



A Roadmap for Benchmarking in Wireless Networks

Shafqat Rehman, Thierry Turletti, Walid Dabbous

► To cite this version:

Shafqat Rehman, Thierry Turletti, Walid Dabbous. A Roadmap for Benchmarking in Wireless Networks. [Research Report] RR-7707, INRIA. 2011, pp.34. inria-00614167

HAL Id: inria-00614167

<https://inria.hal.science/inria-00614167>

Submitted on 9 Aug 2011

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

A Roadmap for Benchmarking in Wireless Networks

Shafqat Ur Rehman — Thierry Turletti — Walid Dabbous

N° 7707

August 2011

A large, light gray stylized 'R' logo that serves as a background for the text 'Rapport de recherche'.

*Rapport
de recherche*

A Roadmap for Benchmarking in Wireless Networks

Shafqat Ur Rehman *, Thierry Turletti *, Walid Dabbous *

Theme :
Équipes-Projets OneLab2

Rapport de recherche n° 7707 — August 2011 — 34 pages

Abstract: Experimentation is evolving as a viable and realistic performance analysis approach in wireless networking research. Realism is provisioned by deploying real software (network stack, drivers, OS), and hardware (wireless cards, network equipment, etc.) in the actual physical environment. However, the experimenter is more likely to be dogged by tricky issues because of calibration problems and bugs in the software/hardware tools. This, coupled with difficulty of dealing with multitude of hardware/software parameters and unpredictable characteristics of the wireless channel in the wild, poses significant challenges in the way of experiment repeatability and reproducibility. Furthermore, experimentation has been impeded by the lack of standard definitions, measurement methodologies and full disclosure reports that are particularly important to understand the suitability of protocols and services to emerging wireless application scenarios. Lack of tools to manage large number experiment runs, deal with huge amount of measurement data and facilitate peer-verifiable analysis further complicates the process. In this paper, we present a holistic view of benchmarking in wireless networks and formulate a procedure complemented by step-by-step case study to help drive future efforts on benchmarking in wireless network applications and protocols.

Key-words: wireless networks; wireless experiments; benchmarking methodology; wireless tools; repeatability; IEEE 802.11

* Planete Project Team, INRIA Sophia Antipolis, France

A Roadmap for Benchmarking in Wireless Networks

Résumé : Experimentation is evolving as a viable and realistic performance analysis approach in wireless networking research. Realism is provisioned by deploying real software (network stack, drivers, OS), and hardware (wireless cards, network equipment, etc.) in the actual physical environment. However, the experimenter is more likely to be dogged by tricky issues because of calibration problems and bugs in the software/hardware tools. This, coupled with difficulty of dealing with multitude of hardware/software parameters and unpredictable characteristics of the wireless channel in the wild, poses significant challenges in the way of experiment repeatability and reproducibility. Furthermore, experimentation has been impeded by the lack of standard definitions, measurement methodologies and full disclosure reports that are particularly important to understand the suitability of protocols and services to emerging wireless application scenarios. Lack of tools to manage large number experiment runs, deal with huge amount of measurement data and facilitate peer-verifiable analysis further complicates the process. In this paper, we present a holistic view of benchmarking in wireless networks and formulate a procedure complemented by step-by-step case study to help drive future efforts on benchmarking in wireless network applications and protocols.

Mots-clés : wireless networks; wireless experiments; benchmarking methodology; wireless tools; repeatability; IEEE 802.11

1 Introduction

Performance evaluation of wireless networking systems and protocols has recently witnessed tremendous research activity. Evaluation techniques employed range from mathematical modeling to in-field experimental evaluation each with its own pros and cons. A survey of the wireless networking literature reveals that majority of articles embrace simulation as a convenient approach for the performance evaluation of such systems [49]. However, lack of realism because of simplistic radio models in stochastic simulations can lead to misleading analysis [49] [56].

The best way to achieve realism is to perform in-field experiments using 'real' hardware and software. Unfortunately, wireless experimentation is not a smooth process and configurability and management of even a small number of nodes is cumbersome. Moreover, behavior of the network is tightly coupled with the conditions on the physical layer and lack of control over these conditions poses a big challenge to comparative performance analysis. This is because wireless channels are unpredictable (random), error-prone and could vary over very short time scale (order of microseconds). It is also difficult to avoid colocated wireless networks. However, experimenter does have control over some network parameters which can be configured to fixed values. Henceforth, all the configurable network parameters are considered to be *controllable* and the rest *uncontrollable*. Uncontrollable parameters include station workload (memory/cpu usage, etc.), network traffic load, multipath fading, path loss (attenuation), channel interference, etc. Controllable parameters entail scenario configurations (topology, traffic, wireless card configurations). Controllable parameters can also include meta-data such as system hardware/software specifications.

It may seem trivial to keep track of all these parameters, but ensuring correctness and soundness of the measurement data can prove to be tricky. Some parameters and metrics concerning channel characteristics are influenced by, among other things, calibration settings. For example, if the impact of power/rate adaptation, noise floor calibrations and interference is ignored, it can lead to misleading results [69] and hence compromise the trustability of the results. Experimental data sets, code, and software are crucial elements in scientific research; yet, these elements are noticeably absent when the research is recorded and preserved in the form of a scholarly article. Furthermore, most researchers do not deposit data related to their research article [67]. In [68], authors conducted an informal study of 33 of last year's accepted articles from the prestigious ACM SIGCOMM 2010 conference. The study indicated problems with 70 % of the accepted papers related to proper description of methodology, experiments, and analysis. This makes it difficult for peers and reviewers to confirm the results and hence reduces trust in the findings.

This boils down to reproducibility of experiments. *Reproducibility* is the ability of an experiment to be replicated by independent reviewers on their own. It is not necessary for others to get exactly the same results. Usually, variation in the measurements is unavoidable. If the this variation is smaller than some agreed limit, the experiment would be deemed reproducible. Reproducibility is often confused with *repeatability* which is being able to do the same experiment over and over at the same site by same experimenters and get the same results (i.e., with variation smaller than the agreed limit) each time. When it comes to simulation, repeatability is easily achievable. Reproducibility, however, may

still require more care and effort. In any case, it may be sufficient to maintain provenance (i.e., chronological record of measurements and analysis steps) of results (figures, tables, graphs, etc.) with all the parameters, data, source code and scripts. However, in real-world wireless experiments, both repeatability and reproducibility are non-trivial and elusive.

Reproducibility has been at the core of research in most fields of science. An experimental result would be worthwhile only if it can be reproduced by peers. In natural sciences, an experiment is considered reproducible if the protocol/procedure used is described in sufficient detail along with reagents, specifications of the equipment, times, temperatures, etc. [13]. However, networking experiments can't be reproduced by such measures because the distributed software is much more complex. Indeed, wireless experiments include additional complexities such as volatile radio spectrum, software/hardware imperfections/calibrations, configurability, management of resources and data, etc. [30]. It is impossible to ensure same channel conditions until and unless the experiment setup is insulated from outside interference using shielded enclosure. That may be the reason why rigorous *peer verification* of experimental results is not becoming a culture in networking field as yet [23]. In this paper, we restrict ourselves to the repeatability of wireless experiments. However, the roadmap presented is valid for reproducibility as well.

Because of uncontrollable experiment conditions, it would be impractical to repeat a *real-world* (non-shielded) wireless experiment the way experiments are repeatable in other fields of science. We, therefore, focus on getting around this obstacle via conducting large number of runs, and clustering them according to the similarity of experiment conditions. Essentially, the entire procedure entails following steps: define the scenario precisely, conduct large number of runs with fixed and recorded controllable parameters, measure all the uncontrollable conditions (i.e., parameters/metrics), cluster the runs as per the conditions, and perform an *apples to apples* comparison. This would provide a level playing ground for the *fair comparison* of networking protocols. This will also lead researchers to archive and share data and code and hence enable future researchers to compare their experimental results with the previous ones [13].

The above procedure requires a highly systematic and scientifically rigorous experimentation approach which is often challenging due to the cost, complexity, and scale of the involved experimental resources, and some potential limitations in the training of the research investigators [68]. Generally, an experimenter is required to deal with a host of issues such as testbed setup, installation of hardware/software tools, calibration and instrumentation of tools, sanity checks, imperfections and limitations of tools, scenario description, scheduling and management of experiment runs, meta-data collection, data cleaning, synchronization and merging of traces, data normalizations/transformations, analysis, reports, data management (measurement data, meta-data, code, assumptions, archiving, sharing, etc.).

In this paper, we provide a roadmap for scientifically rigorous wireless experimentation *in the wild*. We promote the notion of wireless benchmarking which encompasses the aforementioned requirements and can provision a level playing ground for the realistic comparative evaluation of networking protocols. More precisely, benchmarking signifies *fair apples to apples comparison* of wireless systems relative to a *reference evaluation*. The seemingly simple task of benchmarking is surprisingly complex in wireless networks because it requires

an *in-field* evaluation of the system to ensure real world operational conditions. The complexity of such an evaluation is compounded by the lack of control on experimental conditions and lack of tools to deal with the issues listed above.

The remainder of this paper is organized as follows. In section 2, we elaborate state of the art and provide critical analysis of the existing work. In section 3, we provide a brief overview of benchmarking and list benefits and challenges for wireless networks. Section 4 provides detailed step-wise account of your proposed methodology for wireless experimentation and benchmarking. In section 5, we provide an experimental case study which implements all aspects of the benchmarking methodology. Section 6 concludes the paper.

2 Related Work

In this section, we focus primarily on experimentation methodologies that have been designed to make experimental results trustable by facilitating repeatability and peer-verification. Also, we look into other contributions such as guidelines, tools and data repositories aimed at benchmarking in wireless networks.

In [65], after an extensive analysis of published work, authors conclude that lack of information about scenario, methodologies, and meta-data hampers reproducibility and peer verification. They have proposed a web portal called LabWiki where experimenters can describe the experiments and store all the information about an experiment. Each of the artifacts of an experiment and the experiments themselves are identified by public URLs which can be linked easily from any LabWiki pages. Authors propose to use the R language to analyze measurements collected on the portal. After the experiment development process is finished, a Portal user may choose to open up his or her LabWiki permissions to specific reviewers or the general public. However, the data repository is not implemented as yet and the portal is still in development process [66].

