NUM_TX_DATA_FRAME, UPI_RN.NUM_RX_ACK, UPI_RN.BUSY_TYME , UPI_RN.NUM_RX_MATCH])

POST MORTEM

What we demonstrate to other experimenters?

- Interactions between local and global control logic
- UPI utilization for prototyping protocol adaptations and extensions
- Platform-independent control logic
- Flexibility of the approach for defining novel logics
- Example of network protocol to optimize the network scenario

ENABLING EFFICIENT AIRTIME MANAGEMENT FOR IEEE 802.11 NETWORKS

GOALS

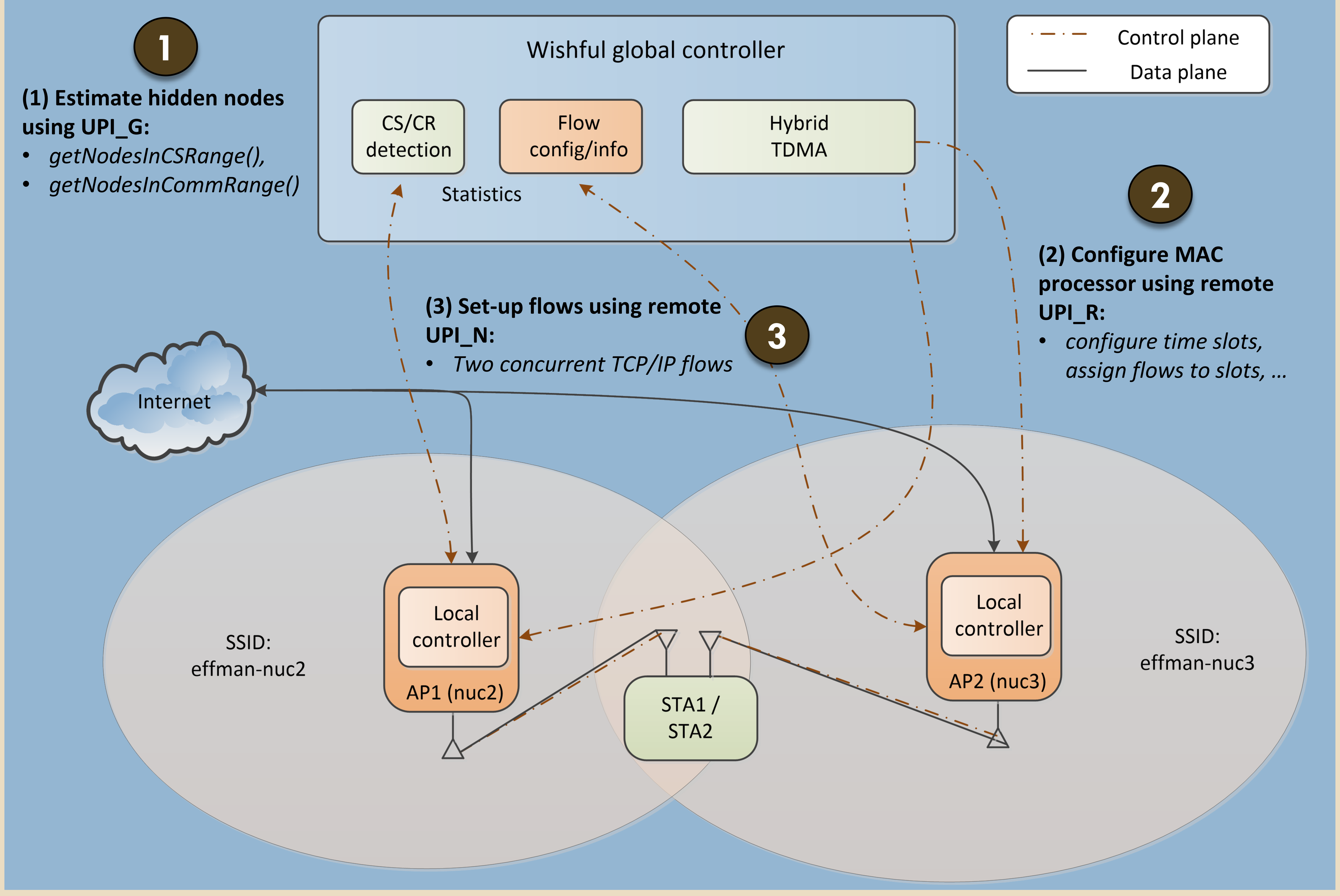
- A widely known problem experienced in IEEE 802.11 (WiFi) networks is performance degradation due to co-channel interference because of hidden nodes.
- The impact can be mitigated by preventing overlapping transmissions (in time) between co-located APs by efficient airtime management through interference avoidance techniques.

