Benchmarking in Wireless Networks

Shafqat Ur Rehman, Thierry Turletti, Walid Dabbous

Benchmarking in Wireless Networks

Shafqat Ur Rehman, Thierry Turletti, Walid Dabbous

***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 controllable and uncontrollable hardware/software parameters and unpredictable characteristics of the wireless channel in the wild, poses significant challenges in the way of experiments 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 experiments, large amount of data and facilitate reproducible analysis further complicates the process. In this report, we present a holistic view of benchmarking in wireless networks; introduce key definitions and formulate a procedure complemented by step-by-step case study to help drive future efforts on benchmarking of wireless network applications and protocols.**

***Index Terms*—wireless networks; wireless experiments; methodology; wireless Tools; repeatability; IEEE 802.11.**

I. Introduction

WIREless technologies are increasingly being deployed en masse at the edge of Internet because of increased flexibility and cost-effectiveness. Wireless networking holds the promise of "anytime, anywhere" distributed computing because of proliferation of lightweight portable computing devices, miniaturization of wireless equipment and advances in wireless technology [6]. With the continuous development of wireless Internet as a business tool, wireless network performance has assumed greater importance. As a consequence, performance evaluation of wireless networking systems and protocols has witnessed recently 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. However, lack of realism because of simplistic radio models in stochastic simulations can lead to misleading analysis [55], [61].

The best way to achieve realism is to perform in-field experiments using 'real' hardware and software. This would provision an ultimate mechanism to address the limitations of simulation based analysis models. Unfortunately, wireless experimentation is not a smooth process and configurability and management of even a small number of nodes is cumbersome. In a real world environment, physical layer fundamentally affects operation at all layers of the protocol stack in complex ways and behavior of the physical layer is tightly coupled to the physical environment and precise conditions under which an experiment is conducted [54]. Wireless channels are unpredictable (random), error-prone and could vary over very short time scale (order of microseconds). It is also difficult to avoid affecting collocated wireless networks. The mobility of uncontrolled radio sources, physical objects, and people makes these conditions nearly impossible to repeat.

Repeatability 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 repeatable if the protocol/procedure used is described in sufficient detail along with reagents, specifications of the equipment, times, temperatures, etc. [60]. However, networking experiments can't be reproduced by such measures because the distributed software is much more complex. That is may be the reason why rigorous peer verification of experimental results is not becoming a culture in networking field as yet [27]. Wireless experiments are particulary harder because of additional complexities such as volatile radio spectrum, software/hardware imperfections/calibrations, configurability, management of resources and data, etc. [36]. If achieved, repeatability will lead researchers to archive and share data and code and hence enable future researchers to compare their experimental results with the previous ones[60].

Despite significant challenges, researchers have realized the importance of repeatable network experimentation [60], [56], [35], [54]. Recent trends in networking research indicate progress in the following directions.

- Online network measurement data repositories are being deployed [56], [57]. The purpose of these repositories is to facilitate archiving, publishing and sharing of network measurement data. Effort is also being made to develop supporting analysis tools [58], [57].
- There has been more interest in testbeds such as Emulab [59] and ORBIT [27]. These testbeds are largely focused on making it easier to control the testbed resources.
- Researchers are developing management and control tools to conduct experiments with ease and efficiency [37]. This may include automated management of software installation/updates, and other steps such as configurations, data collection, etc. Tools may also help create experiment scripts through automatic code generation using templates. Management framework may also facilitate interactive monitoring of experimental workflow and possibly modifying the experimental parameters on the fly [37].

Authors are with Plante project-team, INRIA, Sophia Antipolis, France. email: Shafqat-Ur.Rehman, Thierry.Turletti, Walid.Dabbous@inria.fr

However, these tools generally lack functionality to process wireless headers (such as radiotap headers) and depend heavily on instrumentation to accommodate third-party tools. Sanity checks, data management, analysis and publishing of experiments and results are left to the experimenter to deal with. More generally, the experimenter is required to deal with a plethora of issues as outlined below:

- Setup of the network and experimental cluster. Install the necessary hardware/software tools and manage configurations.
- Calibrate core software tools and network equipment to ensure the desired level of accuracy.
- Characterize the wireless environment to better understand conditions such as depth of fading, (co/adjacent)-channel (inter/intra)-radio interference.
- Ensure time synchronization.
- Perform sanity checks and deal with imperfections and limitations of tools.
- Occasionally need to instrument the driver and tools.
- Define scenario precisely, e.g., nature of equipment, landscape, geometric setting of nodes, configurations, workload, etc.
- Schedule and manage large number experiments.
- Capture trace and meta-data (e.g. sampling rate, traffic filters, sniffer settings,etc.) and associate it with the traces.
- Make strategy for data management(repository, associating meta-data, data cleaning), analysis procedures (scripts, data normalization, data transformations, assumptions, etc.)

In this report, we provide a roadmap for taming wireless experiments and a direction for realizing their repeatability. We promote the notion of wireless networks benchmarking which encompasses the aforementioned requirements and can provision a level playing ground for the evaluation of wireless networks. It can facilitate more in-depth investigation of intricate effects of physical layer on cross-layer design. Benchmarking in wireless networks is surprisingly complex 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 and lack of tools to deal with the issues listed above.

Benchmarking requires a highly systematic methodology which would facilitate to manage the experimentation process as a whole and to analyze in tandem all important parameters and metrics of a wireless system under test (SUT). We present practical guidelines for wireless experimentation and a strategy on how to overcome the aforementioned difficulties to achieve the objective of wireless benchmarking. The contributions of this work are as follows:

- We introduce useful terminology for wireless experimentation to make it easier for experimenters to compare and publish results.
- We formulate a practical methodology on what to do and how to do it when conducting wireless experiments.
- We present an experimental case study which provides detailed implementation of the proposed methodology and how to do experiments smoothly and easily.

The remainder of this report is organized as follows. In section II, we provide a brief overview of benchmarking and list benefits and challenges for wireless networks. Section III introduces and elaborates some useful terminology. Section IV provides detailed step-wise account of our proposed methodology for wireless experimentation and benchmarking. In section V, we provide an experimental case study which implements all aspects of the benchmarking methodology.

II. Benchmarking Analysis

Benchmarking is a well known concept in a 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 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 [7]. 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 particulary in wireless networks. The development of new applications and protocols to meet the performance and QoS requirements of emerging wireless network applications such as mesh networks, sensor networks, 4G etc., are impeded by the lack of common benchmarking standard to determine the suitability of certain protocols to certain scenarios [8].

The following subsections shed light on potential benefits and approaches for benchmarking in general, and challenges for benchmarking in wireless experiments.

A. 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.
- Benchmarks 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?”