MyEmulab [55] is a web-portal to the Emulab network emulator testbed. It provides services to build experimental network topologies, upload an experiment description, automatically configure and execute the experiment. Furthermore, it provides Wiki and versioning tools to allow collaboration between members of a given project. However, MyEmulab does not provide services to archive, access, analyze the measurements. Also it does not offer any services to report and share the results.

NEPI (Network experiment programming interface) [70] proposes a framework based on a unified object model to describe a networking experiment which could subsequently be executed on different environments (e.g. simulations, emulations, testbeds). However, it is an ongoing work and differs in terms of focus. Currently, it does not handle real-world wireless experiments and lacks the support of multiple runs, collection of meta-data, data management, analysis, etc.

In [4], a comprehensive practice has been recommended for 802.11 wireless experiments. It contains essential information for setting up test scenarios in different wireless environments and metrics that should be considered in each scenario. In [12] [11], authors provide an insight into the selection of performance metrics for benchmarking. Selection of good metrics is necessary for producing useful benchmarks. However, these contributions are not complemented by a methodology.

In order to extend the value of measurements beyond a certain case study and make them accessible to broader research community, online network measurement data repositories are being deployed [50] [52]. The purpose of these repositories is to facilitate archiving, publishing and sharing of network measurement data. Data repositories have tremendous benefit for benchmarking. They foster collaborative research. They make it easier for peers to revisit the analysis or calculate additional metrics without repeating the entire experimentation effort when existing traces can serve the purpose. Effort is also being made to develop supporting analysis tools [54] [52] [53]. However, these efforts are still in their infancy and are disconnected from experimentation testbeds and methodologies. The only exception is the expected data repository proposed in [65] which is yet to materialize.

During the last decade, wireless experimentation has received significant attention. Various shared as well non-shared testbeds have been setup, toolkits/frameworks have been developed and data repositories have been deployed to efficiently deal with the complexities of wireless experimentation and facilitate analysis. Among notable testbeds and toolkits are Emulab [55], ORBIT (Open-Access Research Testbed for Next-Generation Wireless Networks) [23], MiNT (a *miniaturized mobile multi-hop wireless network testbed*) [62], Ad-hoc Protocol Evaluation (APE) testbed [59], EXC toolkit [30], etc. Each of these platforms is tailored to meet the requirements of a specific area of focus. However, a survey of the contemporary experimentation testbeds and toolkits is beyond the scope of this paper. We, rather, focus more on enabling benchmarking in wireless networks by designing a methodology which could be carried out using existing testbed and tools.

We have formulated a rigorous benchmarking methodology [Section 4] for protocol evaluation in wireless networks and developed a supporting experimentation toolbox referred to as WEX (Wireless Experimentation) toolbox [32] [Section 5]. Our methodology and the corresponding architecture facilitate workflow management of large experimentation campaigns spanning over weeks or even months. An indexing mechanism makes easier parallel processing of all the traces and it is possible to replay an experiment. In addition, we provide data repository and a web portal aimed at making experimentation results shown in networking papers verifiable by the reviewers. In the following sections, we provide an insight into wireless network benchmarking and details about the methodology.

3 Benchmarking Analysis

Benchmarking is a well known concept in many domains of computing and natural sciences. It is often applied to measure quantitative advantage of one system over another similar system. More precisely, benchmarking is evaluation of the performance of a system relative to a reference performance measure. It can be applied to virtually any system (business processes, progress reviews, software/hardware tools, protocols, etc.) which exhibits quantifiable performance indicators. It is proven means of improving performance, efficiency, cost-savings and competitiveness of a system [1]. It facilitates "learning from the experiences of others" and thus enables the identification, adaptation and deployment of protocols that produce the best results. However, the potential of benchmarking hasn't yet been utilized in networking and particularly in wireless networks.

The following subsections shed light on potential benefits and challenges in the way of benchmarking in wireless experiments.

3.1 Importance

Identifying, improving and deploying superior communication standards and protocols adds to the business value of a network and benchmarking provides the foundations. Some of the reasons that benchmarking is becoming more and more promising are outlined as follows:

Benchmarks can be published and used by interested research groups who then can contribute with relevant metrics, models and test scenarios. They enable increased reproducibility of results and provide a common base for fair and consistent comparisons. Standardized workloads, run rules and benchmark tools can speed up the performance estimation and hence organization's ability to make improvements in a more efficient way. The use of different metrics, test scenarios and measurement methodologies complicates comparison. Benchmarks help overcome these problems and promote healthy competition.

Benchmarking helps identify areas of cost reduction, enables a more detailed examination of efficiency and facilitates value-add. Benchmarks are also used to prepare proposals for product selection and system development. They can be employed to investigate how well an initial installation is performing. It is helpful in debugging a given configuration, determining where additional equipment needs to be installed and it can go a long way in providing most cost effective and functional installation in a given environment.

Basically, benchmarking is greatly useful in planning, testing and evaluating network performance. It is of great interest to engineering, marketing and executive level personnel. Staff can better prioritize which network problem needs to be addressed and "how good is good enough?"

3.2 Wireless Benchmarking Challenges

In wireless networks, radio propagation is highly dependent upon the physical environment including geometric relationship between the communicating nodes, movement of nodes, movement of objects and the type and orientation of the antennas used. Unlicensed ISM (Industrial, Scientific and Medical) bands of the spectrum available are being shared by an increasing number of different

devices making wireless medium more interference prone. Also, wireless networks are becoming more and more complicated. Modern APs can dynamically alter power levels and channel assignments in response to changing conditions. The rapid evolution in wireless technologies (e.g., introduction of smart antennas, directional antennas, reconfigurable radios, frequency agile radios, MIMO systems, multi-radio/multi-channel systems) makes benchmarking more complicated. Reproducibility is at the core of benchmarking but factors mentioned above coupled with volatile weather conditions, ever-shifting landscape of obstructions, network equipment aging, software/firmware bugs, etc. make network retesting, reconfiguration and hence reproducibility a big challenge [29].

Vagaries of the wireless medium have a sound impact on the performance of upper layers. Deciding what metrics to calculate and what parameters have direct or indirect impact on the low-level or high-level metrics is challenging. There is a need for tools that collect information at different layers and combine this information to allow a detailed and comprehensive analysis [30]. Data collected can be fairly large. Depending on the number of flows, data rates, duration of the experiment, and number of probes; collected measurement data can run into hundreds of Giga bytes. Synchronizing, merging and managing wireless traces is time consuming. In order to do the analysis, one needs to combine them into one coherent time-ordered sequence. It is costly in terms of time, computational and storage resources.

There are up to fourteen channels on 802.11b/g worldwide out of which only 3 channels are non-overlapping. In most cases, density of wireless nodes and a small number of non-overlapping channels make it impossible to ensure innocuous co-existence of different WLANs. Increased channel interference leads to degradation in network performance. In order to investigate channel interference on network performance, spectrum analysis is indispensable. During the course of our experimentation, we employed Wi-Spy [35] in conjunction with kismet spectools [36] for spectrum analysis using standard laptop computers. Wi-Spy enables to capture the entire 2.4 GHz band but, with a sweep time of around 600 ms, it is significantly slow.

The challenges identified herein are in no way exhaustive. Each wireless technology has its own specific open issues that need to be investigated. No panacea exists that can solve all of these problems. However, it is desirable to identify the strengths and weaknesses of existing solutions through well-formulated benchmarks that enable "apples to apples" comparison and enact a firm basis for future advancements. This will go a long way in alleviating the critical issues such as data rate enhancements, cost minimization, and user security in future wireless networks.

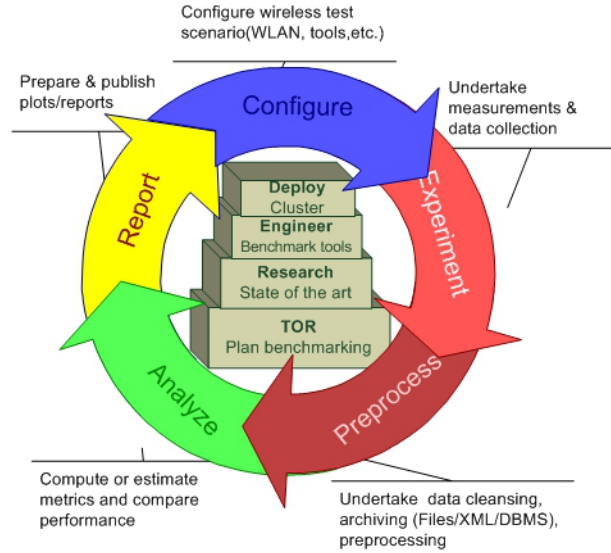


Figure 1: Wireless Network Benchmarking

4 BENCHMARKING METHODOLOGY FOR WIRELESS NETWORKS

In the field of network computing, benchmarking is common for network interconnection devices [3]. Recently, IEEE LAN/MAN Standards Committee prepared a recommended practice for the evaluation of 802.11 wireless networks [4]. Recommendations in [4] are valuable for the benchmarking methodology presented herein.

Figure 1 outlines the set of activities envisioned for benchmarking in wireless networks. The first step is to prepare **Terms of Reference (TOR)** or **Plan**. Other steps involve research on state of the art, design and development of benchmarking tools, setting up network management and experiment control tools. Then there is a cycle of activities as shown by circulating arrows in Figure 1. The cycle should be repeated for initial smoke runs in order to establish the precision and logic of results prior to the running of actual benchmarks. The cycle may also be repeated in order to achieve a certain level of confidence in the results. The activities include configuration of target test environment, performing the experiment, and undertaking measurements and data collection, analyzing and producing results, managing data-centric activities (such as storage, security, sharing, etc.) and preparing and publishing benchmark reports.

