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

Project Deliverable D2.2

Specification of first showcases

Contractual date of delivery:	30-06-2015
Actual date of delivery:	26-06-2015
Beneficiaries:	TCD, NCENTRIC, TUB, CNIT, IMINDS
Lead beneficiary:	TCD
Authors:	Nicholas Kaminski (TCD), Luiz DaSilva (TCD), Ingrid Moerman (IMINDS), Carolina Fortuna (IMINDS), Vincent Sercu (IMINDS), Mikolaj Chwalisz (TUB), Piotr Gawłowicz (TUB), Ilenia Tinnirello (CNIT), Francesco Gringoli (CNIT), Pierluigi Gallo (CNIT), Giuseppe Bianchi (CNIT)
Reviewers:	Anatolij Zubow (TUB), Peter Ruckebusch (IMINDS), Mikolaj Chwalisz (TUB), Piotr Gawłowicz (TUB)
Work package:	WP2 – General requirements and showcases
Estimated person months:	4
Nature:	R
Dissemination level:	PU
Version:	2.4

Abstract:

This public deliverable provides a detailed description of the first set of showcases to be implemented by the end of Year 1 and according key functionalities that will be demonstrated. This document will serve as a guideline for showcase implementations in WP3, WP4, WP5, and WP6.

Keywords:
glossary, showcases, proof-of-concept

Executive Summary

This deliverable reports on the showcases that will be implemented to display the functionality of the WiSHFUL platform. In support of this effort, this deliverable also defines several key terms for use through the project when discussing the various elements of the WiSHFUL system. These key terms provide a solid basis for the definition of the showcases. Following the development of such a foundation, this deliverable provides an update on the motivation orginially provided in D2.1 on High level requirements for testbeds and software platforms. This update sharpens the focus of the project motivation in a manner that supports the definition of showcases around a central theme. Based on this central motivation, the showcases are presented subsequently. Herein, we focus solely on the conceptual framing of the showcases and their relation to the broader project, i.e. we leave the technical details associated with showcases to the appropriate UPI definition deliverables (D3.1, D4.1 and D6.1). Finally we discuss the display of the showcases, focuses on the venues in which they may be presented to a bigger audience.

List of Acronyms and Abbreviations

AM	Aggregate Manager
AODV	Ad hoc On-demand Distance Vector
AP	Access Point
API	Application Programming Interface
BAN	Body Area Network
CPU	Central Processing Unit
CSME	Carrier Sense Multiple Access
DMT	Discrete MutiTone
DSL	Digital Subscriber Loop
DSR	Dynamic Source Routing
DT	Delay Tolerant
EM	ElectroMagnetic
EMS	Experiment Management Server
EWMA	See Figure 8
F4F	Federation for FIRE (Future Internet Research Experimentation)
FBMC	Filter Bank Multi-Carrier
Fed4FIRE	Federation for FIRE (Future Internet Research Experimentation)
FRCP	Federated Resource Control Protocol
FSM	Finite State Machine
HAL	Radio Abstraction Layer
HTTPS	HyperText Transfer Protocol Secure
I/Q	In phase / Quadrature
IEEE	Institute of Electrical and Electronics Engineers
IP	Internet Protocol
KPI	Key Performance Indicator
LTE	Long Term Evolution
LTE-A	Long Term Evolution - Advanced
LQI	Link Quality Indication
PRR	Packet Reception Rate
MA	See figure 8
MAC	Medium Access Control
MTC	Machine-Type Communications
NOC	Network Operations Center
OFDM	Orthogonal Frequency Division Multiplexing

OLSR	Optimised Link State Routing
OMF	OMF Measurement Library
OML	Orbit Management Framework
PLC	Power Line Communication
QoS	Quality of Service
RAM	Random Access Memory
RF	Radio Frequency
RP	Radio Processor
RSpec	Request Specification
RSSI	Received Signal Strength Indication
RT	RealTime
SDR	Software Defined Radio
SFA	Slice Federation Architecture
SSH	Secure SHell
TCP	Transmission Control Protocol
TDM	Time Division Multiplexing
TDMA	Time Division Multiple Access
TSMP	Time Synchronized Mesh Protocol
TSCH	802.15.4e
UC	Use Case
UMTS	Universal Mobile Telecommunications System (UMTS)
UPI	Unified Programming Interface
UPIR	Unified Programming Interface radio
UPIN	Unified Programming Interface network
UPIHC	Unified Programming Interface hierarchical control
URL	Uniform Resource Locator
USB	Universal Serial Bus
VPN	Virtual Private Network
Wi-Fi	Wireless Fidelity
XFSM	eXtended Finite State Machine

Table of Contents

1 Introduction 6

2 WiSHFUL Glossary 7

2.1 Architecture related terminology 7

2.2 Other relevant terminology 8

3 Central Motivation 9

4 WiSHFUL Showcases 10

4.1 Showcases to be demonstrated at the end of Y1: 10

4.1.1 Efficient airtime management 10

4.1.2 Co-existence of heterogeneous technologies 11

4.1.3 Load and interference aware MAC adaptation 13

4.1.4 Portable testbed 14

4.2 Other showcases: 16

4.2.1 Intelligent Download with WIFI Tethering 17

4.2.2 WiFi Offloading 18

5 Display of Showcases 19

5.1 Venues for Display 19

5.2 Display method 19

6 Conclusion 20

7 References 21

1 Introduction

This second deliverable reports on the initial showcases of the WiSHFUL project as planned in WP2 on *General requirements and showcases* and is aligned with all work package objectives, but specifically targets the last one:

- Definition of relevant and convincing showcases in view of promoting the WiSHFUL capabilities.

This document begins by providing an initial glossary of terms to be used within the WiSHFUL project. Such a glossary is necessary to outline consistent language for use during the course of the project in describing the various aspects of the WiSHFUL platform, functionalities, and interaction with other projects and users. Therefore, within this document, we provide the definition of the terms that will be used throughout the project in describing the work to be accomplished. These terms are based on a conceptual architecture for the WiSHFUL platform and relate to the work of other EU projects (specifically Fed4FIRE) where appropriate. As such, this glossary provides the foundation for the rest of the document as well as the rest of the work to be accomplished within the project.

Following the definition of the terms necessary to discuss WiSHFUL, this document revisits the driving scenarios provided in the prior deliverable 2.1 on *High level requirements for testbeds and software platforms* in order to focus this motivation on a specific attribute that drives the showcases discussed herein.