B. Approaches

Evaluation of new network protocols and architectures is at the core of networking research. This evaluation is usually performed using mathematical analysis, simulations, emulations, or experimentation. Mathematical analysis provides tractable models that approximate the system, but that are often simplistic. Simulations allow a fast evaluation process, fully controlled scenarios, and reproducibility. However, they lack realism and are inadequate for accurate characterization of wireless protocols due to composite effects of factors such as hidden/exposed terminals, capture effect, doppler's effect, reflections and interference on upper layer protocols. Experimentation overcomes the aforementioned limitations by provisioning more realistic environment and implementations, but it lacks reproducibility and ease of use.

Simulation, nonetheless, is an essential tool and enables a researcher to quickly assess the performance of a protocol. Experimentation is usually the next step before actual deployment. It serves as the modus operandi to validate the results of simulation. Experimental results are considered more authentic as the evaluation is carried out under realistic physical conditions using real hardware.

The workload is another important aspect of the evaluation process. There are in fact two types of workloads namely synthetic and application based. Synthetic benchmarks are a set of programs which exercise a few specific aspects of system under test. They are carefully designed to statistically mimic some common set of application programs. Synthetic benchmarks are very useful in the sense that they are simpler and it is easier to replicate the same workload for spatial and temporal repeats of the test. Figure 1 shows the typical architecture of a synthetic benchmark in network environment. Synthetic benchmarks are more suitable for micro benchmarking where focus is put on a single function in order to improve that particular function or to test system's suitability for a particular application scenario, e.g., measuring the point-to-point throughput over a wireless connection, or latency of the connection. Synthetic benchmarks investigate a number of different aspects independently. These include scalability, fault-tolerance, reliability, availability, security, cost, quality,

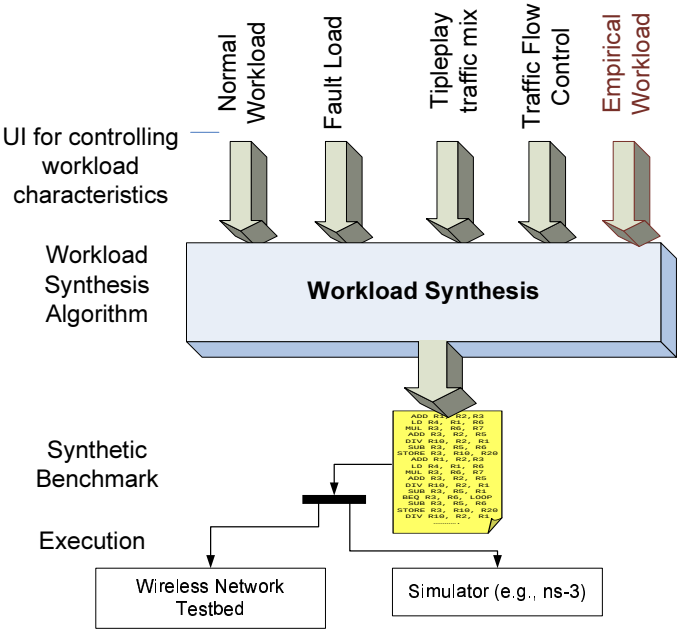

Fig. 1. Workload synthesis in synthetic benchmarking

efficiency, study of performance under varying temporal and spatial traffic patterns. The main objection that can be raised against synthetic benchmarks is that they don't represent system's performance in production environment.

Application or real-world benchmarks run real-world programs and are not aimed at producing benchmark scores. Some example programs are referred herein. ESM (End System Multicast), PPlive, SoPCast are P2P media streaming systems. These programs when used as benchmarks, e.g., for benchmarking P2P streaming over mesh networks; will be referred to as real-world benchmarks because they are actual applications that are used by users in their everyday life. Real-world benchmarks are more difficult to interpret because they are subjective in nature. However, it has to be noted that we can feed empirical workload into a synthetic benchmark to more realistically mimic real-world user applications as shown in Figure 1. Trace based simulations represent an example approach.

In our work, we focused on *wireless networks experimentation* with real application workloads. In those networks, benchmarking can be used to characterize quantitative (e.g. throughput, interference (RSSI, SNR, etc.), signal fading, bit/packet errors, etc.) or qualitative (e.g. security, fairness (inter-protocol or intra-protocol), reliability, etc.) aspects of system under test. Experiments need to be performed using well-formulated test scenarios. Performance needs to be assessed by selecting a small well-defined set of metrics. A blueprint for such an undertaking can be the metrics and best practices proposed by the Benchmarking Methodology Working Group (BMWG) [25] of IETF for the evaluation of interconnection devices and protocols in wired networks. However, there are no such defined metrics in the case of wireless networks. In the next subsection, we present the main challenges for benchmarking of wireless technologies.

C. Wireless Benchmarking Challenges

In wireless networks, radio propagation is highly dependent upon the physical environment, 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 [35].

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 [36]. 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 [40] in conjunction with kismet spectools [41] 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.

III. KEY DEFINITIONS

It is imperative to share a common understanding of the terminology in the field of wireless network measurement and benchmarking. Common terminology enables experts to express their ideas in a particularly concise fashion whereas newcomers may use it to quickly establish a comfortable familiarity with the key jargon. For this purpose, we have prepared a list of potentially important terms that would be desirable in any wireless network benchmarking venture and provide definitions for each of them. Some of these terms have been extracted from both computing and non-computing industries [1], [2], [3], [6] and redefined to fit into the context of wireless network computing.

a) Benchmarkee: It can be any wireless object (application, service, protocol, or wireless system) being benchmarked.

b) Benchmark: Benchmark means a reference point. Originally, the term referred to a mark on a workbench used to compare the lengths of pieces so as to determine whether one was shorter or longer than the desired.