CHALLENGES

- The following UPI functionality needs to be provided:

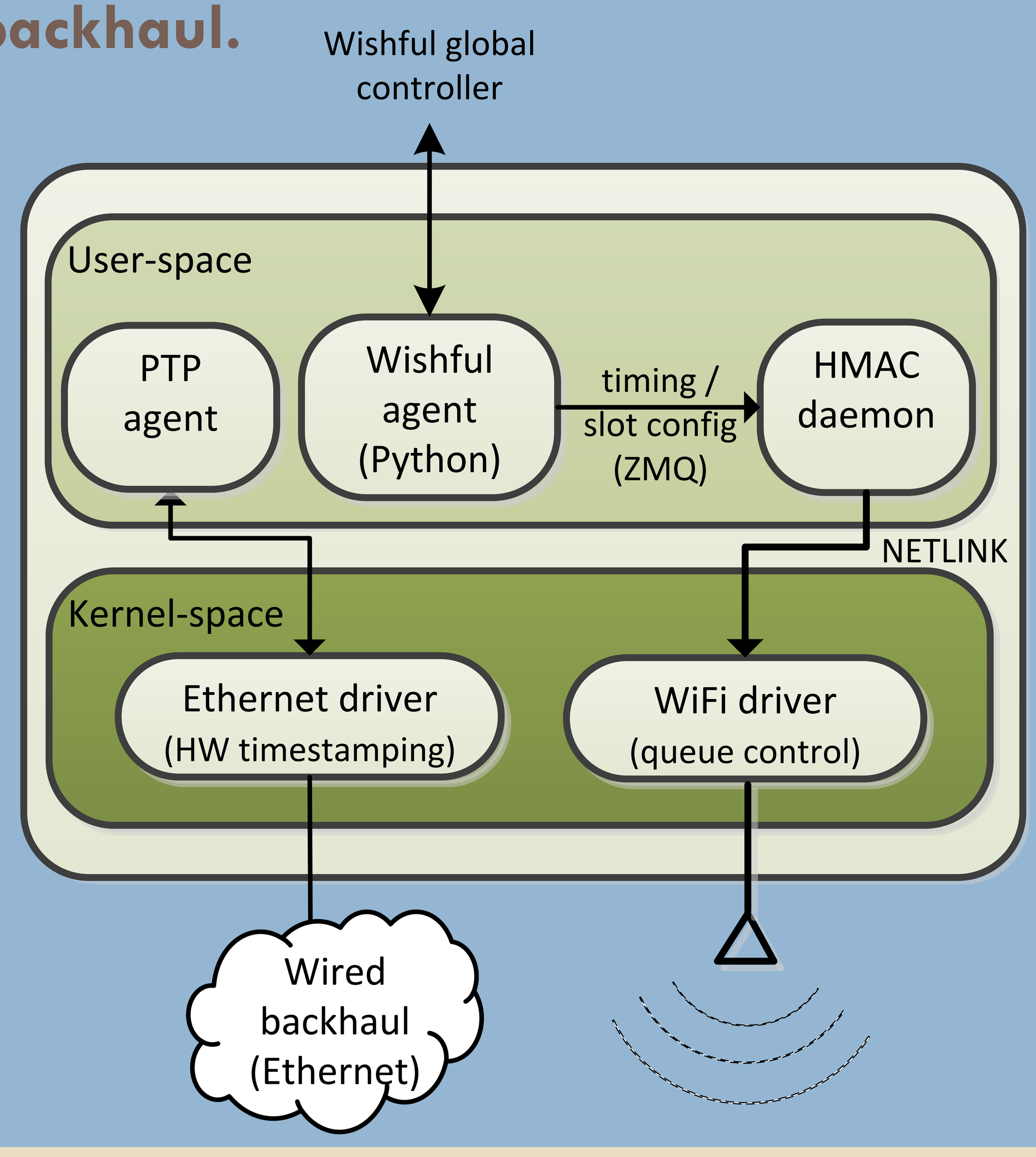
1. Detection of links suffering from hidden node problem,
2. Identification of packet flows,
3. Configuration of TDMA MAC where exclusive time slots are assigned to wireless links suffering from the hidden node problem,
4. Global TDMA MAC requires that all wireless nodes are tightly time synchronized & time slots are aligned.