Based on this motivation, the initial showcases for WiSHFUL are then described. The description given within this document focuses on the definition of the scenario to be addressed and a conceptual approach for applying WiSHFUL functionality within the given scenario; technical details regarding the implementation of each showcase is left to the appropriate UPI centred deliverable(s) (D3.1, D4.1 and D6.1). Moreover, herein we discuss both showcases that are planned for demonstration within the first year of the project, as well as those that are left for future consideration but provide clarification of WiSHFUL functionality. Each showcase is presented following a uniform structure.

Finally this document briefly discusses mechanisms to maximize the impact of the showcases developed within the WiSHFUL project. These mechanisms focus on the selection of appropriate venues for displaying the showcases.

2 WiSHFUL Glossary

In order to ensure a common understanding throughout the project, the consortium has agreed to create a glossary of terminology used during the project. This glossary will be updated and expanded in future deliverables as necessary.

2.1 Architecture related terminology

This section defines the terminology introduced in the high-level WiSHFUL architecture (illustrated in Figure 1).

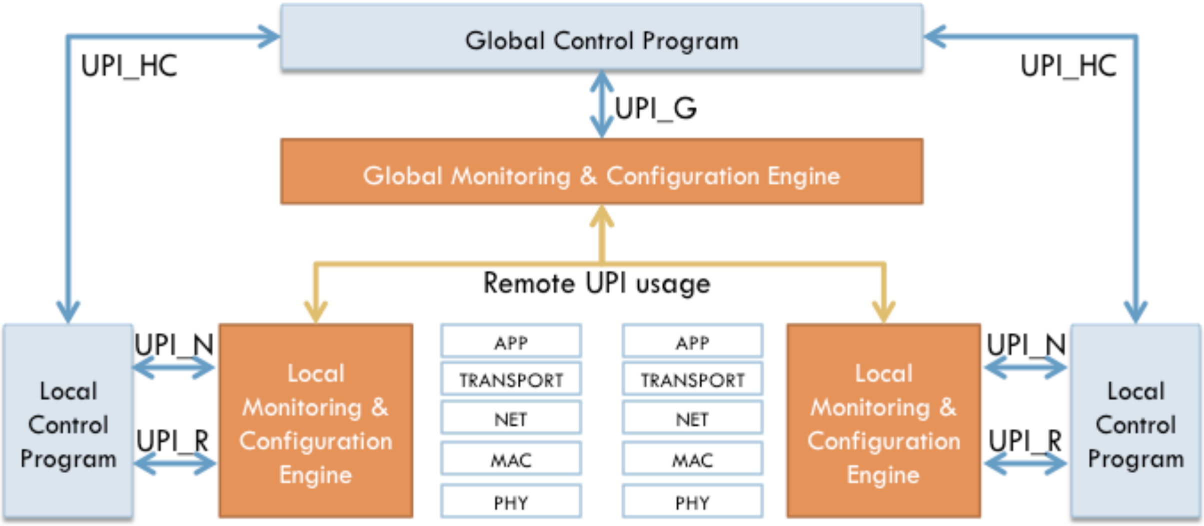

Figure 1. Conceptual diagram of WiSHFUL architecture

Unified Programming Interface – Radio: this is a software interface consisting of a set of functions that ensures uniform control of the radio and lower MAC behaviour on heterogeneous devices. The functions forming the interface are generic, their implementation is hardware and platform specific and is provided by the Local Monitoring and Configuration engine. It is defined by the WiSHFUL consortium.

Unified Programming Interface – Network: this is a software interface consisting of a set of functions that ensures uniform control of the upper MAC and higher layer protocol behaviour on heterogeneous devices. The functions forming the interface are generic, their implementation is hardware and platform specific and is provided by the Local Monitoring and Configuration engine. It is defined by the WiSHFUL consortium.

Unified Programming Interface – Global: this is a software interface consisting of a set of functions that ensures uniform control of the behaviour of a group of heterogeneous devices. The functions forming the interface are generic, their implementation is deployment specific and is provided by the Global Monitoring and Configuration engine. It is defined by the WiSHFUL consortium.

Unified Programming Interface – Hierarchical Control: this is a software interface that enables hierarchical control between control programs that are structured in a hierarchical way. It is defined by the WiSHFUL consortium.

Local Monitoring and Configuration Engine: the role of this engine is to provide a device-specific implementation of the UPI_N and UPI_R. It ensures that the functions defined in the two UPIs execute correctly on the hardware and software platform for which they were developed. It is implemented by the WiSHFUL consortium.

Local Control Program: it is a piece of software that uses the UPI_R and UPI_N and implements the algorithm/logic that controls the radio and network protocol stack and adapts the behaviour of the wireless system to meet the QoS requirements established by end users. It uses locally observed information. Because it uses the UPIs, the code can be compiled and run on several heterogeneous devices within the same device class that support that UPI. Example implementations are provided by the WiSHFUL consortium, most implementations should come from the users.

Global Monitoring and Configuration Engine: the role of this engine is to provide an implementation of the UPI_G that is common to a group of nodes. It ensures that the functions defined in the UPI execute correctly and if necessary simultaneously on the group of nodes for which they were developed. It is implemented by the WiSHFUL consortium.

Global Control Program: it is a piece of software that uses the UPI_G and implements the algorithm/logic that controls the radio and network protocol stack of a group of nodes and adapts the behaviour of the wireless networked system to meet the QoS requirements established by end users. It uses information observed from a group of nodes. Because it uses the UPI_G, the code can be compiled and run in several different deployments that support that UPI. Example implementations are provided by the WiSHFUL consortium, most implementations should come from the users.

Device class: represents a set of devices that are similar in terms of system architecture and capabilities. We consider three classes of devices: 1) microcontroller devices that have a radio chip, 2) general purpose devices with a wireless network interface card and 3) software defined radios.

Upper MAC: The upper-level MAC (upper MAC) is responsible for inter-packet states that are not time critical. This includes among others framing and management functions where some form of negotiation between nodes is required (like association, the allocation of extra time slots, blacklisting channels for hopping sequences).