4.1 Plan:Terms of Reference

This step forms the basis of benchmarking. It sets goals and provides motivation for the undertaking. Type of the network (e.g., WiFi, WiMAX, Bluetooth, GPRS, IrDA etc), area of focus (e.g., wireless application such as peer-to-peer video streaming, content sharing), scope of measurements (i.e., set of metrics), and target deployment environment (indoor, outdoor, etc.) are key considera-

tions. Priority should be given to the area that has greater value to the users or area that consumes larger resources. It has to be decided what should be the key indicators or metrics based on their importance and major aspects of cost. Terminology in the context of area of focus has to be defined so as to avoid confusing connotations. Planning for key benchmarking tasks such as network setup, cluster setup, benchmarking tools (e.g., traffic generators, sniffers, etc.), trace and meta-data collection, preprocessing, statistical analysis, reporting etc., has to be done.

A set of deliverables which conform to the requirements, scope and constraints set out in planning has to be listed and elaborated. A documentation or data management system has to be developed. Terms of reference are subject to change during the process of benchmarking as a consequence of change in high level requirements or system artifacts that may become clear later on.

4.2 Investigate: Existing best practices

Research on the current state of benchmarking and evaluation paradigms across domains relevant to benchmarkee [2] is constructive so that benchmarkers can bring them up to speed with the advances, avoid re-inventing the wheel, be able to make use of the existing knowledge-base of best practices and software tools, and start off from where peer benchmarkers have left. One needs to develop a comfortable understanding of the underlying wireless standards. It is imperative to investigate selection of metrics, run rules, baseline and peak configurations (if any), limitations, risks etc.

For instance if one were interested in improving hand offs in wireless networks, she would try to identify other domains that also have hand off challenges. These could include air traffic control, cell phone switching between towers etc. Typical aspects/metrics to consider would include (but not limited to) handoff costs, efficiency, delay, errors, and QoS. Benchmarking of handoffs is critical because depending on the application scenario, failed or wrong handoffs can result in enormous cost or have disastrous consequences.

4.3 Engineer: Benchmark tools

An ensemble of software tools (benchmarking toolbox) is at the core of any benchmarking endeavor. Before delving into the development of such tools, it is very useful to explore existing tools that serve a similar purpose. The golden rule here is to avoid re-inventing the wheel and re-use the existing code where possible in order to cut down the development cost. Benchmarking tools are desired to evolve as a result of bug-fixes, functional enhancements, re-factoring, re-engineering, etc. An agile development approach would be suitable wherein some of the benchmark tools would be implemented or adopted based on their priority. Sometimes adjustments are required to account for new understanding gained through mistakes during the course of benchmarking.

For example, consider wireless mesh networks. Wireless meshes normally facilitate broadband applications with various QoS requirements. Suppose in order to test the mesh network's ability to carry large amounts of traffic, say in video surveillance of a metropolis, capacity planning might take precedence over security planning. In this case, we can put benchmarking of security or

other qualitative aspects on hold and concentrate on what is more important: throughput.

It is more productive to embed the functional testing or unit testing (in vitro in most cases) within the development process. Indeed, this allows rapid enhancements and re-factoring while ensuring that the core functionality remains intact. A system documentation is necessary to describe the functionality, limitations, direction for future enhancements, dependencies and installation guidelines for the deliverable tools.

4.4 Deploy: Resource and Experiment control

The pre-requisite for benchmark tool suite deployment is setting up a computer network in the target test environment such that it meets all the mandatory software and hardware requirements laid out in the test specification. Typical test environments include calibrated over the air test (COAT) environment, conducted test environment, over the air (OTA) outdoor Line of sight (LOS) environment, OTA indoor LOS environment, OTA indoor non-line of sight (NLOS) environment and OTA shielded enclosure environment [3].

Deployment involves setting in place the network equipment and installing the required software. Setting up of a computing cluster is also desirable in order to manage the execution of experimental tasks on the set of nodes participating in the experiment. It also empowers the benchmarker to perform multiple runs faster and efficiently. It is very imperative to have the network equipment calibrated and all the benchmark software tested. It is good practice to use latest versions of firmware and drivers for all the wireless products. Products are normally shipped with default optimal settings. The decision whether to use baseline configurations or peak configurations or any other custom settings must be carefully considered but security settings might have to be adjusted anyway. Whatever settings are used must be carefully documented along with hardware models.

All of the required protocols will be configured and enabled during the setup. Parameters and settings associated with the devices and applications running thereon that affect the performance will have to be listed. Then, within this list, those parameters and settings that will vary during the experimentation have to be identified so that they can be included in the sampling process of network measurement. For example CPU usage, memory usage, swap usage, interference, etc.

We need to document all relevant configurations regarding devices (OS kernel, CPU, memory, etc.), tools (sniffers, spectrum analyzers, etc.), network (security usage, (TX, RX) signal levels, RTSCTS usage, etc.), number of senders/recievers, etc. Key to successful benchmarking is holding as many parameters as possible constant in order to isolate the contribution of specific elements being compared.

4.5 Configure: Wireless experimentation scenario

Configurations elaborated in section 4.4 are general and are concerned with the network resources and experiment cluster setup. All of the benchmark tests would have to be run without changing the general configuration/setup of the devices in anyway other than that required for the specific test scenario. In this

step, all the tools necessary to carry out the tasks specified in the experimentation scenario have to be calibrated. Nodes participating in the experiment such as source(s), receiver(s), probes and spectrum analyzers have to be configured. This is usually repeated for each run of the experiment to ensure a clean start.

4.6 Experiment: Undertake experiment execution and data collection

Multiple independent applications, such as data and streaming media, should be run and behavior of the wireless network should be measured according to the test specifications using a suitable sampling rate [11]. Applications should be representative of the real world situation and capable of associating with the wired/wireless interfaces of the devices and generating traffic at the desired rate. Benchmarkee should be tested under different frame sizes especially max and min legitimate frame sizes and enough sizes in between to get a full characterization of its performance [3].

Workload tools of the benchmark toolbox are expected to produce normal workload as well as fault-load. Fault-load represents stressful conditions to emulate real faults that are experienced in the real systems. Appropriate level of detail about the workload is important in order to make meaningful analysis. Network load characteristics along with extraneous and internal RF interference should be measured. Network variations such as link failures and congestions have to be reported. Meta data about the *result elements* (such as traffic samples, RF interference samples) and *configuration elements* (such as network settings and parameters) would aid in keeping track of the context in which experiment was performed. It is also important to structure the chain of steps between launch and termination of the experiment and maintain version control of the participating scripts. Employing visual tools to explore in-consistencies and changes in scenario definitions of the subsequent runs can result in big payoffs.

Performing network measurements is a complex process. Precision and accuracy of measurement devices and tools has to be documented. It must be clear to the benchmarkers as to whether they are measuring what they actually wish to measure. A general strategy would be to gather more than one type of data set - either from a different location in the network or from a different time [14]. Measurement data needs to be collected using open industry-standard formats. Collection of meta-data, even if its immediate benefit is not obvious, may become useful in future benchmarking practice. It can be extremely helpful to seek out early peer review of proposed measurement effort.

4.7 Preprocess: Data cleansing, archiving and transformation

The data collected in the experiment execution stage needs to be cleansed. This could be achieved by employing self-consistency checks and investigating outliers and spikes. The first question to ask would be if the experiment was performed all right. Validity and integrity of measured data has to be assessed. Traces collected using a sniffer may contain significant amount of exogenous traffic. In order to reduce transformation and processing time, it may be desirable to filter out irrelevant data before transformations and analysis. We need tools to

verify that measurements indeed provide a true picture of wireless network. One approach would be to create 802.11 finite state machines, look for inconsistencies and come up with an executive summary on the quality of measurements. If the measurements lack the desired level of integrity and validity, it would be required to repeat the experiment with better experience and improvements in the tools gained in previous measurement cycles.

Conducting benchmarking is costly in terms of time and resources. This, therefore, necessitates persistent structured storage of measured data using standard data formats such as XML [51], XBRL [19] and database management systems (DBMS). One such example is CRAWDAD [21]. Meta data (interference, variable resources, etc.) should also be associated with the traces.

4.8 Analyze: System performance

Finally, data would be processed to produce the results which represent the values of metrics as specified in the test specifications. For some metrics, data has to be transformed (normalized or converted to different scale) to fit the analysis needs. Effort should be made to minimize the generation of intermediate data between raw measurements and final results. Instead caches can be used for transient intermediate results. This would aid in reproducing the same analysis [14]. Calculation of mean (arithmetic or geometric) behavior over same measurements performed in different ways can provide a good insight of the network performance. Confidence intervals and distributions can also be used to depict the network behavior.

The whole chain of analysis must be documented. Versioning and storage of analysis scripts along with the measured data that underpins the results should be stored. We need to archive both measurement traces and benchmark results. Benchmark results are either obtained through internal benchmarking effort or from partner research groups or organizations. Versioning mechanism has to be employed to facilitate reproducible analysis.

4.9 Report: Benchmarking score

Reports are the windows through which benchmarkers [12] can gain a visual access to the results. They provide detailed insight into the strengths and weaknesses of benchmarkee. All the benchmark-related information which is complimentary to the results must be made available. Meta-data (e.g., precision of tools, accuracy of measurements, etc.) which could be useful for trouble-shooting, decision-making, and management action, should also be reported. Reports should include an executive summary consisting of comprehensive graphs, configured parameters and drill-down details (if any). In fact, reports have to be designed and presented in accordance with the *full disclosure report* (FDR). Full disclosure report, for each benchmark, is prepared in accordance with reporting rules. Producing and interpreting benchmark results crosses into the realm between art and science. Web services are a great way to provide access to the database of benchmark results. Web services, then, can be used by interested organizations and groups to gain access to the results. Web services will also enable distributed access and sharing in the form of web reports.