DEMO SETUP



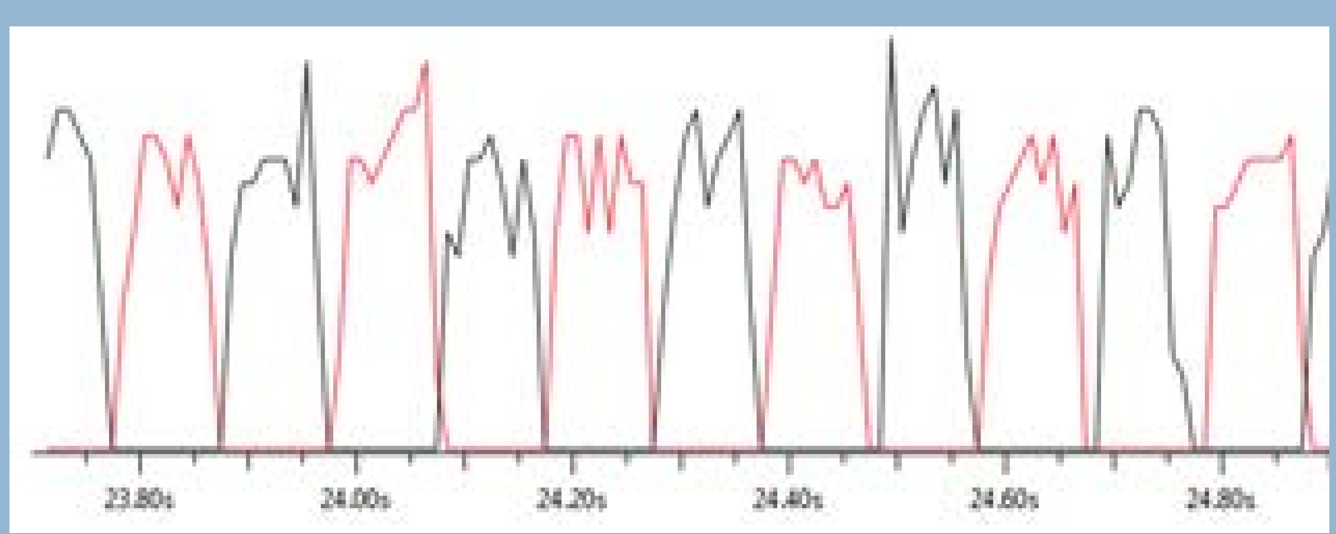
IMPLEMENTATION

- Connector module for Linux platform based on IEEE 802.11 Atheros and open source driver ATH9k,
- Time synchronization over wired backhaul.