In wireless networks, the term may refer to the *desired bottom-line performance* (as in QoS provisioning) i.e., reference point against which the performance of other similar wireless systems is compared. It may also refer to *the best/normal/worst case performance of a wireless system which serves as reference point for comparisons with its performance under other cases.* The definition of what constitutes a best/worst/normal case is subjective and may be decided based on channel conditions (e.g., interference, fading), workload, etc.

c) Benchmark Toolkit: Software designed to implement and carry out benchmark tests. It may consist of just one client/server tool to calculate basic metrics throughput, RTT, packet loss, etc., or it may consist of an ensemble of tools to enable more sophisticated benchmark tests .

d) Benchmarking: Benchmarking is the systematic evaluation of wireless solutions in search of the ones capable of delivering better performance to end users. It can be performed by running benchmark tool (suite) in order to assess the performance of benchmarkee relative to the reference benchmark.

e) Benchmarker: The human resources conducting the benchmarking.

f) Performance Benchmarking: It means comparing the performance data obtained from measuring similar applications or protocols.

g) Best-in-class benchmarking: It is applying performance benchmarking and identifying solutions that produce the best results in a particular class. Protocols designed to serve the same purpose are said to belong to the same class. For instance, rate adaption protocols (RAPs), handover protocols, streaming media protocols, etc., represent different classes of protocols.

h) Best-in-class: Application or protocol that is shown to produce superior results, selected by the best-in-class benchmarking and successfully demonstrated.

i) Run rules: Run rules define what constitutes a valid test with this benchmark. Usually these define valid configurations, experimental limitations, etc.

j) Metric: Metric is a measure of an observable behavior of an application, protocol or any other system. It may also refer to the measure of a particular characteristic of a protocol's performance or efficiency. Each benchmark is required to define a set of valid metrics for this particular benchmark.

k) Primary Metric: Primary metrics directly impact users' experience, e.g., voice quality.

l) Secondary Metric: Secondary metrics impact the primary metrics. For example packet loss, jitter and latency impact voice quality.

m) Metric User: The user (person or application) who needs the information supplied by the metric and will be making decisions and taking action based upon the metric. If a metric does not have a user it should not be produced.

n) Result: It is the value of a metric being reported for the benchmark.

o) Benchmark Score: It is the set of final results that is used for comparison.

p) Reference Time Duration: It is the time interval for which benchmark score is computed. It is decided by excluding configuration time, wake-up time. Generally, one would need to strip off steady state time and grace period as well.

q) Benchmarking gap: It is the difference in performance between a particular object (for example, a novel protocol) and other objects in the comparison, the measured advantage of the benchmark object over other objects.

r) Full disclosure Report (FDR): It is the complete documentation of a benchmark's score along with the meta-data (relevant system and benchmark configuration). There should be sufficient detail and coverage for someone else to be able to reproduce the tests.

s) Reporting rules: The set of benchmark rules that defines what constitutes a valid full disclosure for that benchmark. Usually these define what parts of the benchmark configuration and the system configuration(s) that need to be detailed and disclosed.

t) Repeatability: Repeatability is the "closeness of the agreement between the results of successive measurements of the same measurand or metric" [36].

u) Reproducibility:: Repeatability is prerequisite for reproducibility which implies that other benchmarkers should be able to recreate the experiments and produce comparable results.

IV. BENCHMARKING METHODOLOGY FOR WIRELESS NETWORKS

In the field of network computing, benchmarking is common for network interconnection devices [4]. Recently, IEEE LAN/MAN Standards Committee prepared a recommended practice for the evaluation of 802.11 wireless networks [6]. Recommendations in [6] are valuable for the benchmarking methodology presented herein.

Figure 2 outlines the set of activities envisioned for wireless network benchmarking. 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 and tools. Then there is a cycle of activities as shown by circulating arrows in Figure 2. 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, 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.

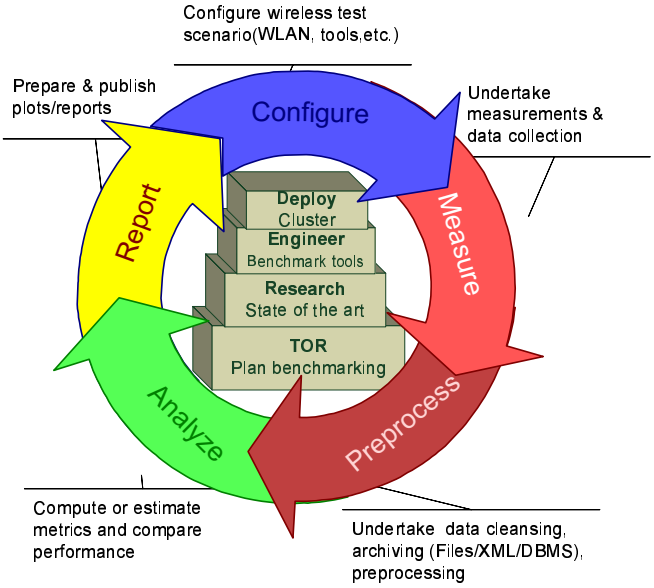

Fig. 2. Wireless Network Benchmarking

A. 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., Wi-Fi, 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 considerations. 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.

B. Research: State of the art

Research on the current state of benchmarking and evaluation paradigms across domains relevant to benchmarkee [5] 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, he/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.

C. 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, add-ins, 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 just-in-time 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 functional. A system documentation is necessary to describe the functionality, limitations, direction for future enhancements, dependencies and installation guidelines for the deliverable tools.

D. Deploy: Network and Computing Cluster

The pre-requisite for benchmark tools 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 are 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 [4].

Deployment involves setting in place the network equipment, installing the required software. Deploying a computing cluster is also desirable in order to manage the execution of experimental tasks on the set of nodes participating in the wireless experiment. It also empowers the benchmarker to perform multiple runs faster and efficiently. Multiple runs are necessary in order to have confidence in the measurements. 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.

System specifications of DUT e.g., Chipset, OS Kernel version, CPU, RAM, SWAP size, WLAN driver, firmware version have to be documented. Versions and configurations of all the tools (wireless sniffers, spectrum analyzers, calibration tools, traffic generators, data access technologies etc) have to be documented. Network settings e.g., SSID, security usage (WEP, TKIP, CCMP etc.), traffic signal levels (transmit power, receiving sensitivity), frame size, frame space, radio distance (increasing/decreasing), RTS/CTS usage (enabled or disabled)/threshold, number of STAs generating the traffic, traffic direction etc. have to be documented. Network parameters such as topology, size and capacities have to be listed. Typical device setup settings that should be configured, measured and reported are antenna type, antenna diversity, channel, transmission power, RTS/Fragmentation threshold, MAC/PHY revisions and security settings. Key to successful benchmarking is holding as many parameters as possible constant in order to isolate the contribution of specific elements being compared.