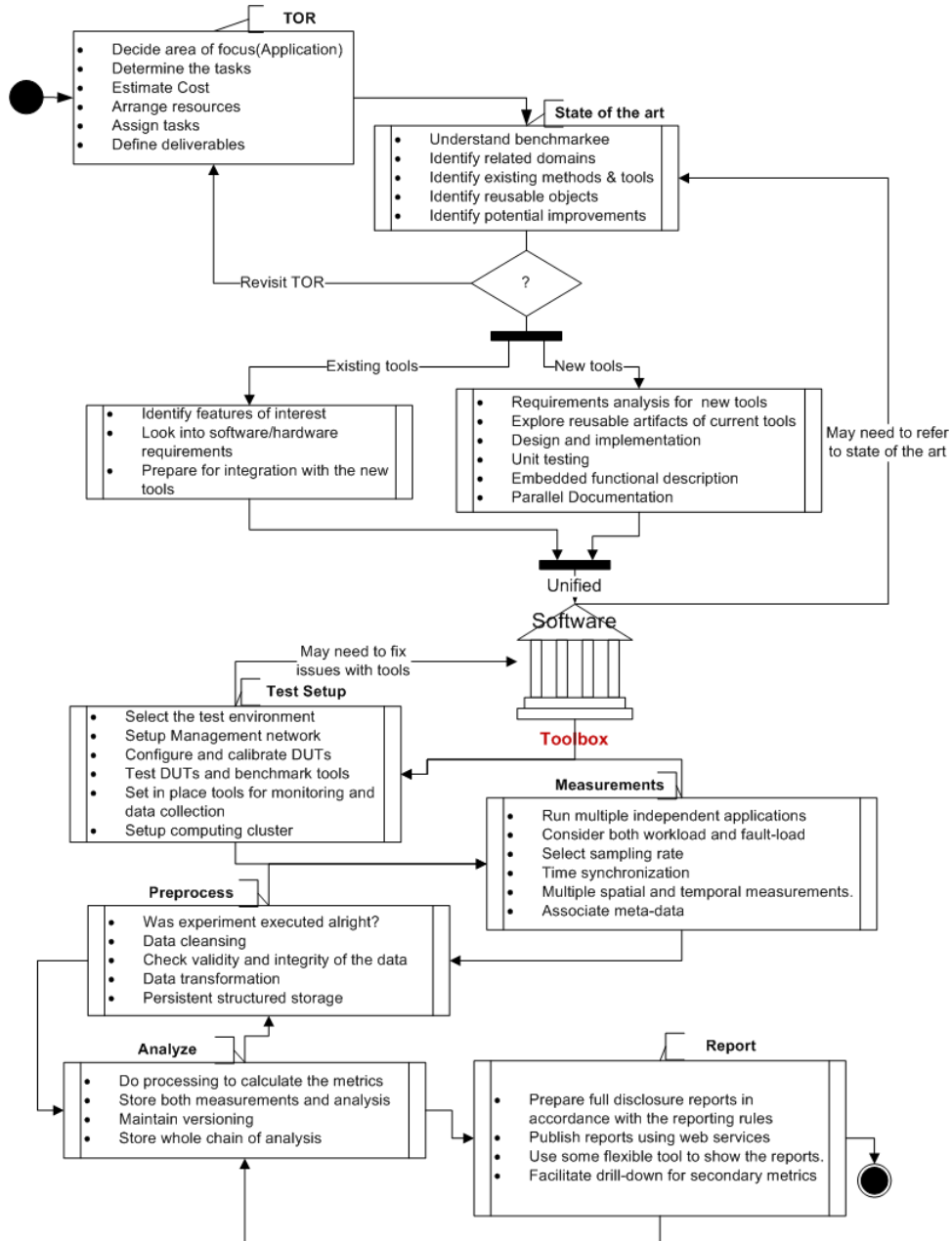


Figure 2: An instance of wireless benchmarking process

4.10 Benchmarking methodology in a nutshell

Figure 2 illustrates the flow of events in a typical benchmarking process. Each step of the process is annotated with tasks that should be accomplished before proceeding to the next step.

5 CASE STUDY

In this case study, we investigate all the steps presented in the above recommended practice for wireless benchmarking. We will restrict ourselves to the case of wireless channel characterization. Channel characterization enables researchers to investigate the influence of environment on wireless network performance and allows them to understand how well a protocol or an application performs in different wireless environments. Benchmarking provides a whole new perspective to the analysis by facilitating the identification of performance or benchmarking gaps. It makes easier the diagnosis of performance issues and provisions a better understanding on how to close the benchmarking gap. It took us around one year to investigate the wireless experimentation issues and establish the proposed benchmarking methodology. The following material is a step by step account of the process. The material required to reproduce the results in the case study is available at [33].

5.1 Plan: Terms of Reference

In this step, we lay out the foundations for the undertaking by giving a clear direction, setting a stage and provisioning the means to carry out the required benchmarking activities with efficiency. We select the wireless technology to be benchmarked, a small set of representative metrics (benchmark score), target deployment environment, resource requirements, set of tasks, output (final deliverable) and risks involved as detailed in Table 1.

5.2 Investigate: Existing best practices

We conducted an extensive analysis of the state of the art in wireless/wired networks as well as other computing and non-computing fields. The purpose was, in part, to understand the notion and utility of benchmarking in various fields. We also contemplated benchmarking jargon, concepts, practices, application scenarios and obtained interesting insight into this important paradigm which has been often undervalued in networking research. Computational benchmarks such as NPB (NAS Parallel benchmarks) [6], Standard Performance Evaluation Corporation (SPEC) [8] benchmarks, Transaction processing performance council (TPC) [7] benchmarks were looked into. Non-computational benchmarks especially those employed in the evaluation of business processes were also investigated. In fact, benchmarking forms a key component of business excellence models in today's businesses as a means to measure competitiveness. Examples are Global Benchmarking Network (GBN) [58], European Foundation for Quality Management (EFQM) [57], etc. This investigation helped us polish our benchmarking terminology vis-a-vis fine-tune benchmarking methodology for wireless networks. We also investigated practices for wireless performance evaluation [4], development of metrics [11] [12], experimentation platforms and

Table 1: Planning for wireless benchmarking

Activity	Specifications
Type of network	IEEE 802.11
Area of focus	Channel characterization
Benchmarkee	WiFi channel
Metrics	Co-channel and adjacent channel interference, K -Factor (ricean fading model), received power, RSSI, packet error rate (PER) , bit error rate (BER), packet loss
Deployment environment	Over-the-air(OTA) line-of-sight(LOS) or OTA non line-of-sight(NLOS) non-shielded indoor environment
Tasks	Network setup, cluster setup, experiment description (ED), scheduling, data collection, analysis, reporting, etc.
Resources	Hardware (Computers with PC slot, Atheros Wireless cards, Spectrum analyzers, High speed Ethernet switches, Database Server), Software(Madwifi driver, Traffic generators, sniffer, MySQL DBMS, SUN Grid Engine (clustering software), Human (Benchmark facilitators, Network Administrator, Software developer, Benchmarkmarker)
Cost	Cost of Resources
Deliverables	Metrics, Benchmark score, full disclosure reports
Risks	Bugs in drivers/sniffers, limitation of spectrum analyzers, acquisition of resources

tools as listed in Table 2 which could facilitate wireless experimentation and hence aid in benchmarking. In the end, we selected some of the existing tools to incorporate them in our experimentation platform which is demonstrated in [32].

Table 2: State of the art: Literature and Tool(suite)s

Research	References
Understanding the benchmarkee	Everything You need to know about benchmarking [1], Draft Recommended Practice for the Evaluation of 802.11 Wireless Performance [4], Framework for Performance Metrics Development [11], Strategies for sound internet measurement [14], etc.
Tools	CrunchXML [38] , Wireshark/Tshark [41], WiPal [53], Netdude [20], TCPDump [40], Iperf, Netperf [10], ttcp/nttcp/nuttcp [25–27], Nettek [24], DDoS benchmarks [28], MGEN, Kismet Spectrum tools [36], GNU Radio toolkit [39], etc.
Platforms	OMF [31], Emulab

5.3 Engineer: Benchmark tools

After stepping into the wireless experimentation arena, we have explored and put to test a number of tools in order to gauge their functional suitability, precision and correctness. It became clear to us that we need to develop some new tools, instrument/enhance existing ones and harness them all together to achieve sound wireless experimentation and the larger objective of benchmarking in the end. Some of the existing tools such as TCPDump [40], tshark/wireshark [41], sun grid engine (SGE) [45], etc., served the purpose well apart from calibrations and fine tunings. TShark/WireShark suffer from performance degradation, in case of large trace files, which we hope to overcome by employing more efficient tools such as WiPal [53]. In addition, we needed functionality to perform basic sanity checks, large scale scheduling, trace management, manipulation of wireless headers, (meta-)data management and post-processing, reporting, etc. Therefore, we indulged in the development of our own platform. Also, we developed a packet injector for traffic generation called *Direct Packet Sender* (DPS), based on BSD raw sockets [46], in order to inject packets directly at the link layer. Given in Table 3 is a list of tools that we brought together to build our wireless experimentation platform, henceforth, known as Wireless Experimentation (WEX) Toolbox [32].

Table 3: WEX Toolbox

Function	Tool
Workload/Traffic generators	Direct Packet Sender (DPS) (New), Multi-Generator or MGEN (Instrumented), Iperf
WLAN device driver	Madwifi (Instrumented)
Spectrum analyzer	Kismet Spectrum Tools (Instrumented)
Sniffer	TCPDump
Packet Analyzer	Tshark, Wireshark
Sanity checks	Unit test suite (New)
Scheduler	SGE Scheduler, Scheduler support tools (New)
Content / data Management (CM)	Data Cataloguer (New)
Database schemas	DB schema manager (New)
Database Management System (DBMS)	MySQL
Merge / Synchronization	CrunchXML(New)
Extract, Transform and Load(ETL)	ETL Scripts (new)
WEX Cluster	SGE 6.2u2_5

Amongst the tools listed in Table 3, we modified MGEN, Madwifi and kismet spectrum tools as follows:

MGEN [47] was modified to customize the packet format. We stripped off unwanted fields from the payload other than the sequence number and the timestamp. Madwifi was instrumented to disable transmission of duplicate packets in case of packet injection. We customized the format of output from kismet spectrum tools, associated timestamps with the frequency samples and inserted a code snippet to archive spectrum information.