RESULTS

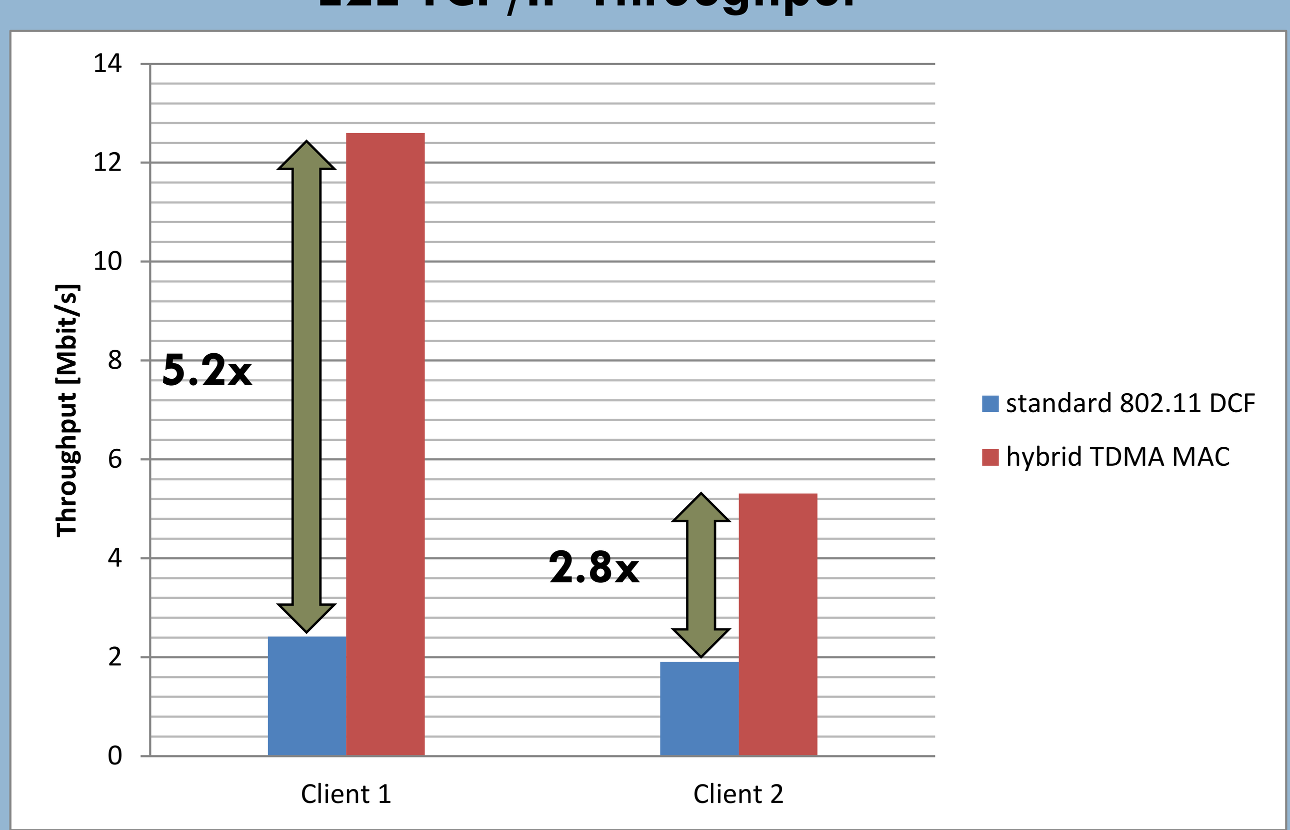
- IO graph - number of packets sent over time



Red = flow AP1->STA1, **Black** = flow AP2->STA2

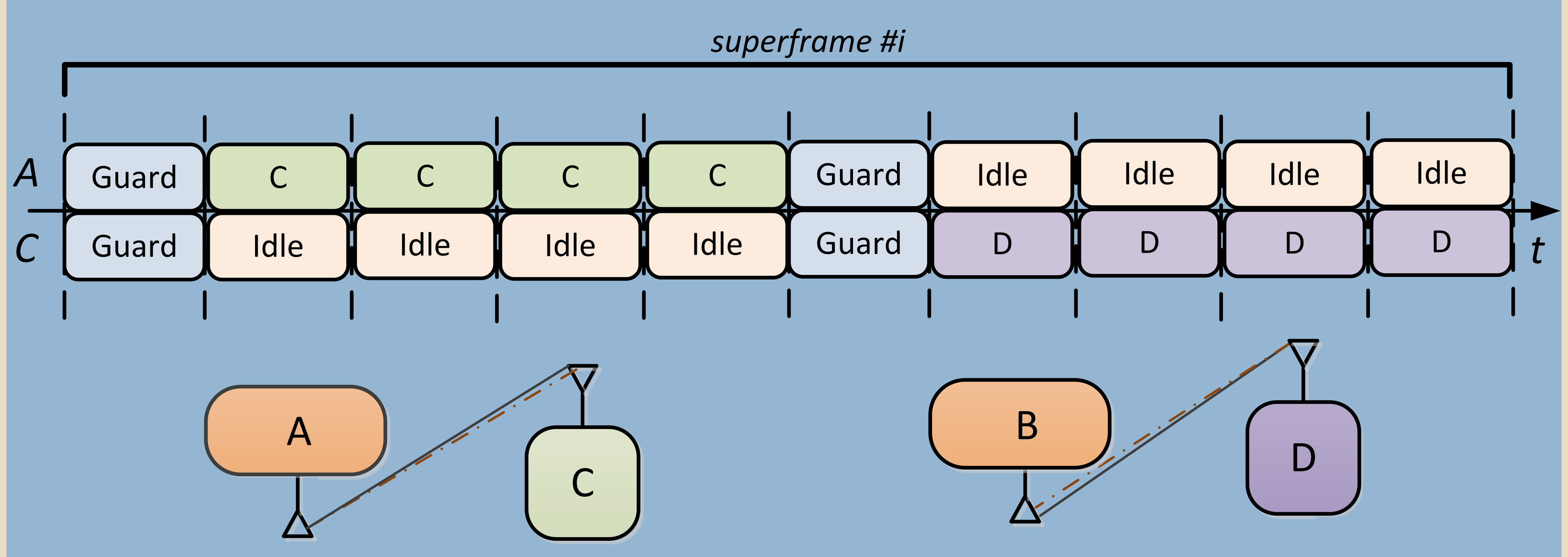
- Comparison with standard 802.11:

E2E TCP/IP Throughput



IMPLEMENTATION

- Global TDMA MAC:



POST MORTEM

- Using Wishful framework an experimenter can easily prototype own algorithms for interference avoidance,
- Provided UPI functionality is already sufficient to tackle another well-known problem, i.e. exposed terminal problem,
- The UPI_R::activate() hides all complexity involved in setting-up & controlling (on a per-flow basis) hybrid MAC layer,
- Additional UPI functions assist the user in making reasonable decisions: i) *getNodesInCSRange()*, ii) *getNodesInCommRange()*

INFRASTRUCTURE-INITIATED HANDOVER FOR IEEE 802.11 NETWORKS

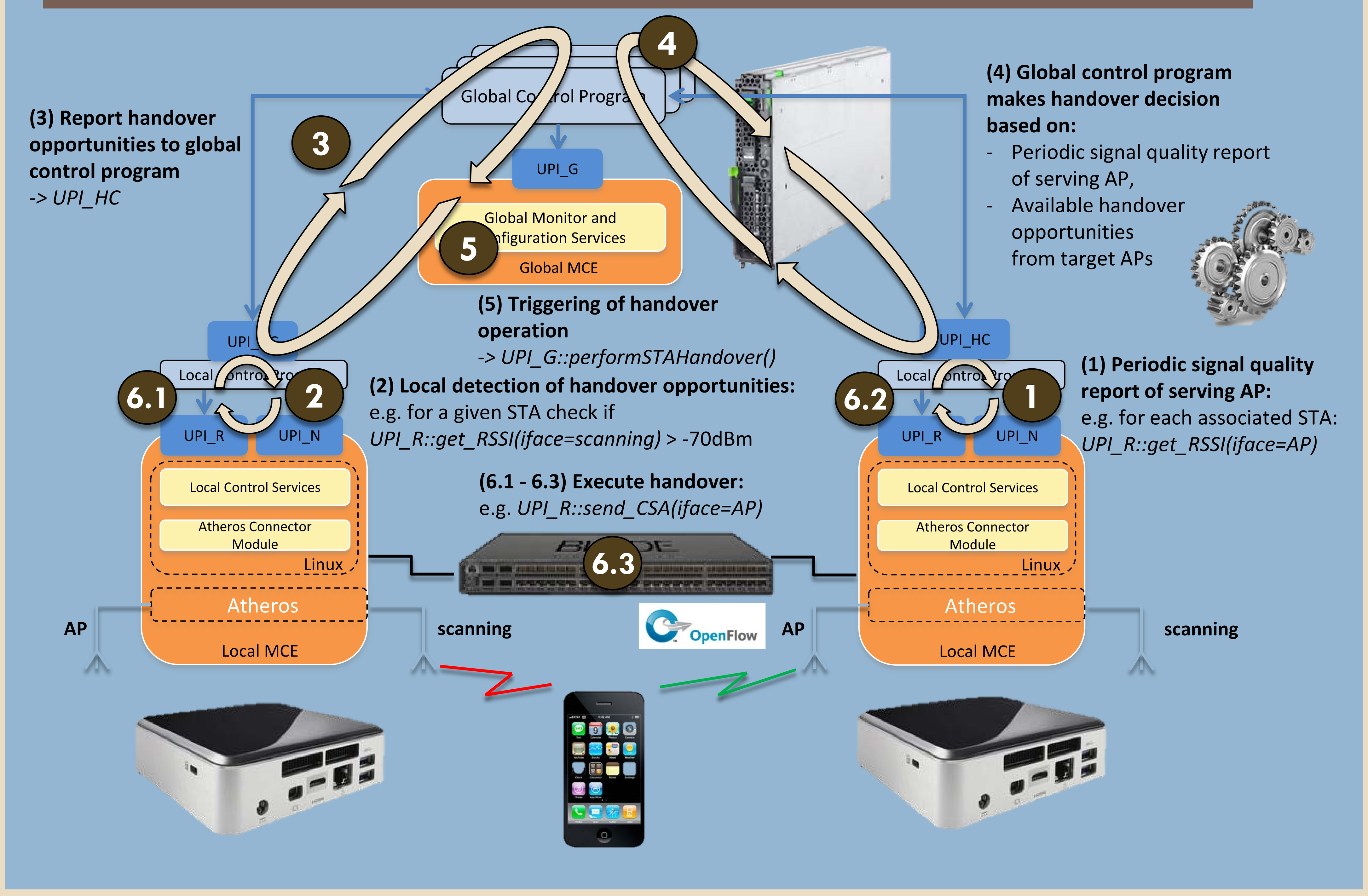
GOALS

- In standard IEEE 802.11 client stations decide on handover (HO) operation using just local information leading to suboptimal performance results,
