Cognitive Radio Experimentation World

CREW Methodology for performance evaluation

Authors

Stefan Bouckaert (IBBT), Michael Mehari (IBBT), Wei Liu (IBBT), Ingrid Moerman (IBBT), Peter Van Wesemael (IMEC), Danny Finn (TCD), Mikolaj Chwalisz (TUB), Jan Hauer, (TUB), Michael Doering (TUB), Nicola Michailow (TUD), David Depierre (TCS), Christoph Heller (EADS), Miha Smolnikar (JSI), Zoltan Padrah (JSI), Matevz Vucnik (JSI)

About this document

This document describes the CREW methodology for experimental performance evaluation. While the scope of the CREW methodology is the analysis of cognitive networking and cognitive radio solutions, the methodology is broader in a sense that it may be applied to a wider range of (wireless) networking experiments.

The content in this document is largely taken from CREW deliverable D4.2, which is yet to be approved by the European Commission. As soon as deliverable D4.2 is approved, the latter document will supersede the information contained in this document.

In addition to the methodology described in this document, (a growing list of) more concrete hints concerning the use of the different testbeds may be found on the CREW portal, more specifically in the methodology section www.crew-project.eu/portal/methodology

Table of contents

1 Best practices for experimental performance evaluation

Irrespective of a specific research field, any researcher or developer thinking of using experimental methods for designing and evaluating solutions is presented with a lot of questions related to the methodology. The first question one should ask themselves is a very fundamental one: is experimental validation/experimental design the optimal strategy to reach my goals?

Without denying the value of theoretical research or simulations, it is fair to say that experimentally-supported and experimentally-driven research has always been important in recent research history, across many research domains. The ultimate example of what experimental research can lead to in the field of ICT is probably the development of the Internet as we know it today. Even more recently, the FIRE (Future Internet Research and Experimentation [1]) initiative of the European Commission -where CREW is a part of- indicates the value of experimental research in today's ICT research ecosystem.

Especially in wireless networks, experimentally-driven research is often indicated to be the ideal solution to overcome the limitations of network simulators [2,3], which struggle to accurately model the complex behaviour of the wireless environment. While it is true that the outcome of simulations can be (easily) misinterpreted, this does not mean that using experimental validation methods "by default" leads to results that can 100% be trusted: if an experimenter does not carefully plan or execute an experiment, wrong conclusions may easily be drawn from an experiment.

For wireless networks in general and cognitive networks in particular, **the CREW project offers two important contributions that help to reduce the risk of drawing wrong conclusions from an experiment significantly**:

- Tools and testbeds supporting the experimenter. Instead of having to set up an ad-hoc test environment for each CR/CN experiment by themselves, experimenters can make use of the CREW federation. As such, researchers now have access to a large diversity and quantity of devices and tools. Furthermore, as the federation is accessible to a wide public, relevant comparison (benchmarking) of CR/CN solutions becomes possible, thus increasing the value of the experimental results.
- An experimentation methodology and good practices for experimenting on top of the CREW federation. In and outside the scope of CREW, the members of the CREW consortium have used their testbeds and tools themselves for evaluating CR solutions and as such they want to share their experience with the research community.

The practical experiences that were gathered during this process were compiled into the following subsections. First, Section 1.1 presents the general experimentation methodology that was followed in the past to come to, among other things, the results that were previously published in D6.1. This methodology is designed in such way that it is relevant for all CREW testbeds, and –with minor changes- also for a wider range of experimentation facilities. However, to help the experimenters with concrete hints for executing concrete experiments on top of the CREW facilities, readers are referred to the CREW portal:

- Detailed information on the different CREW facilities: www.crew-project.eu/portal/reference
- Good practices per testbed: http://www.crew-project.eu/portal/methodology

1.1 General experimentation methodology

There are four important steps in the experimentation process: defining an experiment, executing an experiment, retrieving and processing results, and sharing the results. The general experimentation methodology starts from these four steps, but puts them in a wider context by adding all additional steps that are important in the experimentation process.