The configurations of tools in our deployment of the platform are discussed below.

5.4 Deploy: Resource and experiment control

The experimental LAN and cluster was setup in indoor environment. The deployment details are demonstrated in [32].

5.4.1 Resource Control Setup

We setup a wired local area network (LAN) in order to manage experimentation cluster, experiment workflow and data collection. All the stations are connected to each other through gigabit switches. MyPLC [43] is used for setting up and managing the LAN computers. Fedora 10 images are prepared using vserver [44]. All the tools required on each node are bundled into this image. The image is customized for each node to allow for network configurations. The specifications of the network equipment and tools are shown in Tables 4 and 5.

Table 4: LAN setup (Hardware requirements)

Hardware	Specifications
Computers	Dell Latitude E6500 laptops
Switches	Linksys SRW2016 16-Port 10/100/1000 Gigabit Switch
Ethernet Card	Intel® 82567LM Gigabit LAN card
Wireless Card (built-in)	Intel® WiFi Link 5300 AGN
Wireless Card (External)	Atheros 5212 PCI card (For experimental wireless network)
Spectrum Analyzer	Wi-Spy 2.4x
Processor	Two x86-based Intel core duo processors (@ 2.4 GHz)
Physical memory	4 GB

Table 5: LAN setup (software requirements)

Software	Specifications
OS	Fedora 10 (Kernel 2.6.27.14)
Wireless driver	MadWifi 0.94 revision 4928
Sniffers	Tshark, tcpdump, wireshark
Network file sharing	NFS 1.1.4-1
Time synchronization	NTP 4.2.4p7 (ntp-sop.inria.fr)
Network Management	MyPLC [43]
Spectrum Analyzer	Kismet Spectrum Tools [36]
Wireless Tools	Wireless Tools for Linux version 29 [34], Compat Wireless 2.6.32 [42]

5.4.2 Experiment Control Setup

WEX Toolbox [32] employs SGE (Sun Grid Engine) [45] in order to manage scheduling and execution of tasks on the experimental cluster. The cluster consists of master and execution nodes. The functionality of the cluster is divided into two parts, namely Control Network and Experimental Network. The entire cluster, in this scenario, consists of 7 Dell Latitude E6500 laptops, but it can be extended to a large number of computers (a few hundred) quite easily because we employ MyPLC [43] to manage the network resources. MyPLC employs virtual server [44] based Fedora OS images which are installed with all the required software. Because of the centralized management, it saves the experimenter from the hassle of catering to the setup issues on individual machines. Tools employed by the cluster are described in [32]. They are grouped under *control network* or *experimental network*. In our current deployment, *control network* consists of a master node, a database server and an analysis server, whereas *experimental network* consists of one access point, one source node, one receiver, two probes and one Wi-Spy based spectrum analyzer. The cluster can easily support groups of senders, receivers, probes, spectrum analyzers, access points, etc.

Control/Management Network

It provisions command and control interface for experimental network and enables remote configurations. Also it provisions a reliable mechanism to schedule tasks and collect data (traces and meta-data) according to the run rules in distributed computing environment. Master or server node is the brain of *control network* and is used to configure and manage rest of the nodes.

Table 6: WEX cluster (server side)

Tool	Description
Scheduler	Configured to run every 8 seconds to schedule the execution of pending tasks
NTP	Network Time Protocol to ensure time synchronization
NFS server	Directories containing SGE binaries and experimentation scripts are shared on the cluster server
MySQL database server	Time sequenced unified repository for traces
Crunch XML	Export traces from intermediate XML format to database relations.
Logs	Errors, warnings, information during the course of operation of cluster.
Jobs	Experimental tasks are translated to jobs which are scheduled for execution on EN.
Java	Java version 1.6.0_17, Java TM SE Runtime Environment (build 1.6.0_17 - b04)

Experimental Network

All the nodes in *experimental network* are designated as execution nodes mainly because they run experimental tasks and applications as instructed by the scheduler daemon running on master node.

Table 7: WEX cluster (client side)

Tool	Description
SGE Execution daemon	Responsible for managing the execution of jobs on client nodes
NTP	Network Time Protocol to ensure time synchronization
NFS client	Shared directories are mounted

5.5 Configuration: The Wireless experiment scenario

Tasks in this activity may vary greatly from scenario to scenario. Therefore, the configurations laid out hereunder are specific to the scenario chosen for this case study. The focus is on capturing the characteristics of wireless medium in order to enable in depth analysis of wireless network performance under varying channel conditions. To that end, we use the packet injection technique to generate traffic at the source node, capture traffic over the selected channel using probes and monitor RF activity in the $2.4GHz$ band using Wi-Spy spectrum analyzer [35]. Often, multiple runs of a wireless experiment for the same scenario are necessary. At the end of each run, data is transferred to the content/collection server. At the end of an experimentation session, data is pre-processed, analyzed and full disclosure reports are generated. Scenario configurations consist of relative placement of nodes, software/hardware configurations, wireless interface configurations, experimentation workflow configurations, etc., as described below.

5.5.1 Placement of nodes

Around 20 nodes are positioned in 8×5 m room in a regular fashion as shown in Figure 3. The nodes used in the case study are Source (labeled in red), Server, Probe 16, Probe 21, Probe 44, Probe 49 and Wi-Spy. The relative distances between the nodes can be estimated from the room dimensions. Source, Probe 16, Probe 21, Probe 44, Probe 49 and Wi-Spy participate in the experiment.

All the nodes are placed on top of wooden tables with metal structures underneath. All of the stations are at 0.75 m height from the floor. The room is located at the top floor of a 3-storey building and is not RF isolated from the outside world. Actually, many APs are present at the different floors of the building, which makes possible to run experiments in a real working environment. As interferences are not controlled, it is crucial to be able to monitor the RF spectrum during the various experimentations.

5.5.2 Software Parameters

Sun Grid Engine version 6.2u2_5 [45] is used for scheduling experiment tasks. The scheduler is configured to periodically check the execution queues for pend-

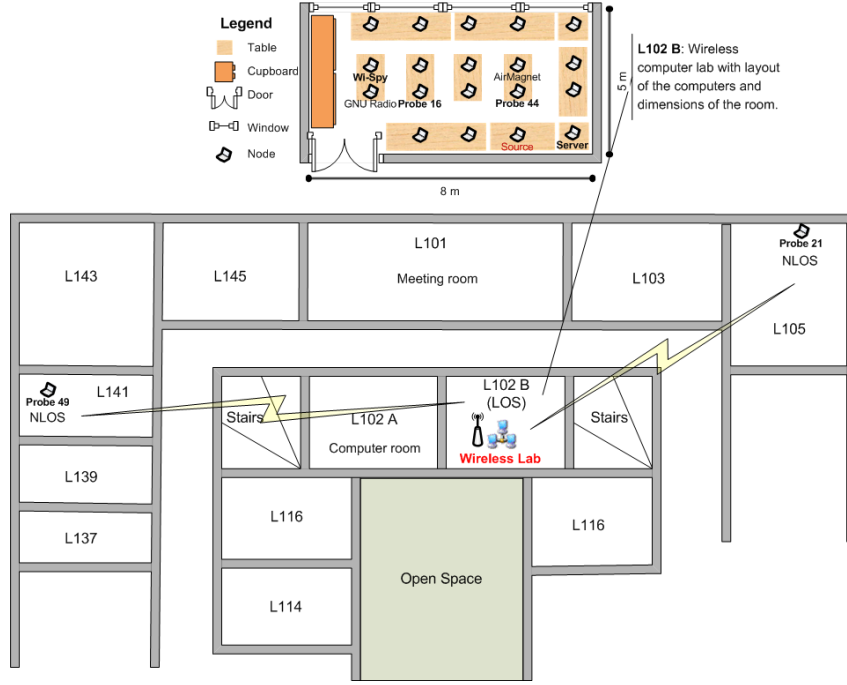


Figure 3: Indoor experimentation setup and placement of nodes

ing experimental tasks. Wireless tools for Linux [34] version 29 is used for interface configurations. Packet injection [46] is used for traffic generation. In order to harness MetaGeek’s Wi-Spy 2.4x portable USB spectrum analyzer [35].

5.5.3 Hardware Parameters

All the nodes have *x86* based architecture with 2 dual core CPUs. Each node has a total physical memory of *3.5GB* and total swap size of *1024.0MB*. Wi-Spy 2.4x is configured to scan radio activity in the entire 2.4 GHz band. We use Atheros wireless card (GWL *G650*) with Madwifi (Multimode Atheros driver for Wi-Fi on Linux) version 0.9.4 revision 4128 from the trunk.

5.5.4 Wireless Parameters

MAC and PHY revisions used by the driver are 2414 and 2413 respectively. Channel type is *11g* (operates in 2.4 GHz frequency range). Channel 11 is selected (this tells nodes to lock to the channel frequency 2.452). Fragmentation, RTS and retries are turned off. Transmission (TX) power is fixed at 18 dBm which is the maximum value for our Atheros wireless cards.

5.5.5 Reference Time duration

The total run time for an experiment is 345 seconds. Reference time duration for which results are calculated is 300 seconds.



Figure 4: Timeline of events for each run

5.5.6 Run Rules and Workflow Configurations

An experiment is formulated as a set of tasks which are configured to be executed according to a finite state machine (FSM). The workflow is as follows:

Wireless interfaces on the Source, Probe 44, and Probe 16 are configured 10 seconds after the launch of an experiment run. After waiting for another 15 seconds, TCPDump is launched on Probe 44 and Probe 16, and spectrum analyzer is launched on the Wi-Spy machine. Tcpcdump and spectrum tools are scheduled for execution for total duration of 320 seconds each. After waiting for another 10 seconds, DPS is put into action for exactly 300 seconds. DPS is terminated 10 seconds before the termination of TCPDump and spectools. The timeline of the flow of events during an experiment is demonstrated in Figure 4. Traces obtained for the first 10 and the last 10 seconds are discarded. The delays at the start and the end serve as grace periods. Long delays at the beginning are intended to allow the driver to reach steady state in terms of noise floor calibration and internal state. Also, there is an inter-run gap (i.e., pause between successive runs) when the experimentation session consists of multiple runs. The gap is set to 75 seconds.