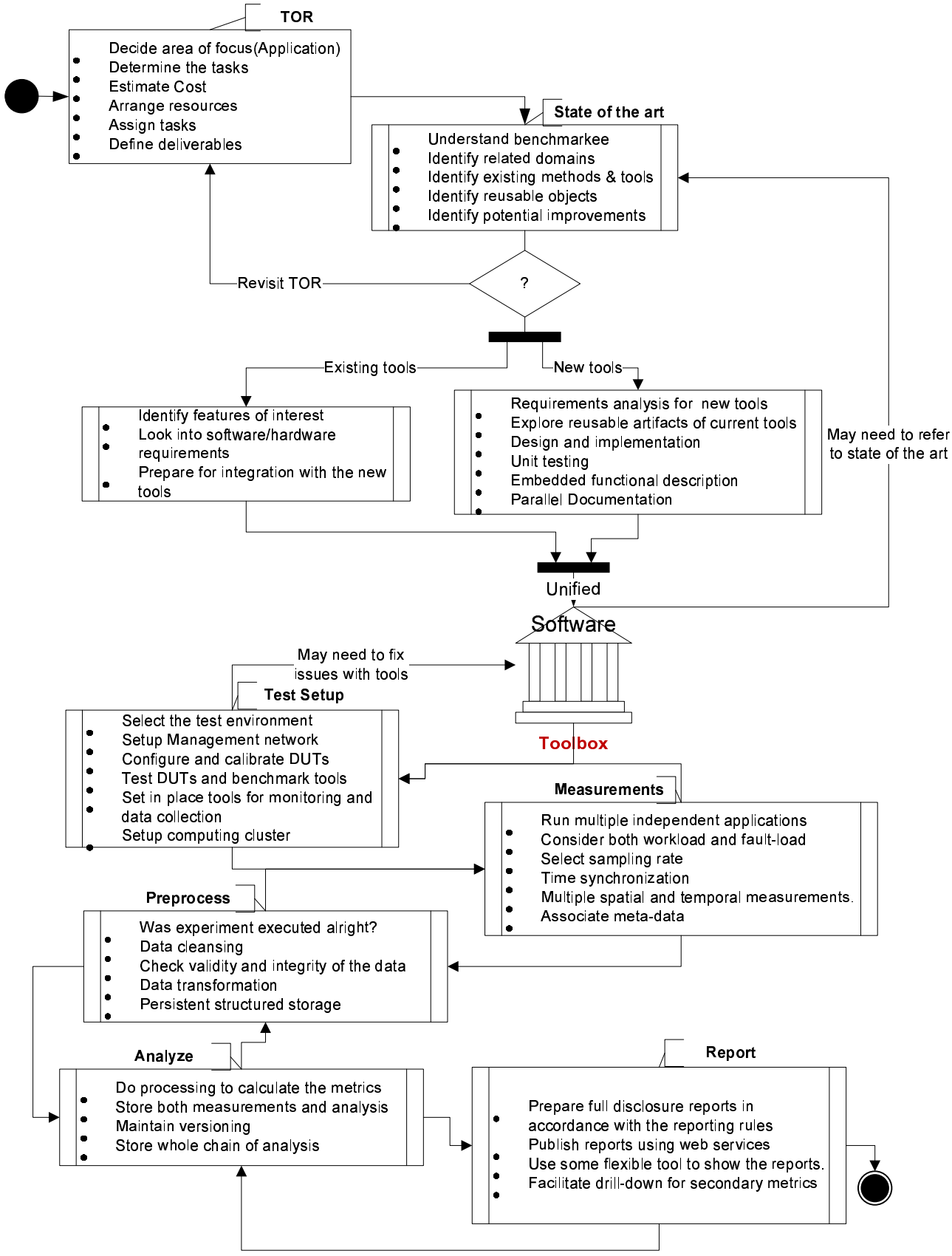

Fig. 3. An instance of wireless benchmarking process

E. Configure: Wireless Test

Configurations elaborated in section IV-D are general and are concerned with the LAN and 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, a wireless experimental network has to be setup and configured. All the tools necessary to carry out the tasks specified in the experimentation scenario have to be configured. This is usually repeated for each run of the experiment to ensure a clean start.

F. Measure: Undertake 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 [13]. 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 [4].

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 reproducible analysis.

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 [25]. 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.

G. Preprocess: Data cleansing, archiving and transformation

The data collected in the measurement 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 and database management systems (DBMS). One such example is CRAWDAD [24]. Meta data (interference, variable resources, etc.) should also be associated with the traces.

H. Analysis

Finally, it 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 [25]. 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.

I. Report

Reports are the windows through which metric customers can gain a visual access to the benchmark 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. 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.

J. Benchmarking methodology in a nutshell

Figure 3 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. It may be required to revisit the previous steps in order to address issues that surfaced at a later stage.

V. 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, especially the cross-layer impact, 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 implement the case study and investigate the wireless experimentation issues. The following material is a step by step account of the process.

A. Terms of Reference(Benchmarking Plan)

TABLE I
PLANNING FOR WIRELESS BENCHMARKING

Activity	Specs
Type of network	IEEE802.11
Area of focus	Channel characterization
Benchmarkee	WiFi channel
Metrics	BER, co-channel and adjacent channel interference, K-Factor (ricean Fading Model), 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, 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, Benchmarker)
Cost	Cost of Resources
Deliverables	Metrics, Benchmark score, full disclosure reports
Risks	Bugs in drivers/sniffers, limitation of spectrum analyzers, acquisition of resources

B. State of the art

Table II includes the list of papers relevant for the state of the art.

TABLE II
STATE OF THE ART: LITERATURE AND TOOL(SUITE)S

Research	References
Understanding the benchmarkee	Everything You need to know about benchmarking [7], Draft Recommended Practice for the Evaluation of 802.11 Wireless Performance [6], Framework for Performance Metrics Development [13], Strategies for sound internet measurement [19], etc.
Tools	CrunchXML [43] , Wireshark/Tshark [46], Netdude [20], TCPdump [45], Iperf, Netperf [10], ttcp/nttcp/nuttcp [30], [31], [32], Nettest [29], DDoS benchmarks [33], MGEN, Kismet Spectrum tools [41], GNU Radio toolkit [44], etc.
Platforms	OMF [37], Emulab

C. 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 existing tools served the purpose well apart from calibrations and fine tunings. Given in Table III is a list of tools that we brought together to build our experimentation platform, henceforth, known as Wireless Experimentation (WEX) Toolkit [38].

TABLE III
WEX TOOLKIT

Function	Tool
Workload/Traffic generators	Packet injector (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

The details of our modifications and improvements are as follows:

- MGEN [53] 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.

D. Deploy (LAN and Computing Cluster)

The experimental LAN and cluster was setup in indoor environment. The deployment details are demonstrated in Figure 4.

1) LAN setup: We setup a wired local area network (LAN) in order to manage experimentation cluster, experiment workflow and data collection. All the computers in the wireless laboratory are connected to each other through gigabit switches. MyPLC [49] is used for setting up and managing the LAN computers. We prepare Fedora 10 images using vserver [50]. 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 IV and V.