Lower MAC: The lower-level MAC (lower MAC) directly interacts with the PHY Tx and Rx cores and handles all wireless transmissions and receptions. Minimizing processing latency in the lower level MAC is critical in order to meet the channel access timing requirements. Typical lower MAC functions are: sending, receiving, CCA, back-off, inter frame spacing, CTS/RTS, ACKs, slot synchronization (adjust timing using info in synchronization beacon), next slot scheduling, superframe scheduling, channel hopping, etc.

2.2 Other relevant terminology

Showcase: We use the term showcase to describe an experiment that exhibits the utility of the WiSHFUL platform and mechanisms to address some aspect of the central motivation of the project. A showcase is presented in an easy-understandable and convincing way to a broader community in order to atttract experimenters to use the WiSHFUL platform for their wireless developments.

Experiment: The term experiment refers to the use of research infrastructure to investigate a particular phenomenon. The lifecycle of an experiment can be split in the following phases, initially defined in Fed4FIRE and used in D2.1 of WiSHFUL and slightly adapted here:

- **Resource discovery**: discovery of the facility resources by means of a uniform resource description model for all testbeds.
- **Resource requirements**: detailed specification of the resources required for the experiment. The requirements can be computational, network or storage related, or can even be as specific as the need for software libraries.
- **Resource reservation**: there can be different models for the reservation of resources:
    - No hard reservation or best-effort (use of a calendar loosely linked to the testbed)
    - Hard reservation (once reserved, one has guaranteed resource availability)
    - Future reservation (one should reserve sufficient time in advance)

  - Instant reservation (one can only do instant reservations)
- **Resource provisioning**: two models are defined for the provisioning of resources (e.g. the loading of a new operating system, the configuration of network links):
  - Direct provisioning (API): the testbed provides an API where experimenters can instantiate selected resources.
  - Orchestrated provisioning: a functional component decides which resources are the best matches for the experiment.
  - Advanced configuration: installation of complex software packages (e.g. to be done using tools such as Ansible)
- **Experiment control**: control of the testbed resources during experiment execution. This can be achieved either by using an FRCP compatible tool or SSH. The experiment control can be composed of predefined interactions and/or commands to be executed on resources, either at startup or during the experiment workflow.
- **Monitoring**: a separation is made between monitoring of resources and experiments.
  - Resources: monitoring of the infrastructure health and resource usage (e.g. CPU, RAM, network traffic, availability of resources).
  - The experiment: monitoring of user-defined experimentation metrics of the solution under test (e.g. throughput and packet loss of a wireless connection).
- **Teardown Phase:** During this phase, experimenters tear down the solution under test, performing any necessary garbage collection, shutdown, or final logging procedures.
- **Release Phase:** In this phase experimenters release the research infrastructure used during testing and vacate the test platform.

Testbed: The term testbed is used to refer to the collection of hardware resources provided by a WiSHFUL facility provider that are managed by the same entity. The testbed also includes the software tools, control channels, and other connectivity to support testbed operation (resource discovery, resource reservations, resource provisioning, experiment control and monitoring). This excludes any experimenter developed functionality.

Solution Under Test: This refers to the (wireless) solution and functionality developed by the experimenter and deployed on the testbed resources during the resource provisioning phase of his experiments.

Demonstrator: A demonstrator is a technical example of specific functionality or features of WiSHFUL not necessarily intended for public display. As such, demonstrators do not imply the extra effort towards the development of display mechanism implied by the showcases. Demonstrators may become showcases, if necessary, but are primarily intended as purely technical proof-of-concept instantiations of specifc WiSHFUL functionality and features.

Solution Under Test Configuration Control: This control concerns the setup and initialization of the experiment in terms of the solution under test. This control operates during the deployment phase of an experiment.

Wireless Control Program: This control guides the operation of the solution under test during the course of the operation of the experiment. This program employs the WiSHFUL UPIs to control the infrastructure according to the logic developed by the experimenter. As such, this program operates during the operational phase of an experiment.

3 Central Motivation

As a central motivation for the WiSHFUL project, we consider the emerging wireless ecosystem, which includes heterogeneous technologies, operators, and service providers coexisting in a single environment that features a high-density deployment of wireless devices. High heterogeneity in device capabilities (in terms of spectral bands, coverage, management functionalities, networking models, etc.) combined with limited open, vendor-independent configuration interfaces complicate achieving the often conflicting goals of independent providers and integration of technologies to provide coherent service. In the emerging wireless ecosystem, wireless devices employ multiple radio interfaces, spanning several standards (such as LTE, WiFi, Bluetooth) or offering more esoteric capabilities in the form of programmable interfaces, based on software defined radio (SDR) techniques. Developing techniques suitable for this emerging ecosystem requires services capable of investigating a broad range of technologies in a unified manner.

Specifically, in deliverable 2.1 *High level requirements for testbeds and software platforms* we identified driving scenarios for the project that centred on the concept of increasing density of wireless devices. Within that deliverable we identified three different contexts where such increasing density is especially relevant: body area networks, home networks, and smarts cities. These contexts and the unifying factor of increasing wireless device density provide the central motivation for the showcases of the WiSHFUL project as they provide a relevant and significant use case for the application of the functionality developed within the WiSHFUL project. Specifically within the first year of the project we are focusing on enabling techniques to manage the effects of interference that accompanies such high density scenarios. Each showcases focuses on a different source for such inter-device interference and displays an approach by which the functionality of WiSHFUL supports the mitigation of this interference.

4 WiSHFUL Showcases

Here we provide an initial listing of the showcases that demonstrate the WiSHFUL functionality and platforms. This collection of showcases include those that will be displayed by the end of the first year of the project as well as other conceptual showcases that are useful for defining the functionality of WiSHFUL UPIs.

4.1 Showcases to be demonstrated at the end of Y1:

The show cases listed here are those that the WiSHFUL consortium will demonstrate by the end of the first year of the project. In this way, these showcases will provide the first public demonstration of the WiSHFUL project.

4.1.1 Efficient airtime management

Overview

In dense wireless networks, co-channel interference is a fundamental problem, thus efficient airtime management through parameter adaptation and interference avoidance techniques (e.g. time sharing) is very important. This showcase displays a mechanism for accomplishing such airtime management across IEEE 802.11 WiFi Access Points (AP) based on the functionality of WiSHFUL UPIs.

Goals

- Avoid co-channel interference through adaptation of wireless parameters and efficient allocation of transmission time for co-located APs;
- Dynamically control multiple access points within a network in a coordinated way.