5.5.7 Metrics

For this case study, we will measure RF interference, RSSI, Ricean K factor, Bit error rate, packet error rate and packet loss ratio at the probes.

5.6 Experiment: Undertake experiment execution and data collection

5.6.1 Launch experiment

A bootstrap python program, called primary scheduler, generates an initial schedule for all runs of the experiment. Input parameters of the primary scheduler are desired number of runs and session start time. Initial schedule is a set of startup tasks, one for each run. A startup task encapsulates information such as start time of a run and link to the scenario definition files. Startup tasks are submitted to the grid engine right away. When an startup task gets executed by the grid engine, it generates a secondary schedule based on the scenario definition. Secondary schedule formulates the state machine of the run and governs the flow of tasks. These tasks specify each and every action to be performed on the target cluster nodes. Typical actions include scenario configurations, BSS

setup, workload generation, traffic sniffing, capturing spectrum information, etc. Each task is converted as a job and submitted to the grid engine. We employ a naming convention based on timestamp and node ID to identify each run.

5.6.2 Workload generation

Our packet injector DPS enables us to generate custom payload with different sizes, formats and traffic distributions. This is important for the soundness of performance measurements. In this scenario, DPS is used to generate data packets with a payload of 100 bytes each. Packets are transmitted at the maximum rate possible. We set the link bandwidth to 1 Mbps by setting the physical bit rate of the wireless interface to 1 Mbps. This results in DPS transmitting at an effective rate of less than 1 Mbps. In order to be able calculate bit errors, we set bits in the payload to all 1's. First 8 bytes of the payload are reserved. Rest of the bytes is used for calculating bit errors per packet.

5.6.3 Trace capture

We use TCPDump to capture packet trace and spectrum analyzer's to capture RF trace.

5.7 Preprocessing: Data cleansing, archiving and transformation

We identify each trace by assigning it an identification tag based on the timestamp and node ID. At the end of each run, traces are collected at the server. However, preprocessing is deferred until the end of entire experimentation session. This makes it easier to manage the traces. We filter out unwanted extraneous packets to save space, speed up packet transfer to database and later on decrease analysis time. Extraneous packets are the ones originating from other wireless networks deployed in the vicinity of wireless experimental setup. Traces are exported to an intermediate XML format which is then used to filter out relevant packet fields to MySQL database on a database server using CrunchXML [38].

5.8 Analyze: System performance

We implemented various scripts and programs to analyze packet traces. Analysis code is an ensemble of C++ programs, python and SQL scripts. An effort was made to avoid maintaining intermediate states and data. This means that analysis is performed on the actual data store each and every time. This practice facilitates reproducible analysis. In this section, we explain selected metrics and the mechanism to calculate each of them.

5.8.1 Channel interference and RF activity in 2.4 GHz band:

Because the radio spectrum used by wireless LAN is freely available for public and research use, it is usually highly congested. Interference can be caused by not only wireless networks but also by devices such as cordless phones, Bluetooth, microwave etc., using the same channel or channels adjacent to the selected communication channel. The purpose is to capture frequency fluctuations

in the entire wireless spectrum of either 2.4 GHz band (or at least adjacent channels) and study the impact of the level of interference on performance metrics such as BER/PER, packet loss, etc.

Spectools [36] is configured to log frequency fluctuations for 2.4 GHz band. It collects information consisting of frequency range 2.400 to 2.483 at 419 points with a step size of 119 kHz. The rate at which it can capture samples depends on the processing time, called sweep time, for each sample. We have observed that it takes more than 500 ms to process one RF sample. The trace file is a sequence of tuples of the form time, frequency, amplitude. Using this trace file, one can plot a variety of graphs, e.g., frequency vs. amplitude, amplitude vs. frequency vs. time, frequency vs. running, average and peak amplitudes, etc.

5.8.2 Received Signal Strength Indicator (RSSI)

RSSI is a measure of power present in the RF signal. RSSI implementation varies from vendor to vendor. In madwifi, RSSI is equivalent to signal-to-noise ratio (SNR) and essentially is a measure of signal power above the noise floor. It is calculated for each packet by subtracting noise power from the received signal power.

5.8.3 Ricean K Factor:

Ricean K Factor is one of several measures of wireless channel characterization. K factor completely defines Ricean distribution. The higher K is, the less signal fading is. Rayleigh distribution is a special case of Ricean distribution. When the direct LOS or dominant component between the transmitter and the receiver disappears, K approaches 0 and Ricean distribution degenerates to Rayleigh distribution. We estimate K factor from empirical data. We employ a moment based method to estimate K [37]. K is obtained using the following equation

$$K = \frac{\sqrt{1-\gamma}}{1-\sqrt{1-\gamma}} \quad (1)$$

where $\gamma = V[R^2]/(E[R^2])^2$, with $V[.]$ denoting the variance, $E[.]$ denoting the expectation and R denoting the received signal envelope.

We developed both Matlab and SQL based scripts for estimating the K factor. Received power measurements are extracted from the received packets. Wireless interface measures the power in dBm which is a logarithmic scale. We convert the power measurements into Watts, normalize and then apply the formula 1.

5.8.4 Bit Error Rate (BER):

BER is a very important metric which shows the efficiency and resilience of the entire communication path including transceivers and the communication channel. BER is calculated as the number of bits flipped in each packet having bad CRC flag. We consider only the payload bits and any bits flipped in the packet headers are not accounted for.

5.8.5 Packet Error Rate (PER):

PER is the ratio of number of packets received with CRC errors to the total number of packets sent.

5.8.6 Packet Loss Ratio:

Packet loss is the number of packets that fail to reach the destination. In order to make it more discernible, we report *Packet loss ratio* which is the ratio of the number of packets lost to the total number of packets transmitted. Number of lost packets is determined from the sequence numbers of correct received packets.

5.9 Report: Benchmarking score

This section elaborates benchmarking score or the result set along with the meta data necessary to fully describe and possibly reproduce an experiment. Subsection 5.9.1 highlights full disclosure report (FDR). Subsequent subsection demonstrates the plots for metrics explained in section 5.8.

5.9.1 Reporting Rules:

Table 8 shows what could be included in the full disclosure report. The specific details about individual parameters are provided in aforementioned configurations.

Table 8: Full Disclosure Report (FDR)

Report Item	Details
LAN parameters	Hardware configurations Software configurations
Scenario parameters	Topology (placement) Landscape Software parameters Hardware parameters Wireless parameters Run time and reference time duration Workload parameters
Metrics	Channel Interference (Frequency vs. Amplitude) Received Signal Strength Indicator (RSSI) K Factor $[0 \rightarrow \infty]$ Bit Error Rate Packet Error Rate Packet Loss Ratio

To give the reader an insight of how an eventual FDR might look like, we have prepared an FDR for one of the metrics, namely K factor, which is available at [33]. Similar reports would be desirable for all the metrics chosen in any wireless benchmarking undertaking. Ideally, they should be dynamically generated from a central trace repository such as CRAWDAD [52] in response

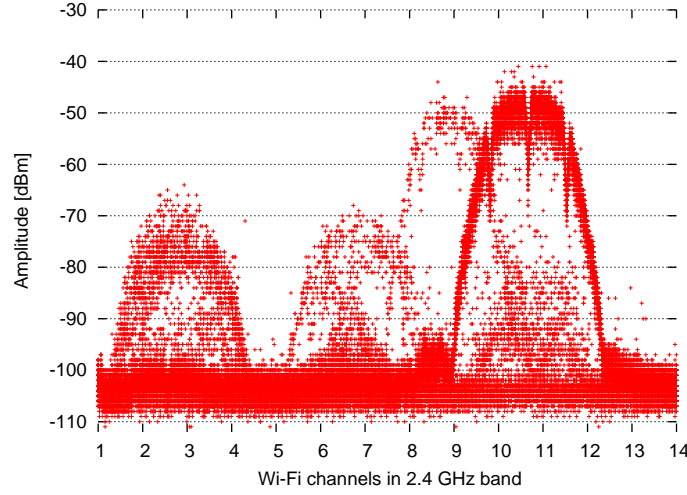


Figure 5: Spectrum analysis: adjacent and co-channel interference

to the user request. They should be made accessible to the broader research community in academia and industry through a distribution technology such as web services. Caching may be used to improve performance.

All the metrics mentioned in Table 8 except channel interference are averaged over 5 runs (corresponding to one session). In order to show the extent of variations around the average metric score, corresponding confidence intervals are also computed for a confidence level of 96% and are plotted for each score.

5.9.2 Channel interference and RF activity in 2.4 GHz band:

The RF landscape in 2.4 GHz wireless band during the course of one experiment run is shown in the Figure 5. The bandwidth of the 2.4 GHz band is 83 MHz i.e., [2400 MHz, 2483 MHz]. IEEE 802.11 divides the band into 14 channels, analogously to how radio and TV channels are sub-divided. All the channels are 22 MHz wide but spaced only 5 MHz apart. Spectrum information captured by Wi-Spy spectrum analyzer is in the form of frequency vs. amplitude. For graphical demonstration, the entire band [2400 - 2483 MHz] was mapped to the corresponding 14 WiFi channels.

5.9.3 Ricean K Factor:

Figure 6 shows K factor values as measured on two LOS (line of Sight) probes named Probe 44 and Probe 16 and two NLOS (Non LOS) probes named Probe 21 and Probe 49. Large value of K signifies less scattering and reflections caused by surrounding objects/walls and hence smaller level of multipath fading which is the case for Probes 21, 44 and 49. Small value of K means greater depth of fading which is the case for Probe 16. Contrary to RSSI which has strong dependence on distance, multipath fading depends on location, orientation with respect to the obstructions in the environment.