TABLE IV
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 V
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 [49]
Spectrum Analyzer	Kismet Spectrum Tools [41]
Wireless Tools	Wireless Tools version 29 [39], Compat Wireless 2.6.32 [48]

2) Cluster setup: The WEX Toolkit takes advantage of SGE (Sun Grid Engine) [51] in order to setup and manage the wireless experimentation cluster. The cluster consists of master and execution nodes. The functionality of the cluster is divided into two parts, namely Control Network (CN) and Experimental Network (EN). 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 thousand) quite easily. Tools employed by the cluster are highlighted in Figure 4. They are grouped under CN or EN. In our current deployment, CN consists of a master node, a database server and an analysis server, whereas EN 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 (CN): It provisions command and control interface for experimental network (EN) 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 CN and is used to configure and manage rest of the nodes.

TABLE VI
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 db 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 SE Runtime Environment (build $1.6.0\_17 - b04$)

Experimental Network (EN): All the nodes in EN are designated as execution nodes mainly because they run experimental tasks and applications as instructed by the scheduler daemon running on master node.

TABLE VII
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

E. 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 (air) in order to enable in depth analysis of wireless network performance under varying channel conditions. To that end, we use a packet injector to generate traffic at the source node, capture traffic over the selected channel using probes

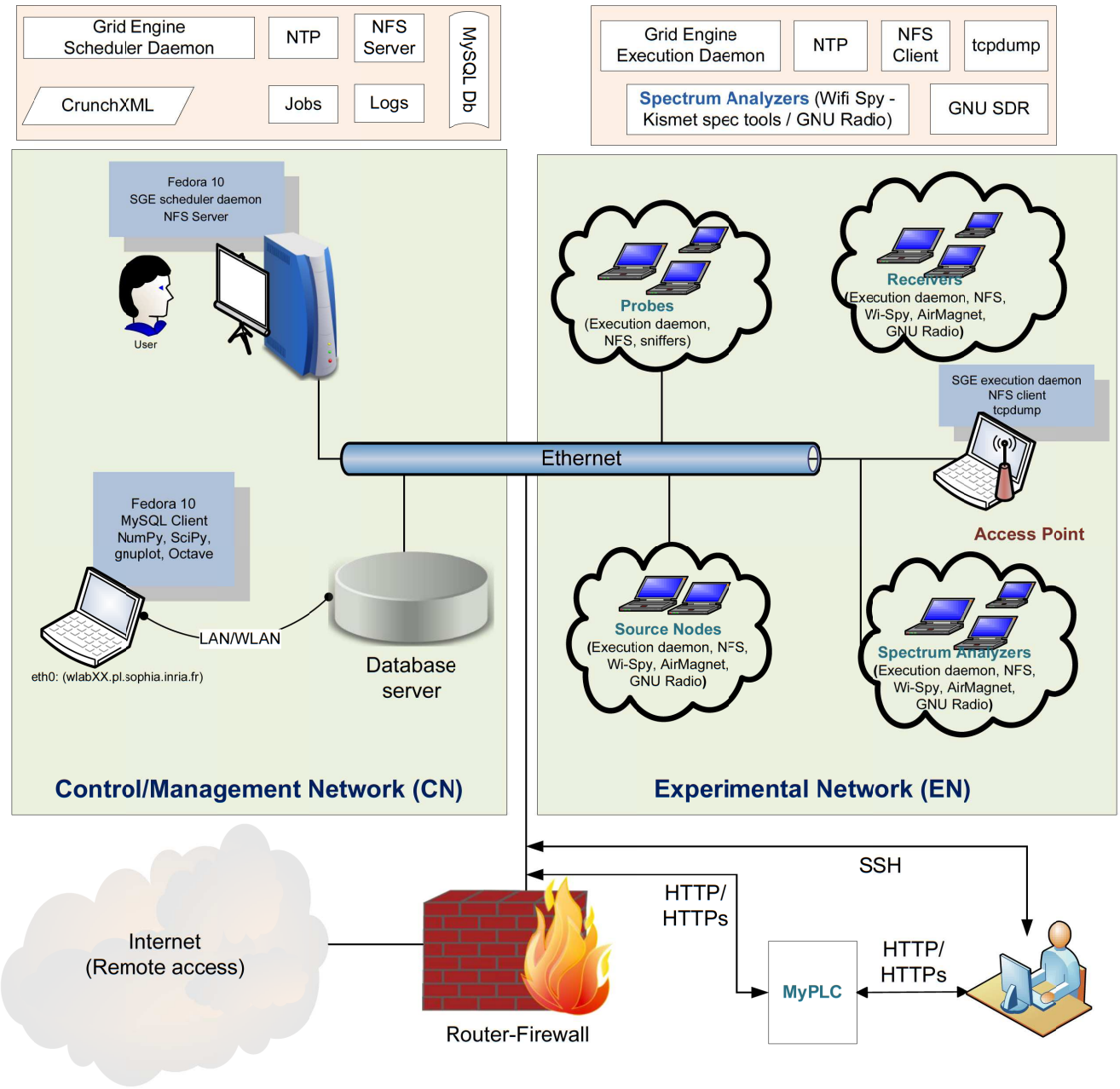

Fig. 4. WEX Toolbox Architecture and deployment

and monitor RF activity in the $2.4GHz$ band using Wi-Spy spectrum analyzer [40]. Most 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 preprocessed, analyzed and full disclosure reports are generated.

1) Physical positioning of nodes: Around 20 nodes are positioned in 8×5 m room in a regular fashion as shown in Figure 5. The nodes used in the case study are Source (labeled in red), Server, Monitor1, Monitor2 and Wi-Spy. The relative distances between the nodes can be estimated from the room dimensions. Please note that only Server is shown to belong to CN. Database server and analysis server are located elsewhere. Source, Monitor1, Monitor2, Wi-Spy belong to EN. Access point is not highlighted because it is not needed in the current scenario. 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-floor 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.

2) Software Parameters: Linux NetworkManager service is disabled on all the EN computers so that it does not interfere with experimental wireless network configurations. Sun Grid Engine version $6.2u2\_5$ [51] is used for scheduling experiment tasks. The scheduler is configured to periodically (every 8 seconds) check the execution queues for pending

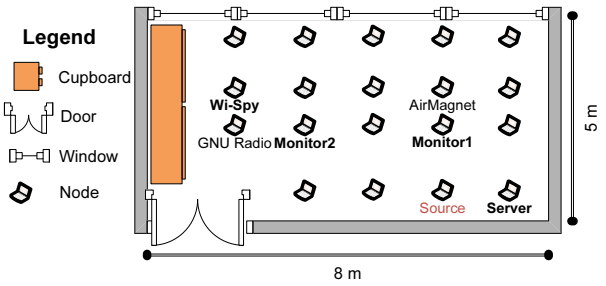

Fig. 5. WEX Toolbox indoor setup and placement of nodes

experimental tasks. Wireless tools [39] version 29 is used for interface configurations. Packet injector [52] is configured for traffic generation. In order to harness MetaGeek's Wi-Spy $2.4x$ portable USB spectrum analyzer [40], we use open-source tools from kismet known as Kismet spectrum tools [41] with custom modifications. We instrumented kismet spectrum tools in order to log spectrum information in the desired format and associate timestamps with the frequency samples.