- An infrastructure-initiated HO scheme removes the STA stickiness by transferring the handover decision from the client to the infrastructure (control plane),
- This will enable the design of novel mobility, client load balancing and interference management schemes.

CHALLENGES

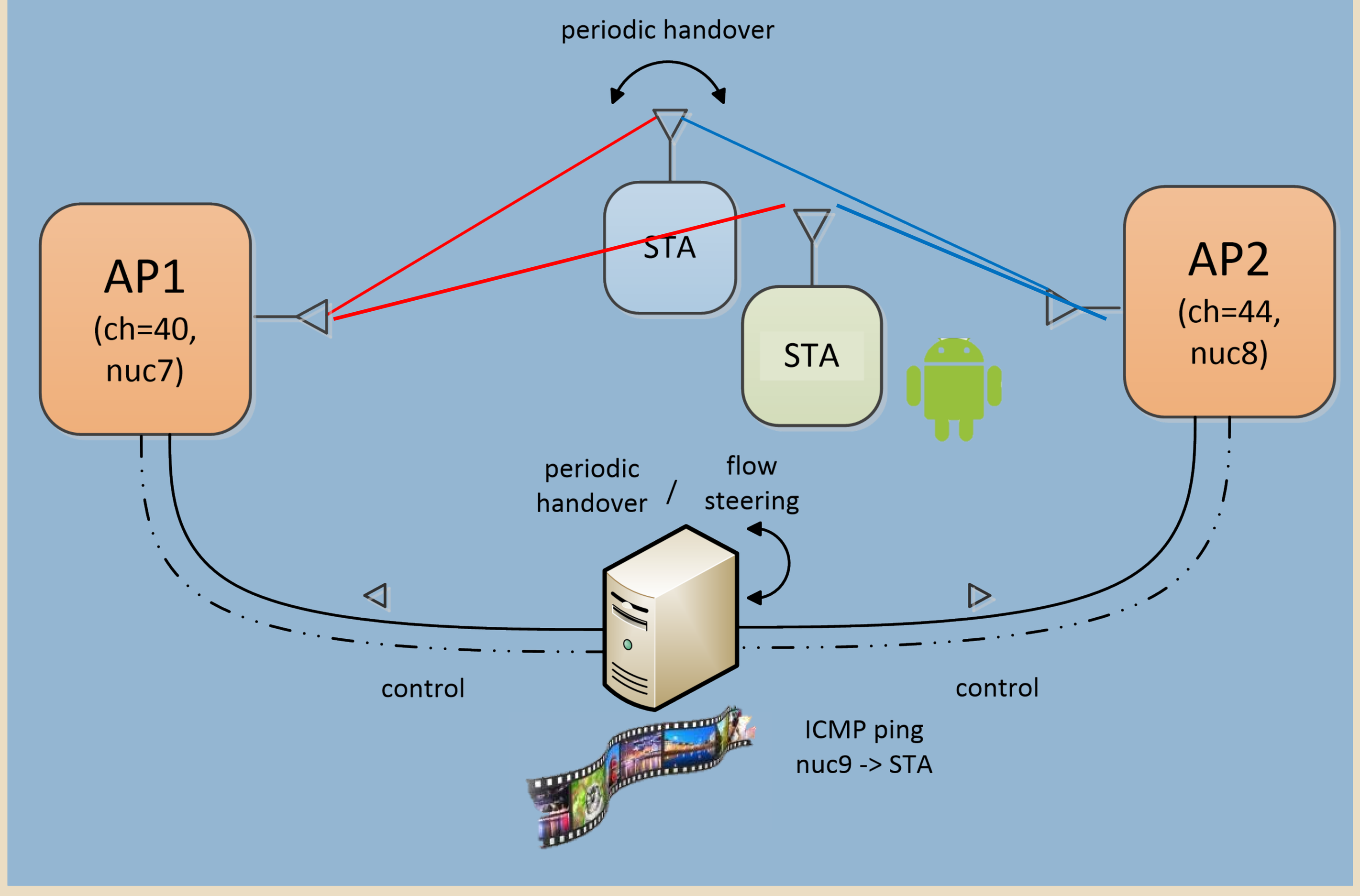
- An experimenter would like to easily prototype own protocols which require an infrastructure-initiated HO,
- Challenge is to find a proper *abstraction* for the WiSHFUL UPI functions, i.e. just a simple API call with all the involved complexity hidden from the experimenter,
- Provide additional UPI functions to assist the experimenter in making reasonable HO decisions, i.e. signal quality of a link.