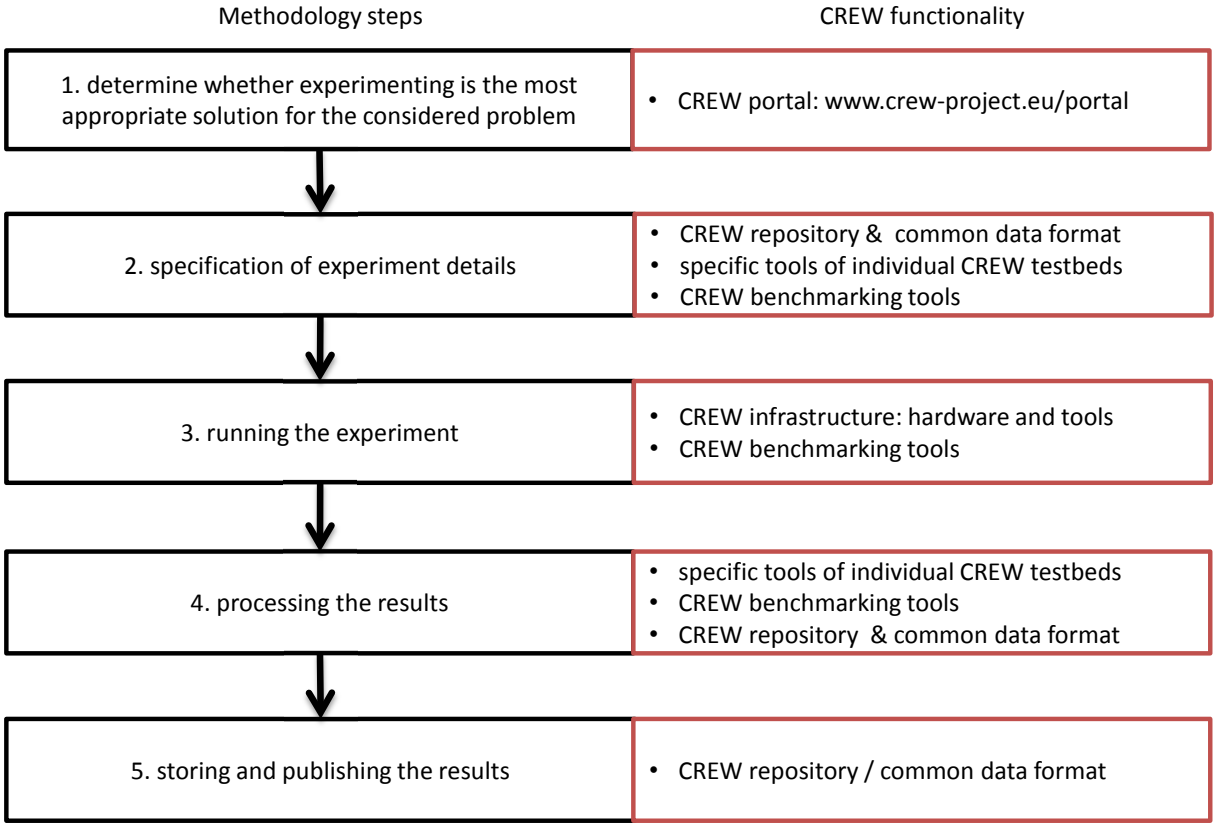

Figure 1 - Methodology overview and related CREW functionality supporting the experimenter

Figure 1 summarizes the different steps in the CREW experimentation methodology, and links the steps in the methodology to the functionality that is developed and provided as part of CREW. In what follows, methodology step is further detailed.

1.1.1 To experiment or not? Determine the appropriate solution for performance evaluation.

As stated in the introduction of this section, the most basic question for any person looking to characterize a solution is whether experimentation is the best possible action for the problem under consideration.

From the perspective of CREW, the ability to carry out real experiments with cognitive devices and cognitive network is obviously important; to find out whether promising theoretical concepts are also realizable in real life, to show decision makers what is really possible with CR today, and to discover potential practical issues that may arise when deploying real cognitive solutions.

Obviously, to be able to make decisions considering "experimenting or not", an experimenter needs to know what is available. The CREW portal (see D3.1 and other documents) was

realized to support experimenters in finding out what is possible and what is not, using CREW. Important hints during the discovery phase are the following:

- Use the filters of the CREW portal at http://www.crew-project.eu/portal/listoftestbeds to quickly narrow down the list of testbeds to a list of relevant testbeds.
- Go carefully through the available documentation to discover the possibilities but also the limitations of a testbed. While testbed environments can be very flexible, compromises will likely have to be made when experimenting, compared to taking a simulation approach. The size of an experiment cannot scale endlessly, all hardware has its limitations (varying parameters is more complex and usually more limited compared to simulation environments), experiments cannot happen faster than real-time. Implementing cognitive concepts may take a lot of time, so knowing any possible limits of the experimentation environment in advance is important in order to make sure that the implementation efforts will also result in an experiment outcome that matches the expectations.
- Just as with any experimentation facility, the CREW federation is in constant evolution. Furthermore, the consortium is open to suggestions that can improve the experimentation experience. Check the available documentation regularly and contact the CREW partners in case of any questions or ideas.

After making a well-informed decision to start experimenting, the concrete experiment(s) can be designed and specified.