Breakthroughs

In this showcase we demonstrate how can manage the air-time efficiently by solving the hidden node problem using the WiSHFUL UPIs. In IEEE-802.11 (WiFi) networks co-channel interference significantly degrades performance due to packet losses (hidden node problem) and channel contention. Consider the example network given in Figure 2. Let us assume that the two APs, AP1 and AP2, are operating on the same radio channel. In such a case a cell-edge user like STA2 may suffer from interference due to hidden node, i.e. the downlink traffic from AP1 to STA2 will collide with traffic originated at AP2. By solving the hidden node problem, the performance of all nodes in neighbouring wireless networks can be increased.

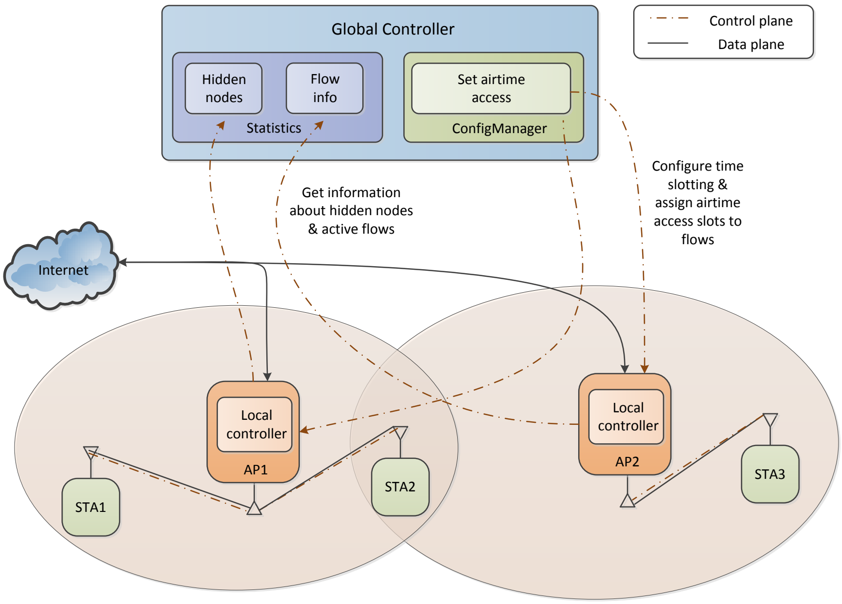

Figure 2 Traffic-aware 802.11 airtime management scenario.

Methodology

We consider a network with four active flows in the following QoS classes – the first three are best effort (BE) while the last one is voice. Each flow is assigned to one of the two APs. Using the WiSHFUL UPI monitoring functionality, we will notice the degredation of quality of service on a single node, which allows identifying hidden nodes and rescheduling flows to avoid interference.

Use of WiSHFUL Functionality

This showcase will primarily make use of UPI_N and UPI_G functionality. Specifically, this WiSHFUL UPIs will be used to monitor the active flows for detecting hidden nodes and to define appropriate channel access patters and time slots for solving the hidden node problem.

4.1.2 Co-existence of heterogeneous technologies

Overview

In dense wireless networks, the co-existence of heterogeneous technologies using the same wireless resources is challenging.

Goals

- Enable the co-existence of heterogeneous networks;
- Dynamically adapt and harmonize spectrum allocation across different wireless technologies;

Breakthroughs

With this showcase, we illustrate how the WiSHFUL UPIs can facilitate efficient spectrum management of co-existing heterogeneous technologies by making them aware of each-other. This will enhance the performance in both networks and make the QoS (throughput, latency, reliability) more predictable. Consider the example IEEE-802.11 (WIFI in 2.4 GHz band) and IEEE-802.15.4e (TSCH) networks illustrated in Figure 3. The simultaneous operation of both networks in close proximity will inevitably lead to performance degradation due to interference. This is because of contention-free explicit scheduling of radio resources in TSCH (timeslotted channel hopping) and the unreliability of carrier-sensing (listen-before-talk) mechanism used in WiFi, are unable to sense any wireless transmission of the other technology. The QoS in both networks can be increased by making them aware of each other.

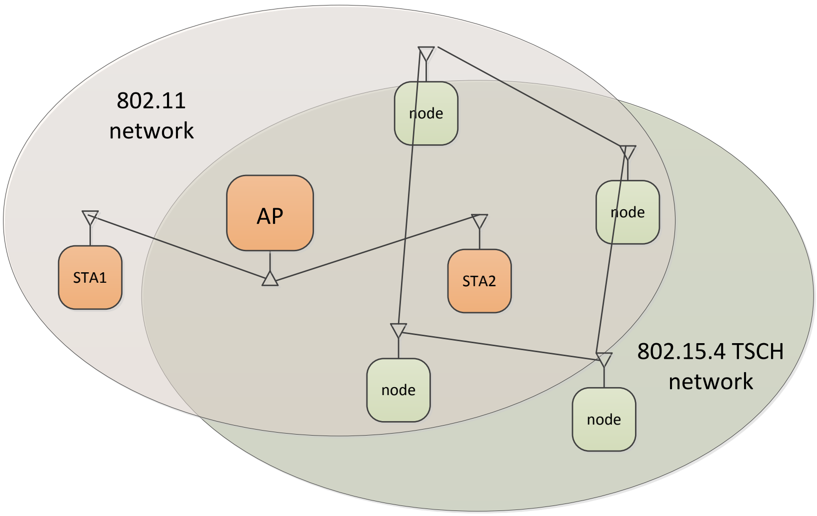

Figure 3 Example illustrating two co-located wireless networks of different technology.

Methodology

One can imagine multiple co-existence schemes for WiFi and TSCH. Some basic schemes can be implemented by only modifying the sensor network. More advanced, and also promising, schemes require cooperation between the networks. In Figure 4 we consider a traffic-aware interference avoidance scheme where, depending on the network load in both networks, other decisions are made.

- In case the sensor network is highly loaded it is more meaningful to perform interference avoidance in the WiFi network, i.e. the sensor network will provide scheduling information so that the WiFi transmissions can be delayed to points in time where no collision with transmission in the sensor network is guaranteed,
- In case the network load in the WiFi is high it is more promising to exclude the spectrum used by the WiFi network from being used in the hopping scheme applied in the sensor network.