IMPLEMENTATION



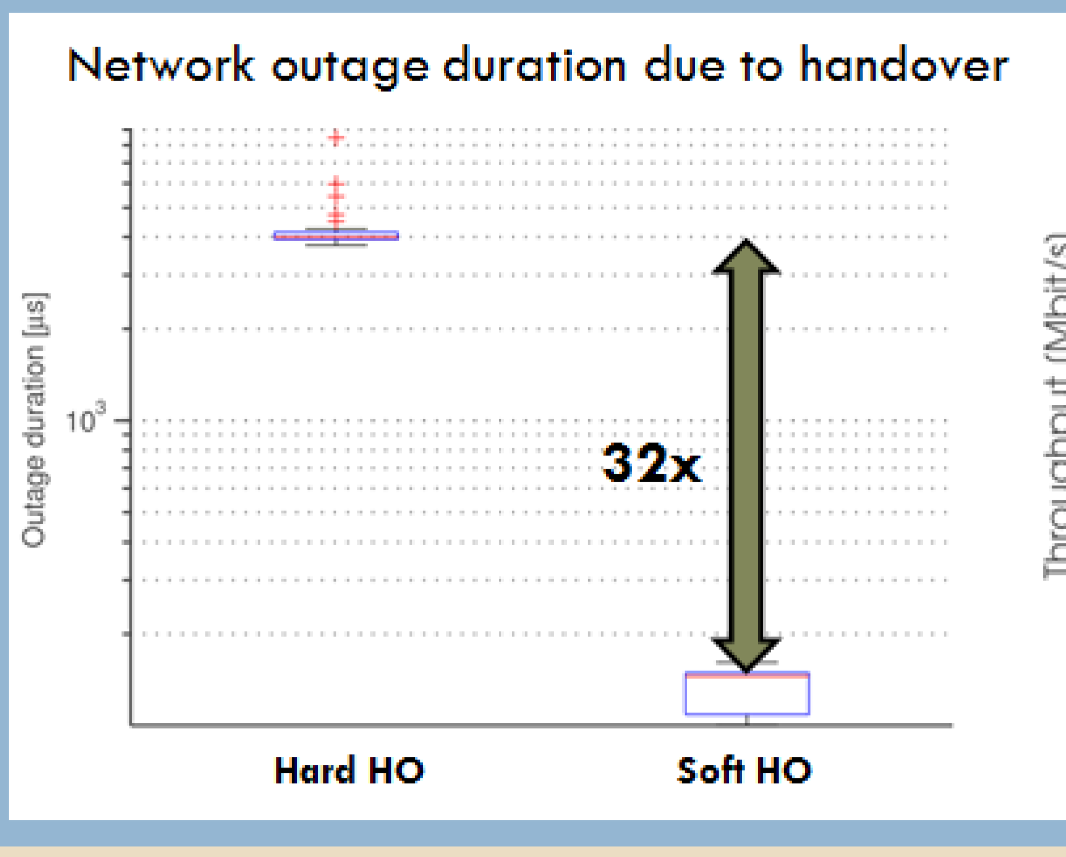
DEMO SETUP

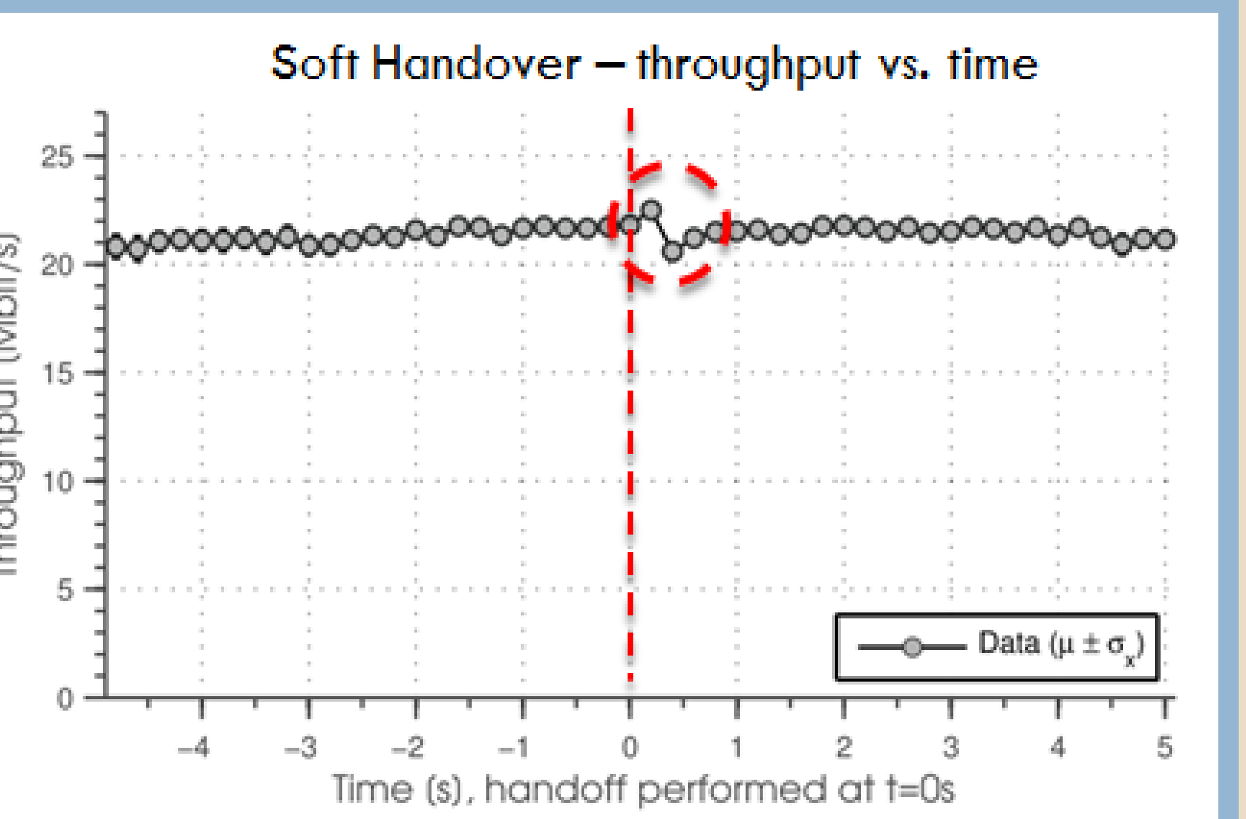
- Periodic handover of STAs between two APs,
- Two different HO schemes were implemented – one STA is served by hard HO whereas the other by soft HO,
- Applications: unbuffered downlink video stream & emulated VoIP flow



RESULTS

- WiSHFUL enables infrastructure-initiated HO for IEEE 802.11 networks,
- Hard HO is suitable for protocols with infrequent handover operations,
- Soft handover provides seamless handover operation but is limited to 5 GHz ISM band,
- Both HO schemes are controlled using the same WiSHFUL UPI function.





Overview of involved UPI Functions

UPI_G:

- performSTAHandover(STA_addr, servingAP, targetAP)

UPI_R (radio/lower MAC):

- sendCSABeaconToSTA(), sendDisassociationToSTA(),
- getRSSI(), getNetworkLoadAtAP()

UPI_N (higher MAC/net):

- registerNewSTAInAP(),
- addSTAToAPBlackList(), removeSTAFromAPBlackList(),
- setARPEntry(), changeRouting() / updateOFTable()

POST MORTEM

- Using the Wishful framework an experimenter can easily prototype own load-balancing, mobility, interference management schemes requiring infrastructure-initiated handover operation,
- UPI performSTAHandover() hides all complexity involved in HO operation,
- Additional UPI functions assist the user in making reasonable handover decision based on: i) signal quality, ii) interference situation, iii) network load.

PROJECT DATA

Start Date: 01/01/2015; Duration: 36 M

EU Funding: 5.171 M€

Contact:

Ingrid Moerman, iMinds, Belgium

Email: ingrid.moerman@intec.ugent.be

Web: http://www.wishful-project.eu/