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 $2.4GHz$ 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.

4) Wireless Parameters: MAC and PHY revisions used by the driver are 2414 and 2413 respectively. Channel type is $11g$ (operates in $2.4GHz$ 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 $18dBm$ which is the maximum value for our Atheros wireless cards.

5) Reference Time duration: The total run time for an experiment is 145 seconds. Reference time duration for which results are calculated is 100 seconds.

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. The flow of tasks through the state machine is governed by a set of rules which are as follows:

Wireless interfaces on the Source, Monitor1, and Monitor2 are configured 10 seconds after the launch of an experiment run. After waiting for another 15 seconds, tcpdump is launched on Monitor1 and Monitor2, and spectrum analyzer is launched on the Wi-Spy machine. Tcpdump and spectrum tools are scheduled for execution for total duration of 120 seconds each. After waiting for another 10 seconds, the packet injector is put into action for exactly 100 seconds. The packet injector is terminated 10 seconds before the termination of tcpdump and spectools. 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 4 minutes.

7) Metrics: For this case study we will measure RF interference, packet loss, empirical estimation of Ricean K factor and confidence intervals for goodput as observed by the probes.

F. Measurement (undertake experiment execution and data collection)

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 schedule 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.

2) Workload generation: A packet injector [52] is used to generate packets with 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 packet injector transmitting at an effective rate of less than 1 Mbps (around 600 kbps). 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.

3) Trace capture: Trace capture starts 10 seconds before the start of traffic and ends 10 seconds after traffic is stopped. This applies to both TCPDump trace and spectrum analyzer's RF trace. TCPDump utilizes the libpcap file format to capture and store the traces. Trace from the spectrum analyzer is captured using plain text format.

G. Preprocessing

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 [43].

H. Analysis

We implemented various scripts to enable rich analysis of the captured traces. Analysis scripts are 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 everytime. This practice facilitates reproducible analysis. In this section, we explain selected metrics and the mechanism to calculate each of them.

1) Channel interference and RF activity in $2.4GHz$ 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 $2.4GHz$ band (or at least adjacent channels) and study the impact of the level of interference on performance metrics such as BER/PER, goodput, etc.

Spectools [41] is configured to log frequency fluctuations for $2.4GHz$ band. It collects information consisting of frequency range 2.400 to 2.483 at 419 points with a step size of $119kHz$. 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 $500ms$ 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.

2) Empirical estimation of 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 [42]. K is obtained using the following equation

$$K = \frac{\sqrt{1-\gamma}}{1-\sqrt{1-\gamma}} \tag{1}$$

where $\gamma = {}^{V[R^2]}/_{E[R^2]}$, with $V[.]$ denoting the variance.

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.

3) Goodput CIs (Confidence Intervals): Because the goodput is a random process, it is necessary to compute confidence intervals to signify variations. The calculations are performed using the two following stages.

Goodput: Goodput is computed using a time window of $100ms$. In our wireless scenario, as the traffic is injected at low rate (at most 1 Mbps), we are able to signify goodput fluctuations better using the said time window. Let N be the number of data packets received per $100ms$, S be the packet size in bytes. Then goodput G (in kbps) is given as

$$G = ((N \cdot S \cdot 8)/1000) \cdot 10 \tag{2}$$

Where S is constant (100 bytes for packet injection) but N is variable.

Confidence Intervals: Confidence interval is calculated for the entire duration of an experiment run and for each probe separately. We calculate the average/mean goodput $\overline{G}$, standard error of mean (σ) and then lower and upper limits of confidence interval [47] as follows:

$$U = \overline{G} + (1.96 \cdot \sigma) \tag{3}$$

$$L = \overline{G} - (1.96 \cdot \sigma) \tag{4}$$

Where $\sigma = STDDEV(\overline{G}/\sqrt{M}$, where M is the size of the set of goodput values, U is the upper limit and L is the lower limit of the confidence interval (CI). Therefore

$$CI \in [L, U]. \tag{5}$$

Equations 3 and 4 show that the probability of goodput $\overline{G}$ lying in the confidence interval $[L, U]$, as shown in Equation 5, is 0.96. The results are reported in section V-I.

4) Packet Loss: Packet loss is the number of packets that fail to reach the destination. It is calculated by subtracting number of packets received from total number of packets transmitted. As we are not keeping track of the number of packets sent, we calculate packet loss by looking at the sequence numbers of the packets. Note that sequence numbers are extracted from the payload.

I. Reporting

This section elaborates benchmarking score or the result set along with the meta data necessary to fully describe and possibly reproduce an experiment. Subsection V-I1 highlights full disclosure report (FDR). Subsequent subsection demonstrates the plots for metrics explained in section V-H.

1) Reporting Rules: Table VIII shows what could be included in the full disclosure report. The specific details about individual parameters are provided in aforementioned configurations.

TABLE VIII
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	Goodput (Confidence Intervals) K Factor [$0 \rightarrow \infty$] Channel interference (Frequency vs. Amplitude)

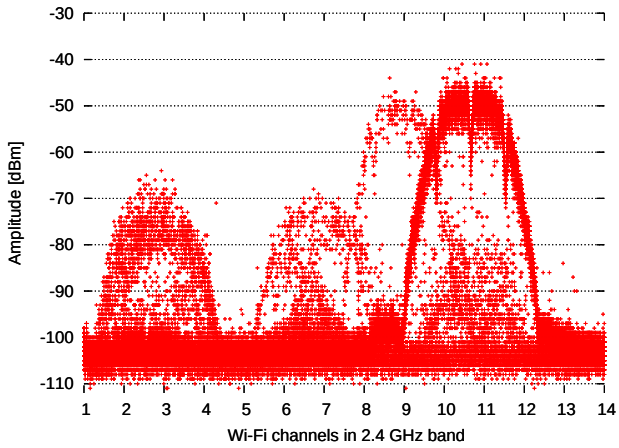

Fig. 6. Spectrum analysis: adjacent and co-channel interference

2) Channel interference and RF activity in $2.4GHz$ band: The RF landscape in 2.4 GHz wireless band during the course of one wireless experiment is shown in the Figure 6. The bandwidth of the $2.4GHz$ band is $83MHz$ i.e., $[2400MHz, 2483MHz]$. 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, we map each frequency to the corresponding WiFi channel. Therefore, the entire band 2400-2483 MHz was mapped to the 14 channels using following formula as the mapping function.

$$C = \left(\frac{X - 2400}{83}\right) \times 13 + 1 \tag{6}$$

Where C is the channel number which falls into the range $[1, 2, ..., 14]$ and X is the frequency to be mapped which falls into the range $[2400, 2401, ..., 2483]$.