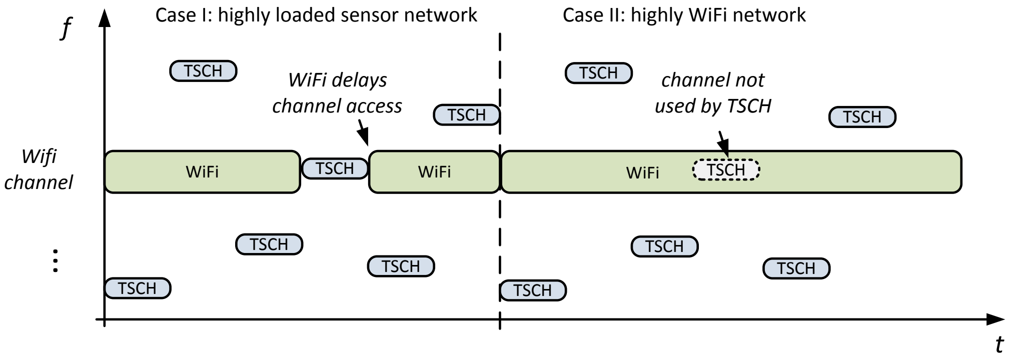

Figure 4. The proposed co-existence scheme for avoiding interference between WiFi and TSCH.

Another approach is to use a cross-technology TDMA protocol to coordinate the transmission between both types of nodes and reduce interference to a minimum. The system runs a TDMA radio program on the WiFi nodes, adapts time slots to traffic requirements, keeps free some slots that are implicitily reserved to TSCH, and uses the remainder for transmission, in order to minimize cross interferences.

Use of WiSHFUL Functionality

In the envisioned co-existence scheme the following functionality needs to be provided by WiSHFUL interfaces:

- Discovery of co-located wireless networks in interference range,
- Information about the network load at each wireless network,
- Information about the MAC schedule in the 802.15.4e TSCH network,
- Possibility to configure the airtime access in the WiFi network,
- Configuration of the spectrum to be excluded from allocation in the 802.15.4 TSCH
- Time synchronization between WIFI and 802.15.4e TSCH
- Tune MAC parameters according with frame size and slots allocated

4.1.3 Load and interference aware MAC adaptation

Overview

A given number of wireless nodes coexist in the same environment with traffic flows that are added dynamically. At first, the nodes run a contention-based access protocol (either a backoff-based protocol with a constant, but tunable, contention window or a persistent access protocol with tunable channel access probability) and adapt the configuration of the contention-based access protocol to support more traffic flows. When the traffic demand gets higher, tuning the CSMA protocol no longer succeeds in supporting more traffic flows. At this point, the nodes need to switch to a TDMA protocol to support more traffic flows. Exogenous interference sources can be active. WiSHFUL provides the means to adapt MAC parameters to handle this scenario and ultimately update the MAC protocol.

Goals

- Detect need for MAC adaptation;
- Tune MAC parameters to cope with changing scenario;
- Swap MAC protocol at the point adaptation is no longer feasible.

Breakthroughs

This showcase will display the ability to use control programs, implementing a cognitive adaptation strategy, across different hardware platforms in order to both fine-tune the contention-based protocol on the local level and to switch to a TDMA based protocol when the traffic demands cannot be supported anymore using the contention-based protocol. In this way we will use a hierarchical control scheme that focuses on adapting the radio functionality. As illustrated in Figure 5, the same control program is used on all device classes and for each technology. This portability for cognitive control is a feature uniquely enabled by the WiSHFUL project.

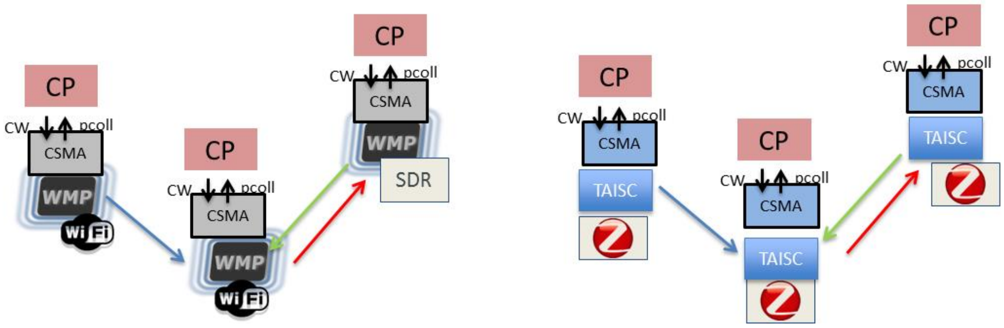

Figure 5 Deployment of a single local program across several platforms.

Methodology

We assume a set-up in which Initially 10 wireless nodes are active using a CSMA MAC with a backoff time of 500 slots. The number of nodes is then increased in steps of 10 until a clear drop in the total number of frames can be detected. To stabilize frame drop rate, the system is tuned by increasing the size of the backoff time to 2000 slots. Then, the number of nodes in the system is further increased until another drop is noticed. At this point, the network is so dense that a switch to TDMA from the previous CSMA is required.

Use of WiSHFUL Functionality

This showcase employs the WiSHFUL UPI$_R$ to tune the MAC layer parameters and UPI$_G$ to swap the MAC protocol across the nodes.

4.1.4 Portable testbed

Overview

Wireless testbeds are imperative for testing innovative technologies such as protocols, hardware, and several other solutions. Many of those technologies will serve in dynamic wireless environments and under challenging conditions. For sake of maintainability and experiment repeatability, however, testbed infrastructure often is fixed. Relocating nodes is difficult since their power supply and network connections are mounted as wall sockets. The testbed environement is thus less dynamic and the conditions are more stable making the evaluation of experimental wireless solutions in testbeds less realistic.

A portable testbed that can be easily deployable on remote, real-world locations is the solution for this problem. All nodes are neatly packed in a sturdy box, complete with their own power supply. An experiment is provisioned while the nodes are still connected through a wired interface, then they can be placed at appropriate locations in the test environment and a wireless mesh backbone is set-up. Afterwards, the normal experiment life-cycle can be followed, making the portable testbed transparent for the experimenter. The portable testbed will also be used on exhibitions for demonstrating the other WiSHFUL show-cases.

Goals

- Develop an easy-to-deploy portable testbed;
- Build a sturdy flightcase, with a sufficiently strong design and easy to wield docking and charging mechanisms;
- Set-up a wireless mesh backbone network to ensure connectivity between the testbed nodes after deployement.
- Make the use of portable testbed transparent for the end-user during an experiment.
- Use the WiSHFUL UPIs to avoid interference between the control traffic in the backbone network and the wireless solutions tested during the experiment.