1.1.2 Design and specify the experiment

Crucial in the design phase (and by extension throughout the entire experimentation process) is to be very precise in logging as much as possible information on the experiment: at any time during or after the experiment, it must be possible to go back to the exact configuration of the experiment.

In CREW D4.1, the experiments have been specified by describing information in two categories:

1. *a configuration scenario* containing the description of (1a) network *conditions* (technologies used in the experiment, topologies that are considered,…), (1b) *applications* (defined in the broad sense: any application at any OSI-stack layer that is part of the experiment, e.g. traffic generation, frequency optimisation, monitoring applications, etc.) together with the parameters that can be varied (and the specific values of these parameters that are considered), and (1c) *interference sources* (real or emulated primary user traffic, real or artificially generated interference);
2. *a description of the performance metrics* that will be recorded during the experiment.

While logging of the above information can be done manually, CREW also offers many tools to simplify and improve the way in which an experiment can be designed and stored:

- **The CREW repository** contains several types of information that are of use to experimenters designing experiments. First of all, to get an idea of how experiments can be described, full experiment descriptions can be found on the repository. Although these experiments may not be fully re-usable, some components of the experiments can be reused. More precisely, the *wireless background environments* that are used in some of the experiments can be downloaded as separate files, which

can be used to (as a base to) generate controlled interference. Also traces (see Section **Error! Reference source not found.**) may be downloaded and reused for the configuration of the experiment, and the same goes for metrics. When new reusable experiments or experiment components are generated, they can in their turn be added to the repository, thus (i) increasing the amount of useful information available on the repository, and (ii) making the information publicly accessible, thus helping to improve the reproducibility and repeatability of experiments.

- **The common data format** and corresponding tool to generate experiment descriptions, available from the portal at http://www.crew-project.eu/portal/CDF allow experimenters to store their experiment configuration in the common data format, again improving repeatability and repeatability.
- **Various tools specific to the individual CREW testbeds** such as the tools used to operate the IBBT testbed, force the user to fully describe the experiments in a clear and unambiguous way. Examples of such descriptions can be found on the CREW repository (see bullet above).
- **The CREW benchmarking tools** also can be used to enforce the full definition of an experiment and to produce a description of the experiment in the common data format. Note that while the benchmarking tools are currently only operational in the IBBT testbed, these tools have clean APIs that make it possible to port them to other testbeds.

When performing CR/CN experiments, it is good practice to separate the description of the solution under test from the description of the wireless background traffic, as it creates a clear separation between the experiment itself and between the external influences interfering with the solution under test, which stimulates reuse of background interference and in its turn enables benchmarking of different solutions against similar background conditions.

It is not always possible to define all aspects of an experiment before actually conducting it. For example, in case an experiment is performed inside an open environment where external interference cannot be controlled, the background interference cannot always be adequately recorded; even if the interference can be recorded, it is not at all straightforward to simply replay this interference. In these cases, it may be interesting to consider following strategy that can be used to emulate realistic wireless background traffic in a repeatable way:

1. Make a recording of spectrum or packet-level information using spectrum analysers or packet recording tools at a certain location of interest.
2. Analyze the recording to derive the relevant statistical information which characterizes the recording (e.g. number of transmitters, frequencies, duty cycles, …).
3. Set up a new wireless background scenario in one of the CREW testbeds, which is modelled according to the information derived from (2). In contrast with the uncontrollable interference at the location of interest, this scenario can be repeated as many times as needed.
4. Use the scenario created under (3) as background scenario, and add the system under test to be considered.

Example implementation of a background scenario created according the above method can be found on the CREW repository. For example, the *wilab-office-1* environment available on http://www.crew-project.eu/repository/background is based on a packet capture of the ISM Wi-Fi traffic as recorded at the IBBT office building in Ghent. The traffic was analyzed to extract those access points which caused most interference on the ISM band, and their

corresponding Wi-Fi clients and their typical use of traffic. Next, this information was used to generate a background interference scenario using multiple wireless access points and clients that can be deployed in the w-iLab.t Zwijnaarde testbed environment. While the resulting background scenario is obviously not identical to the recorded source trace, the spectrum/packet characteristics are very similar, which results in far more realistic background scenarios than if they would have been generated "at random".