RF activity corresponding to each channel is demonstrated in the Figure 6.

3) Ricean K Factor: Table IX shows K factor and corresponding RSSI values as measured on two probes named Monitor1 and Monitor2. Figure 7 is graphical representation of the results. The measurements were made against 5 runs of the same wireless experiment. Large value of K signifies less scattering and reflections caused by surrounding objects and hence less amount of multipath fading which is the case for monitor1. Small value of K means greater depth of fading which is the case for monitor2. This fact is further explained by Figures 8 and 9. The band representing received power at Monitor1 as shown in figure 8 is thinner than the band representing received power at Monitor2 as shown in figure 9. Therefore K factor is greater at Monitor1 than Monitor2.

4) Goodput CIs (Confidence Intervals): Table X shows the average goodput obtained at each probe. The fraction column, marked by $+/-$, highlights the amount to be subtracted from or added to the average goodput in order to get lower and upper bounds of the confidence interval. Figures 11 and 10 demonstrate goodput variations in the form of confidence intervals for 5 runs of the wireless experiment.

TABLE IX
K FACTOR AND RSSI

	Monitor1		Monitor2	
Run #	K Factor	RSSI	K Factor	RSSI
1	098	65	9	61
2	098	66	8	61
3	098	65	8	60
4	107	65	8	60
5	094	65	8	61

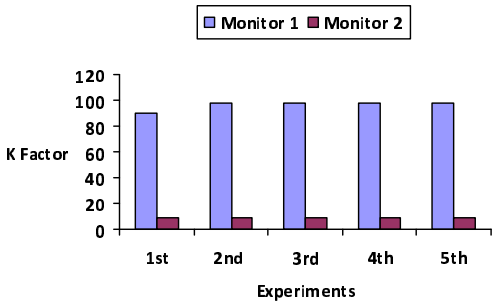

Fig. 7. K Factor estimated against 5 runs of the same experiment

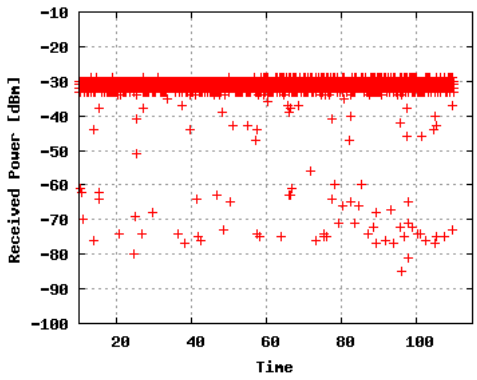

Fig. 8. Received Power at Monitor1

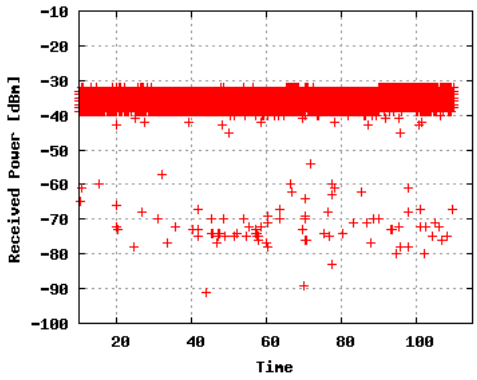

Fig. 9. Received Power at Monitor2

Figures 12 and 11 show goodput as measured on Monitor1 and Monitor2. The slight temporal variations in the goodput on the two probes are the result of queuing delays. The variations can also be caused by the discrepancies in time synchronization on the receivers.

TABLE X
AVERAGE GOODPUT AND CONFIDENCE INTERVALS

	Monitor1		Monitor2	
Run #	Avg. goodput	+/− fraction	Avg. goodput	+/− fraction
1	567.4725	3.6677	567.2647	3.6832
2	580.7072	3.3338	580.4755	3.4221
3	570.4375	3.6339	570.3889	4.5531
4	569.4225	3.6445	569.3106	3.6546
5	570.3096	3.4225	570.1658	3.4224

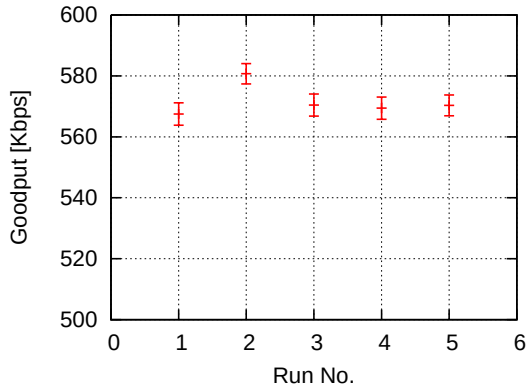

Fig. 10. Mean goodput values and corresponding confidence intervals as measured on Monitor 1

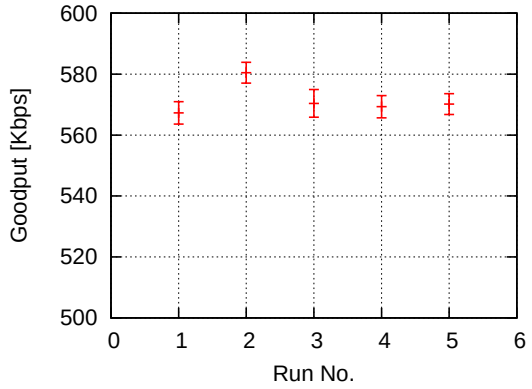

Fig. 11. Mean goodput values and corresponding confidence intervals as measured on Monitor 2