Breakthroughs

This showcase demonstrates how the fixed testbed infrastructure can be transformed in a portable testbed that that allows to experiment on any location. This process will be transparent for the end-user during experiments. We also demonstrate how the WiSHFUL UPIs can be used to setup a robust wireless mesh network as backbone and to avoid interference between the experiment control traffic and the traffic generated by the solution under test.

Methodology

An example deployment scenario for the portable testbed is illustrated in Figure 6.

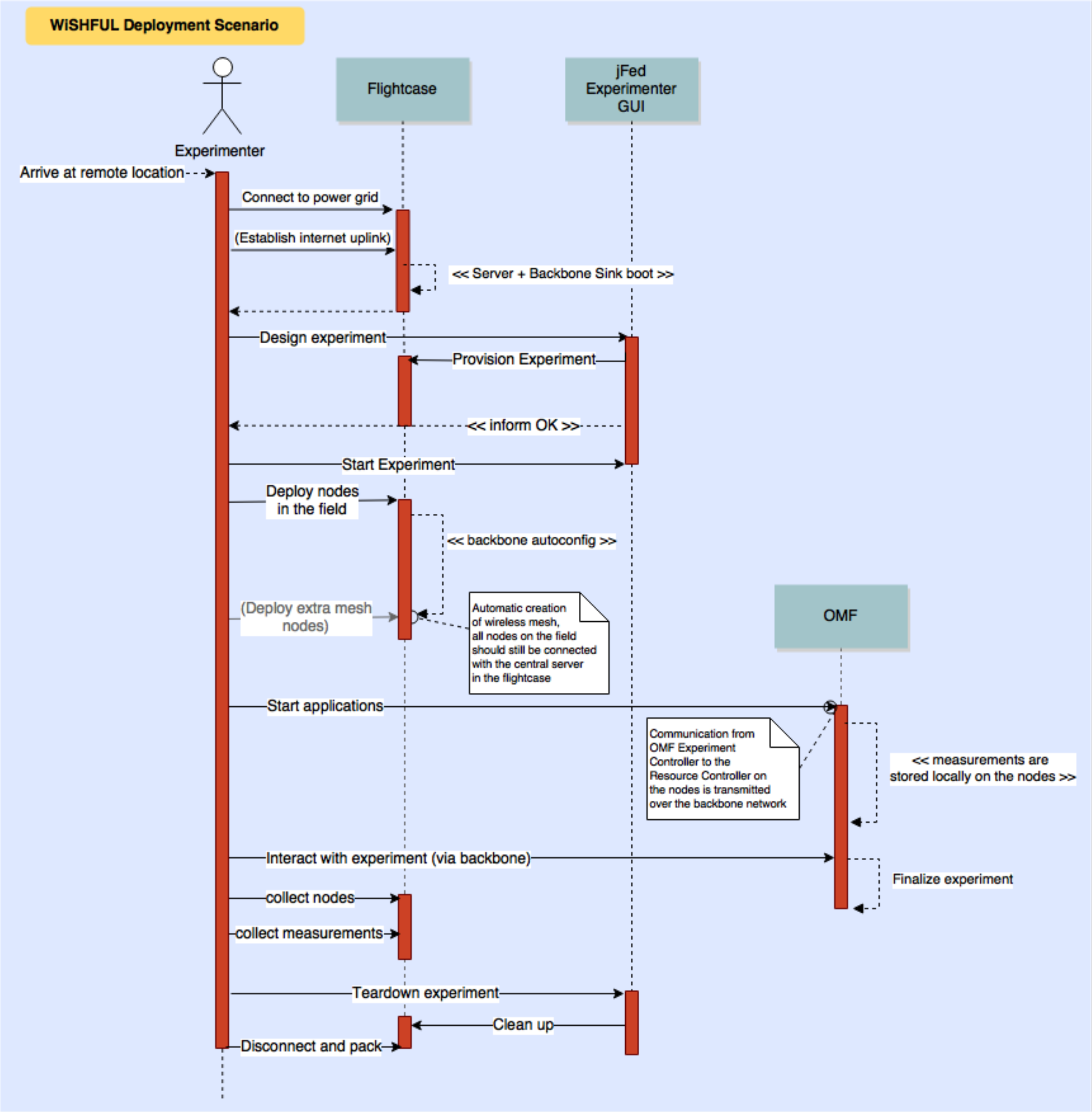

Figure 6 Sequence diagram for the deployment of the portable testbed.

The following steps are required on the portable testbed during experiment life-cycle:

1. When the experimenter arrives at the location, the flightcase is plugged into the power net and the servers and switches inside start to boot. Optionally, the experimenter can connect the switch uplink to the internet.
2. As the servers boot, the backbone also configures itself automatically. It creates a wireless mesh among the nodes.

3. When everything is up and running, the experimenter launches the jFed tool from a laptop that is either inside the flightcase or connected to the central switch. The experiment is designed or loaded from a previous run.
4. jFed will perform the needed actions via the testbed management server and the nodes will be provisioned with the desired software.
5. After this process, the user is informed and the actual experiment is started.
6. The user will deploy all nodes in the field; they remain connected and accessible via the wireless backbone.
7. If there should be a bad wireless link between one or several nodes, an extra backbone node can be added to optimize the mesh network.
8. Via OMF, the experimenter starts his experiment. OMF will make the calls to the nodes over the backbone network. These calls can include (but are not limited to) the setup of a wireless interface, the changing of channels or the starting of an application.
9. While the experiment is running, the measurements are stored locally on the nodes.
10. As the experiment finishes, the experimenter can collect all nodes and properly dock them in the flightcase, physically connecting them again with the core network.
11. The measurements are fetched from the individual nodes and the experiment can be torn down. If the throughput of the wireless mesh network is high enough, or the amount of measurement data is low, the measurement can be transported over the wireless backbone in real time to a database server in the flight case.
12. jFed will ask the central testbed server to clean the nodes up, and the flightcase can be closed and plugged out.

The provisioning and control of fixed testbed nodes is much easier, since all traffic is transmitted over a wired network. Using portable nodes implies that the wired backbone needs to be replaced with a wireless mesh backbone. Moreover, during the experiments, both control and experimental traffic needs to be transmitted on the same medium, posing an interfering challenge that has to be solved.