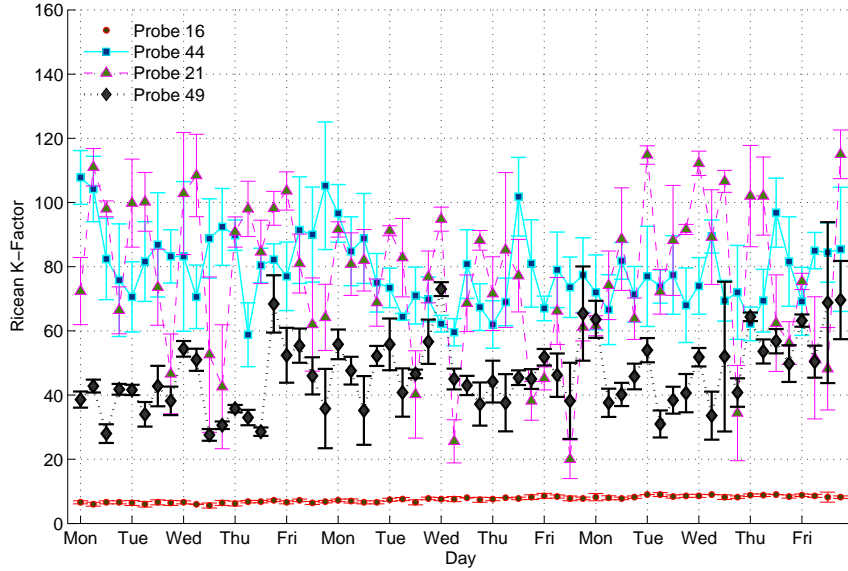


Figure 6: Average K -factor on each probe during each session. There are 4 sessions per day and each x-axis tick mark represents the first session in the sequence.

5.9.4 Bit Error Rate (BER):

As shown in Figure 7, bit error rate is more stable on Probe 16 and Probe 44 compared to probes 21 and 49 which are NLOS. Even between Probes 21 and 49, Probe 49 experiences higher bit errors the reason being Probe 49 is farther plus there is greater human activity in the room.

5.9.5 Packet Error Rate (PER):

Figure 8 demonstrates average packet error rate as experienced by the probes. PER is more reliable metric than BER because it gives an accurate measurement of how many packets were corrupted during each experiment run. Same explanation as provided for Figure 8 in section 5.9.4 applies to PER in Figure 8.

5.9.6 Packet loss ratio:

Packet loss incurred at each probe demonstrated in Figure 9. Probes 49 and 21 experienced greater packet loss and more variations in packet loss compared to Probes 16 and 44. It is interesting to note that packet loss ratio is roughly 10 times greater than PER. This means that, in the real-world wireless, corrupted packets make only a small part of the lost packets.

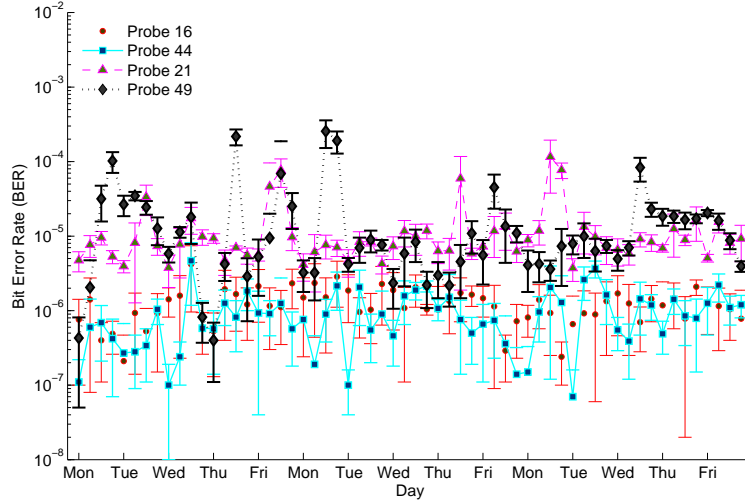


Figure 7: Average BER on each probe for each session. There are 4 sessions per day and each x-axis tick mark represents the first session in the sequence.

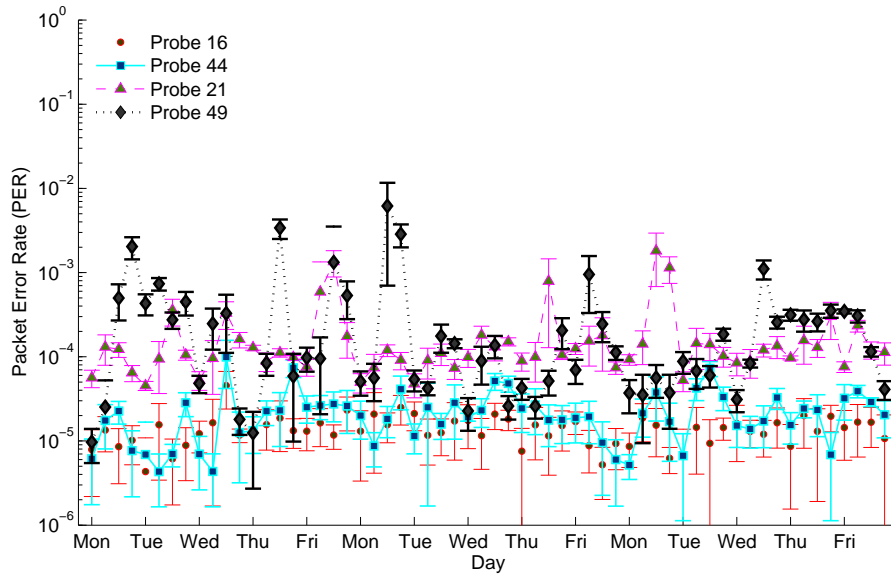


Figure 8: Average PER on each probe for each session. There are 4 sessions per day and each x-axis tick mark represents the first session in the sequence.

6 Conclusion

Benchmarking is a very powerful tool for in-depth objective performance evaluation of wireless network, troubleshooting and management. Benchmarking

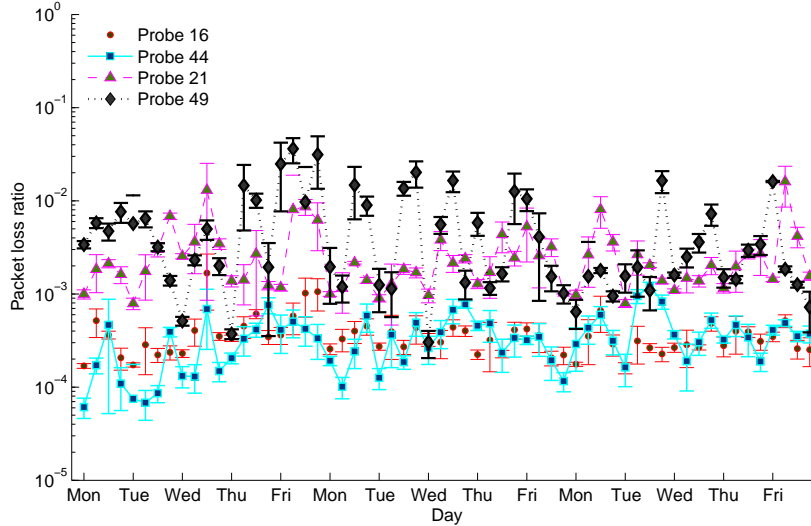


Figure 9: Average Packet loss ratio for each session on each probe. There are 4 experiments (representing 4 sessions) each day. Each x-axis tick mark represents the first experiment in the sequence.

results in value-add and competitiveness by facilitating the selection, adaptation and deployment of best applications and protocols.

However, the potential of benchmarking hasn't yet been realized to its fullest, the reason being inherent complexities in the wireless network and the test environments, lack of best practices and software tools. Over the last few years experimentation has evolved as the de-facto evaluation methodology for wireless networks. We have presented in-depth benchmarking methodology accompanied by practical demonstration of an entire benchmarking cycle. We hope that this will encourage the research community to improve/develop benchmark tools and foster a more collaborative approach by sharing and publishing benchmarks, full disclosure reports and practices. This will make it easier for benchmarkers to conduct large scale experiments with greater control, foster greater collaboration, provide an impetus for objective performance evaluation and hence, proliferate research activities and enhance the ability to innovate.

Acknowledgements

This work was supported by European Community's Seventh Framework Program (FP7/2007-2013) under grant agreement no.224263.