A final set of generic hints during the experiment definition phase follows below:

- In testbed environments, the choice for using specific nodes can have a great impact on the outcome of the experiment. It is therefore advised to characterize a single solution in multiple topologies, using different nodes. The outcome of each of these different experiments should then be compared (see phase 4) and checked for consistency.
- Try to define the expected output (metrics) as soon as possible in the experimentation cycle, and think of how they will eventually be processed. There is nothing more frustrating than to realize another easy-to-record parameter should have been logged to generate a certain result or output graph. Therefore, if not enforced by the testbed, take note of all settings (including version settings of applications, operating systems…) and configure the experiment in such way that all potentially interesting parameters will be recorded, with a sufficient level of detail. From experience, it is better to record a couple of extra parameters that may initially look irrelevant, than miss a single parameter while processing the results.
- Once again, log as much information as possible: although some settings may seem very obvious at the time of the experiment, some weeks (or even years) later it may be less evident –yet crucial- to remember these. Worst case, experiments will have to be repeated, which is only possible in case the experiment configuration contains all details.

1.1.3 Running the experiment(s)

Once the experiment or experiments are defined, they can be executed inside the testbed under consideration. General best practices are presented below:

- In addition to the variations that are needed during the experiment definition, a key factor to achieving reliable results while running the experiments is repeating the experiments multiple times. How many times an experiment should be run depends on the complexity and duration of the experiment, but especially on the variations that are recorded at the results side. While these variations can be monitored manually, the CREW benchmarking framework makes it possible to monitor the variations of the output variables automatically.
- The CREW benchmarking framework can also be used to automatically vary input parameters in between different experiments, thus again saving time for the experimenter.
- When running the initial experiments, it is good practice to move iteratively between phases 3 (running the experiment) and 4 (processing the results), in order to reduce the risk of running a large set of experiments which afterwards prove to be invalid (e.g. because of a bug in the solution under test). By using the CREW testbeds and tools in a proper way (as described on the CREW portal), errors during the experimentation phase will be significantly reduced. Furthermore, some of the CREW testbeds offer

the possibility to get a real-time view on the status and results of the experiments; this is very useful to detect any possible errors in the earliest possible state.

- In some cases it might be useful to repeat experiments at different times of the day. As most testbeds are deployed in environments where people are working (thus passing by the set-ups and/or using the wireless spectrum), results might be affected by factors external to the experiment. Try to get and store a view on the relevant portion of the spectrum before, during and after an experiment as this might help to identify issues when processing the results. Where supported, such assessment of the spectrum can happen automatically by using the CREW benchmarking tools.

1.1.4 Processing experimental results

- Although processing results can be done manually, CREW provides several tools to help the experimenter with this. Several of the CREW testbeds provide support for real-time and post experiment processing of the results, as can be found on the CREW portal. Regardless of whether the results are processed manually or through tools or by using (an adaption of) one of the processing scripts that are found on the CREW repository, it is good practice not to remove any of the source data (i.e. raw metrics, spectrum info) even after it has been processed, as the source data might be required at a later time to calculate additional metrics or to discover the source of anomalies.
- At several occasions during internal CREW experiments, use of the CREW common data format has proved its use when processing data from a set of different cognitive devices. The scripts that are found on the CREW repository typically require data formatted in the common data format as input, which significantly simplifies the comparison of data at a later stage.
- If new scripts are generated to process results, it is good practice to store them again on the CREW repository, as this increases the transparency of the experimentation process and again leads to results that are more easily compared.

1.1.5 Storing and publishing experimental results

- While not publicly accessible at the moment of writing (but accessible on-demand by people using CREW), there is a CREW data server available where results (even large data sets) can be stored for a longer time. The public extension of this data server is found in the CREW repository, where results can be made publicly available, preferably in the common data format. In a later phase, it will either be possible to directly contribute to the CREW repository, or, CREW may decide to merge its data with existing open repositories such as CRAWDAD [4].
- To make it possible for the target audience to understand the background of the experiments that led to the results, it is good practice to share as much data as possible with the target audience. As among other things in scientific papers, it is impossible to share all data in an “offline” way, the CREW repository and common data format can be used to make results available online. A link to the repository can then be added to the scientific (or other) publication. In the long run, this approach could lead to a valuable database of experiments and corresponding results.
- If the steps above were followed carefully, sharing the exact configuration and circumstances of the experiment should not be too complex, as all information is normally already available in a clear and understandable way.

2 References

[1] European Comission, Future Internet Research and Experimentation initative. Homepage: cordis.europa.eu/fp7/ict/fire/

[2] David Cavin, Yoav Sasson, and Andre Schiper. “On the accuracy of MANET simulators”, POMC '02: Proceedings of the second ACM international workshop on Principles of mobile computing, pages 38--43, New York, NY, USA, 2002. ACM Press.

[3] Todd R. Andel and Alec Yasinsac. On the credibility of manet simulations. Computer, 39(7):48--54, 2006.

[4] CRAWDAD Community Resource for Archiving Wireless Data At Dartmouth. Homepage. http://crawdad.cs.dartmouth.edu/