A wired backbone also has the advantage that it usually runs on dedicated and reliable gigabit links, while wireless communication doesn't. Protocols and communication streams will have to be addressed and compression might be necessary. In a first iteration, the FRCP protocol will be investigated to see if it can be used to do experiment control on the portable testbed.

When the nodes are inside the portable case, there has to be a robust mechanism for docking them. While this mechanism holds them in place, a connection can also be made to both charge the battery and foresee the nodes with a wired connection (PLC). This mechanical challenge is thus extended with an electrical one: making sure Ethernet communication can flow over the charging pads.

More implementation details of the portable tested can be found in D6.1.

Use of WiSHFUL Functionality

The portable testbed itself is an aspect of the functionality provided by the WiSHFUL project. Furthermore the WiSHFUL UPIs will be used to set-up the wireless mesh backbone network and to avoid interference between the experiment control traffic and the traffic generated by the solution under test.

4.2 Other showcases:

Additional showcases that are useful for defining the WiSHFUL UPIs are listed here. These showcases will not be expressly targeted for demonstration within the first year of the project.

4.2.1 Intelligent Download with WIFI Tethering

Overview

Recently, with rapid growth of number of smartphones and mobile devices equipped with various wireless technology interfaces, tethering popularity gains more and more popularity. It is a very convenient, ad-hoc and low-cost wireless Internet access technology. In most cases Wi-Fi tethering is used, which allows sharing the Internet connection provided by 3G/4G technology with other devices (eg. laptops) using Wi-Fi network.

However, in most cases, there is a limit on amount of data that cellular subscriber can download every month. After exhaustion of available data transfer, user connection can slow down significantly or in worse case, one can be charged for any extra data he/she downloaded. As long as the user is aware of all his/her network transmissions, there is no problem. Unfortunately, it can happen that operating system and applications will perform upgrades and download a huge amount of data that user may not even notice until he/she gets a bill from telecom company. Our idea is to recognize and prevent any "unnecessary" traffic flows. By "unnecessary", we understand flows that require downloading a huge amount of data and there is no problem to defer it to a later point in time. Operating system updates (e.g. Windows/Linux) are perfect example here.

Goals

- Intelligently control use of WiFi network;
- Filter download requests based on their context, including connectivity situation and content priority.

Breakthroughs

Our basic idea is to filter out "unnecessary" traffic flows when being connected to a tethering AP (Figure 7). For this purpose, we can make use of IEEE 802.11u that defines Generic Advertisement Service. GAS is mechanism that delivers information to the STA from advertisement services. It allows stations to obtain information about network services. The standard defines a number of advertisement protocols that can be used with GAS: Access Network Query Protocol (ANQP), Media Independent Handover (MIH), Emergency Alert System (EAS) as well as proprietary vendor specific protocols (which will be most useful for us). The GAS mechanism allows the STA to know in advance the AP capabilities, even before associating with it, which is an important feature for this showcase.

Methodology

With the help of GAS we are able to block the "unnecessary" flows already on the WiFi end-user terminal (e.g. laptop). After reception of the specific IE (information element) from the tethering AP, the terminal should translate it into local firewall filtering rules and apply them. One possible way to achieve that is use of netlink interface and netfilter framework provided by Linux (used by iptables). Note, here both the tethering AP as well as the WiFi end-user terminal need to be WISHFUL-compliant. In a second option, which does not require the end-user terminal to be WiSHFUL-compliant the blocking of "unnecessary" flows is performed in the tethering AP which is fully transparent to the end-user terminal.

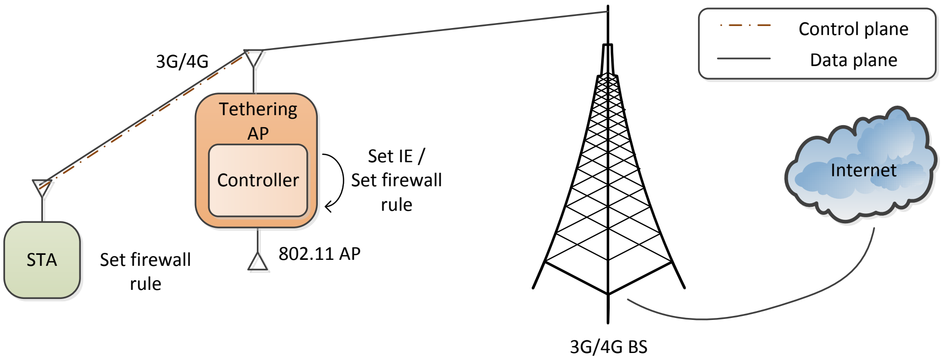

Figure 7. Controlling WiFi tethering operation.

Use of WiSHFUL Functionality

For the first option, the WiSHFUL UPIs will provide a way to program the Information Elements (IE) send in the beacon frames of the tethering AP. Moreover, on the end-user terminal, the WiSHFUL UPIs will offer functionality to read the received IEs, as well as to program the firewall, i.e. reject all outgoing traffic to a specific remote host.

For the second option, the WiSHFUL UPIs will allow the tethering AP to detect and block "unnecessary" flows.

4.2.2 WiFi Offloading

Overview

Although the capacity of cellular networks constantly increases thanks to technological enhancements, the throughput they provide can turn out to be insufficient, because traffic demand increases even faster. On the other hand, most of mobile devices are not only equipped with LTE interface, but also with a WiFi chip. It is therefore promising to offload traffic from mobile networks to WiFi and use them as an extension to the cellular network. This is one of the features in 5G that gains a lot of interest by telecom operators. The main reasons for this approach are the high data rates and availability of WiFi networks in urban environments.

Goals

- Enable offloading of cellular traffic through WiFi networks;
- Empower network performance aware techniques for network selection.

Breakthroughs

Currently, mobile devices only implement a limited method for deciding when to offload traffic to WiFi (Figure 8). When a mobile terminal discovers and connects to a WiFi network, it steer all its traffic over the WiFi network. The main drawback of this solution is the lack of QoS considerations that can lead to situation where a mobile device will switch from a high data rate cellular connection to a low data rate WiFi connection.