References

- [1] Robbin Mann, "Everything You need to know about benchmarking," 3rd International Benchmarking Conference, October 9-10, 2008, Budapest, Hungary

- [2] Camp, R. (1989). "Benchmarking. The Search for Industry Best Practices That Lead to Superior Performance," Productivity Press
- [3] RFC 2544. "Benchmarking Methodology for Network Interconnect devices," March 1999
- [4] P802.11.2/D1.01. "Draft Recommended Practice for the Evaluation of 802.11 Wireless Performance," Feb 2008
- [5] D.J. CORBETT, A.G. Ruzelli, D. Averitt, G. O'HARE, "A procedure for benchmarking MAC protocols used in wireless sensor networks." Technical Report, School of Information Technologies, the University of Sydney, August 2006.
- [6] NAS Parallel Benchmarks (NPB),
<http://www.nas.nasa.gov/Resources/Software/npb.html>
- [7] Transaction Processing Performance council (TPC), <http://www.tpc.org/>
- [8] Standard Performance Evaluation Corporation (SPEC),
<http://www.spec.org/>
- [9] A. Kashyap, S. Ganguly, S.R. Das, "Measurement-Based Approaches for Accurate Simulation of 802.11-based Wireless Networks," MSWiM 08: Proceedings of the 11-th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems Vancouver, BC, Canada, October 2008.
- [10] Netperf: A network performance benchmark, <http://www.netperf.org>
- [11] Internet draft, "Framework for Performance Metrics Development," November 2008.
- [12] Internet draft, "Reporting IP Performance Metrics to Users," July 2008.
- [13] Manolescu et al. "The repeatability experiment of SIGMOD 2008", SIGMOD Record, 37(1):39–45, Mar. 2008.
- [14] Vern Paxson, "Strategies for sound internet measurement," Internet Measurement Conference , October 26, 2004, Italy.
- [15] Benchmarking Methodology Working Group (BMWG),
<http://www.ietf.org/proceedings/95apr/ops/bmwg.html>
- [16] Extensible Markup Language (XML), <http://www.w3.org/XML/>
- [17] Internet Draft, "Information Model and XML Data Model for Traceroute Measurements," October 23, 2008.
- [18] RFC 4741, " NETCONF Configuration Protocol," December 2006
- [19] Extensible business reporting language (XBRL), <http://www.xbrl.org>
- [20] Network dump data displayer and Editor (NetDude),
<http://netdude.sourceforge.net/>

- [21] A Community Resource for Archiving Wireless Data at Dartmouth.
<http://crawdad.cs.dartmouth.edu>
- [22] Kannan Srinivasan, Maria A. Kazandjieva, Mayank Jain, Edward Kim, Philip Levis, "Demo Abstract: SWAT: Enabling Wireless Network Measurements," ACM SenSys, Nov 5-7, 2008, Raleigh, NC, USA.
- [23] Maximilian Ott, Ivan Seskar, Robert Siraccusa, Manpreet Singh, "Orbit Testbed Software Architecture: Supporting Experiments as a service," Testbeds and Research Infrastructures for the DEvelopment of NeTworks and COMmunities (Tridentcom), Feb 23-25 2005, Trento, Italy.
- [24] Secure Network Testing and monitoring,
<http://acs.lbl.gov/~boverhof/nettest.html>
- [25] Test TCP (TTCP) benchmarking tool for Measuring TCP and UDP Performance, <http://www.pcausa.com/Utilities/pcattcp.htm>
- [26] New TTCP (NTTCP), <http://linux.die.net/man/1/nttcp>
- [27] NUTTCP-Network performance measurement tool,
<http://www.lcp.nrl.navy.mil/nuttcp/nuttcp.html>
- [28] Erinc Arikan, "Attack Profiling for DDoS Benchmarks," MS Thesis, University of Delaware, August 2006.
- [29] Sachin Ganu, Haris Kremo, Richard Howard, Ivan Seskar, "Addressing Repeatability in Wireless Experiments using ORBIT Testbed," Testbeds and Research Infrastructures for the DEvelopment of NeTworks and COMmunities (Tridentcom), Feb 23-25 2005.
- [30] Wolfgang Kiess, "On Real-world Experiments With Wireless Multihop Networks," PhD dissertation, 2008.
- [31] The OMF Testbed Control, Measurement and Management Framework,
<http://omf.mytestbed.net>
- [32] WEX Toolbox,
<https://twiki-sop.inria.fr/twiki/bin/view/Projets/Planete/WEXToolkit>
- [33] Channel characterization using WEX Toolbox,
<https://twiki-sop.inria.fr/twiki/bin/view/Projets/Planete/ChannelCharacterization>
- [34] Wireless Tools for Linux,
http://www.hpl.hp.com/personal/Jean_Tourrilhes/Linux/Tools.html
- [35] Wi-Spy 2.4x, <http://www.metageek.net/products/wi-spy-24x>
- [36] Kismet Spectrum Tools, <http://www.kismetwireless.net/spectools/>
- [37] A. Abdi, C. Tepedelenlioglu, G. B. Giannakis, and M. Kaveh, "On the estimation of the K parameter for the Rice fading distribution," IEEE Commun. Lett., vol. 5, pp. 92-94, Mar. 2001.

- [38] Bilel Ben romdhanne, Diego Dujovne, and Thierry Turetletti, "Efficient and Scalable Merging Algorithms for Wireless Traces", ROADS'09, October 14, 2009, Big Sky, Montana, USA.
- [39] GNU Radio, <http://gnuradio.org/redmine/wiki/gnuradio>
- [40] TCPDump, <http://www.tcpdump.org/>
- [41] Wireshark, <http://www.tcpdump.org/>
- [42] Stable compat-wireless releases,
<http://wireless.kernel.org/en/users/Download/stable/>
- [43] PlanetLab MyPLC, <https://svn.planet-lab.org/wiki/MyPLCUserGuide>
- [44] vserver capable kernel, <http://svn.planet-lab.org/wiki/VserverCentos>
- [45] Sun Grid Engine, <http://gridengine.sunsource.net/>
- [46] Packet Injection and Sniffing using Raw Sockets,
<http://security-freak.net/raw-sockets/raw-sockets.html>
- [47] Multi-Generator(MGEN), <http://cs.itd.nrl.navy.mil/work/mgen/index.php>
- [48] Glenn Judd, and Peter Steenkiste, "Repeatable and Realistic Wireless Experimentation through Physical Emulation," In HotNets-II, Cambridge, MA, November 2003. ACM.
- [49] D. Kotz, C. Newport, R. S. Gray, J. Liu, Y. Yuan, and C. Elliott, "Experimental evaluation of wireless simulation assumptions," MSWiM '04: Proceedings of the 7th ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems. New York, NY, USA: ACM, 2004, pp. 78-82.
- [50] Internet Measurement Data Catalog, <http://www.datcat.org/>
- [51] Extensible Markup Language (XML), <http://www.w3.org/XML/>
- [52] A Community Resource for Archiving Wireless Data,
<http://crawdad.cs.dartmouth.edu/>
- [53] WiPal: IEEE 802.11 traces manipulation software,
<http://wipal.lip6.fr/>
- [54] D. G. Andersen and N. Feamster, "Challenges and opportunities in Internet data mining", Technical Report CMU-PDL-06-102, Carnegie Mellon University Parallel Data Laboratory, Jan. 2006.
- [55] University of Utah Flux Research Group, "Emulab: The Utah Network Emulation Testbed", <http://www.emulab.net/>
- [56] E. B. Hamida, G. Chelius and G. M. Gorce, "Impact of the physical layer modelling on the accuracy and scalability of Wireless Network Simulation," SIMULATION, September, 2009.
- [57] European Foundation for Quality Management, <http://www.efqm.org/>

- [58] Global Benchmarking Network (GBN), <http://www.globalbenchmarking.org/>
- [59] H. Lundgren, D. Lundberg, J. Nielsen, E. Nordström, and C. Tschudin, "A large-scale testbed for reproducible Ad Hoc protocol evaluations," In Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC), pages 337-343, March 2002.
- [60] ISI, University of Southern California. The network simulator - ns-2. <http://www.isi.edu/nsnam/ns/>.
- [61] D. Johnson, T. Stack, R. Fish, D. M. Flickinger, L. Stoller, R. Ricci, and J. Lepreau, "Mobile emulab: A robotic wireless and sensor network testbed," INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings, pp. 1-12, April 2006.
- [62] P. De, A. Raniwala, S. Sharma, and T. Chiueh, "Mint: a miniaturized network testbed for mobile wireless research," INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE, vol. 4, pp. 2731-2742 vol. 4, March 2005.
- [63] P. De, A. Raniwala, R. Krishnan, K. Tatavarthi, J. Modi, N. A. Syed, S. Sharma, and T. cker Chiueh, "Mint-m: an autonomous mobile wireless experimentation platform," in MobiSys '06: Proceedings of the 4th international conference on Mobile systems, applications and services, (New York, NY, USA), pp. 124-137, ACM, 2006.
- [64] J. Zhou, Z. Ji, M. Varshney, Z. Xu, Y. Yang, M. Marina, and R. Bagrodia. Whynet: a hybrid testbed for large-scale, heterogeneous and adaptive wireless networks. In WiNTECH '06: Proceedings of the 1st international workshop on Wireless network testbeds, experimental evaluation and characterization, pages 111-112, New York, NY, USA, 2006. ACM.
- [65] G. Jourjon, T. Rakotoarivelo, C. Dwertmann, M. Ott, *Executable Paper Challenge: LabWiki: an Executable Paper Platform for Experiment-based Research*, Procedia Computer Science, 2011.
- [66] labwiki, <http://omf.mytestbed.net/projects/omf/wiki>
- [67] Executable Paper challenge, <http://www.executablepapers.com/about-challenge.html>
- [68] Guillaume Jourjon, Thierry Rakotoarivelo, Max Ott, *A Portal to Support Rigorous Experimental Methodology in Networking Research*, 17-19 April 2011, Shanghai, China.
- [69] Cristian Tala, Luciano Ahumada, Diego Dujovne, Shafqat-Ur Rehman, Thierry Turetletti, and Walid Dabbous, *Guidelines for the accuratedesign of empiricalstudiesin wirelessnetworks*, 7th International ICST Conferenceon Testbedsand ResearchInfrastructures for the Developmentof Networks and Communities(TridentCom2011), April 2011, Shanghai, China.
- [70] Network Experiment Programming Interface, <http://yans.pl.sophia.inria.fr/trac/nepi/wiki/nepi>



Centre de recherche INRIA Sophia Antipolis – Méditerranée
2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Centre de recherche INRIA Bordeaux – Sud Ouest : Domaine Universitaire - 351, cours de la Libération - 33405 Talence Cedex
Centre de recherche INRIA Grenoble – Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier
Centre de recherche INRIA Lille – Nord Europe : Parc Scientifique de la Haute Borne - 40, avenue Halley - 59650 Villeneuve d'Ascq
Centre de recherche INRIA Nancy – Grand Est : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex
Centre de recherche INRIA Paris – Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex
Centre de recherche INRIA Rennes – Bretagne Atlantique : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex
Centre de recherche INRIA Saclay – Île-de-France : Parc Orsay Université - ZAC des Vignes : 4, rue Jacques Monod - 91893 Orsay Cedex

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
<http://www.inria.fr>
ISSN 0249-6399