In current networks, operators cannot influence the decision on mobile stations to offload traffic, but the idea is so appealing, that some activity by several standardization forums was taken. They propose operator-controlled WiFi, which are deployed and managed by an operator and/or its partner. In 3GPP Release 12, some WLAN/3GPP inter-working aspects were standardized. Their aim is to provide the network operator control mechanisms to steer traffic offloading in both the downlink and uplink. Using WiSHFUL UPIs, these solutions, can provide the mobile station with parameters such as receive power level threshold. When the received power is higher than this threshold mobile can offload its traffic via a network operated controlled WiFi AP.

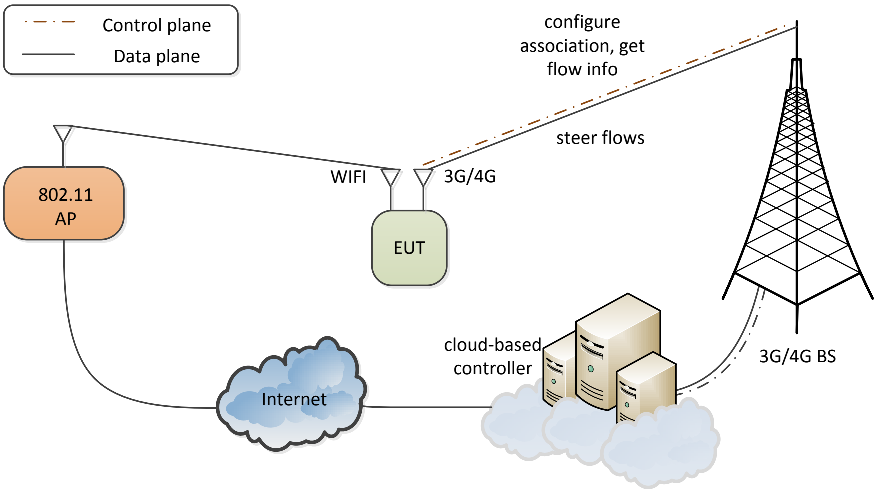

Figure 8. Controlling WIFI offloading.

Use of WiSHFUL Functionality

To support WiFi offloading, several WiSHFUL functions are needed. First, the network operator will use WiSHFUL UPIs to define parameters (e.g. receive power thresholds) that will be send to the mobile STAs. These parameters allow STAs to connect to the best network (WiFi or LTE). Second, the network provider needs to decide which flows should be offloaded from cellular to WiFi networks. For example, all VoIP traffic stays in the cellular network, because it provides a more robust and reliable connection, and all flexible flows using the TCP protocol (eg. file transferring, web browsing) are offloaded to WLAN. This decision can be enforced using the WiSHFUL UPIs.

5 Display of Showcases

Here we briefly list the venues where we the showcases can be displayed as well as appropriate discplay methods.

5.1 Venues for Display

As specified in the WiSHFUL proposal document, we aim at minimum 2 events per year (see KPI page 6 in the DoA). We will target venues best suited for the display of WiSHFUL showcases. The events, listed below, are considered to be good candidates for this purpose:

- ICT (ICT 2015 Innovate, Connect, Transform): the biggest ICT event in Europe
- EuCNC: European Conference on Networks and Communications
- Net Futures: yearly event organissed by teh Europen Commisiion to maximize competitiveness of the European technology industry.
- ETSI workshops such as the ones organized by teh Reconfigurable Radio Systems Committee
- WinnCom-Europe: Wireless Innovation Forum European Conference on Communications Technology and Software Defined Radio Wireless Innovation Forum ()
- IEEE DySPAN: Dynamic Spectrum Access Networks
- IETF (Internet Engineering Task Force) interoperability events
- EPRA: European Platform of Regulatory Authorities

5.2 Display method

As stated, the main idea behind the showcases is to illustrate the control capabilities of the WiSHFUL UPIs that allow creating more dynamic and flexible wireless solutions. With these showcases we want to attract other experimenters both in research and industry to participate in the open calls and to use and extend the WiSHFUL UPIs for their solutions. For this purpose, we will draw attention and awareness for the WiSHFUL project using fleyers, posters and talks in workshops organized in the context of the venues mentioned above. When the WiSHFUL functionality is implemented (end of year one) also demonstrators can be shown during the workshops. Where feasible, these demonstrators will make use of portable testbed, thereby also showcasing this WiSHFUL contribution.

Beside the participation on venues, the showcases will also be made available for external parties via the WiSHFUL portal. The portal allows executing the showcases on the testbed infrastructure hosted by the WiSHFUL consortium, thereby demonstrating its usage and capabilities. Via the WiSHFUL portal, specific challenges can be organized that allow external experimenters to upload a solution for a particular showcase and compete with the solution provided by the WiSHFUL partners. This will increase the engagement of external experimenters using a competitive and entertaining approach.

6 Conclusion

Within this document we have provided an initial glossary that enables to discuss the solutions, developed within the WiSHFUL project, in a consistent manner. Using this glossary, it is possible to describe the various aspects of the WiSHFUL platforms, functionalities and interactions with other projects and users. For this purpose, also a conceptual architecture is proposed, which can be applied on each platform and technology considered by the WiSHFUL projects. The consistency in terminology was verified with other, related EU projects (specifically Fed4FIRE).

Following the definition of the terms necessary to discuss WiSHFUL, we have herein refocused the driving scenarios provided in the prior deliverable 2.1 on *High level requirements for testbeds and software platforms* in order to drive the showcases. Based on this motivation, we have described the initial showcases for WiSHFUL that will be implemented and demonstrated after year 1. Each showcase provides a definition of the scenario to be addressed and a conceptual approach for applying WiSHFUL functionality within the given scenario. The technical details regarding the implementation of each showcase will be given in deliverables D3.1, D4.1 and D6.1. Additionally, two showcases were discussed that are left for future consideration and provide clarification of the WiSHFUL functionality.

Finally within this document, we briefly address means to maximize the impact of the showcases developed within the WiSHFUL project. In this regard, we examine the selection of appropriate venues for display of the showcases.

7 References

[1] Young, Alexander R., et al. "CSERE (Cognitive System Enabling Radio Evolution): A modular and user-friendly cognitive engine." Dynamic Spectrum Access Networks (DYSPAN), 2012 IEEE International Symposium on. IEEE, 2012.

[2] Fortuna, Carolina, and Mihael Mohorcic. "A Framework for Dynamic Composition of Communication Services." ACM Transactions on Sensor Networks (TOSN) 11.2 (2014): 32.