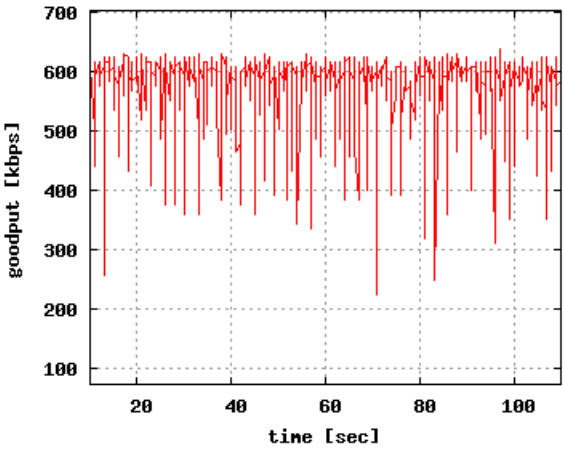

Fig. 12. Goodput as measured on Monitor 1

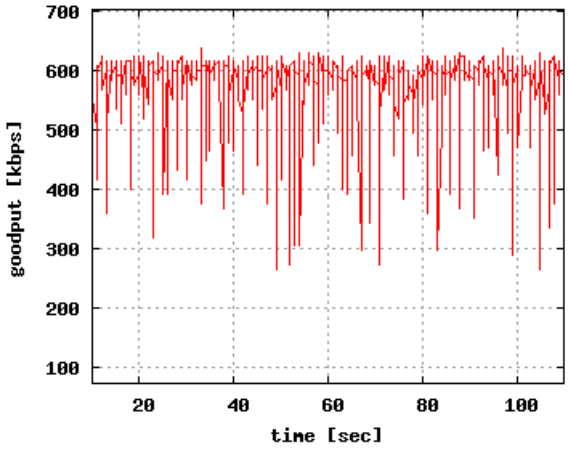

Fig. 13. Goodput as measured on Monitor 2

TABLE XI
PACKET LOSS

	Monitor2		Monitor1	
Run #	Lost packets	Percentage	Lost Packets	Percentage
1	68	0.0010	21	0.0004
2	86	0.0011	52	0.0007
3	64	0.0008	12	0.0003
4	40	0.0005	24	0.0003
5	50	0.0007	32	0.0004

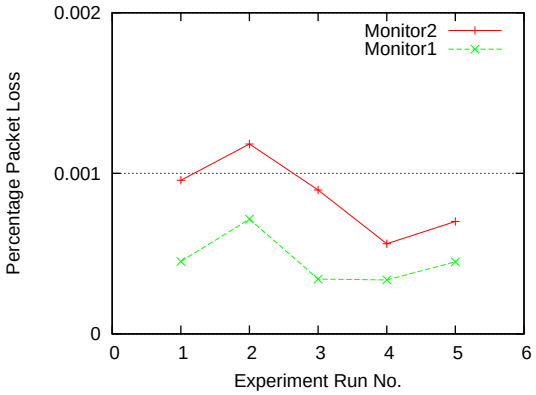

Fig. 14. Percentage packet loss on Monitor1 and Monitor2

5) Packet loss: Packet loss incurred at each probe during the course of five runs is shown in Table XI and demonstrated in Figure 14. Packet loss is less than 0.2 percent which is quite insignificant. However it is obvious that Monitor1 experienced greater packet loss despite the fact that it is closer to the source and multipath fading is less severe as shown in Figure 7. This is probably because Monitor1 is experiencing more interference which is not accounted for by Ricean K factor.

VI. CONCLUSION

Benchmarking is a very powerful tool for in-depth objective performance evaluation of wireless network, troubleshooting and management. Benchmarking results in value-add and competitiveness by facilitating the selection, adaptation and deployment of best applications and protocols of the industry.

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 standard tests and measurements. Over the last few years experimentation has evolved as the de-facto evaluation methodology for wireless networks. Experimentation solves the problem of realism but complicates comparison due to spatial and temporal vagaries of the radio environment. The work presented here intends to improve one's worldview of wireless benchmarking paradigm. We introduced key terminology, an objective classification, in-depth procedure and practical demonstration. We hope that this will encourage the research community to develop new benchmark tools and foster a more collaborative approach by sharing and publishing benchmarks, full disclosure reports and procedures.

ACKNOWLEDGEMENTS

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

REFERENCES

[1] NAS Parallel Benchmarks (NPB), http://www.nas.nasa.gov/Resources/Software/npb.html
[2] Transaction Processing Performance council (TPC), ¡http://www.tpc.org/¿
[3] Standard Performance Evaluation Corporation (SPEC), http://www.spec.org/cpu2006/
[4] RFC 2544. "Benchmarking Methodology for Network Interconnect devices," March 1999
[5] Camp, R. (1989). "Benchmarking. The Search for Industry Best Practices That Lead to Superior Performance," Productivity Press
[6] P802.11.2/D1.01. "Draft Recommended Practice for the Evaluation of 802.11 Wireless Performance," Feb 2008
[7] Robbin Mann, "Everything You need to know about benchmarking," 3rd International Benchmarking Conference, October 9-10, 2008, Budapest, Hungary
[8] A. Kashyap, S. Ganguly, S.R. Das, "Measurement-Based Approaches for Accurate Simulation of 802.11-based Wireless Networks," Proc. MSWiM, Vancouver, BC, Canada, October 2008.
[9] 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.
[10] Netperf: A network performance benchmark, ¡http://www.netperf.org¿
[11] Measurement Lab (M-Lab), ¡http://www.measurementlab.net¿
[12] PlanetLab, http://www.planet-lab.org
[13] Internet draft, "Framework for Performance Metrics Development," November 2008.
[14] Internet draft, "Reporting IP Performance Metrics to Users," July 2008
[15] Extensible Markup Language (XML), http://www.w3.org/XML/
[16] Internet Draft, "Information Model and XML Data Model for Traceroute Measurements," October 23, 2008.
[17] RFC 4741, " NETCONF Configuration Protocol," December 2006
[18] Extensible business reporting language (XBRL), http://www.xbrl.org
[19] Vern Paxon, "Strategies for sound internet measurement," Internet Measurement Conference , October 26, 2004, Italy.
[20] Network dump data displayer and Editor (NetDude), http://netdude.sourceforge.net/
[21] Romaric Guillier, Ludovic Hablot, Pascale Vicat-Blanc Primet, "Towards a User-Oriented Benchmark for Transport Protocols Comparison in very High Speed Networks," INRIA technical report, July 2007.
[22] SPEC CPU2006 Benchmarks, http://ixbtlabs.com/articles2/cpu/speccpu2k6-intro.html
[23] Wireless Mesh Test Suite, http://www.veriwave.com/products/WLAN_mesh.asp
[24] A Community Resource for Archiving Wireless Data at Dartmouth. http://crawdad.cs.dartmouth.edu
[25] Benchmarking Methodology Working Group (BMWG), http://www.ietf.org/proceedings/95apr/ops/bmwg.html
[26] 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.
[27] 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
[28] D. Raychaudhuri et al. Overview of the ORBIT radio grid testbed for evaluation of next-generation wireless network protocols. In WCNC, pages 16641669, Mar. 2005.
[29] Secure Network Testing and monitoring, http://acs.lbl.gov/~boverhof/nettest.html
[30] Test TCP (TTCP) benchmarking tool for Measuring TCP and UDP Performance, http://www.pcausa.com/Utilities/pcattcp.htm
[31] New TTCP (NTTCP), http://linux.die.net/man/1/nttcp
[32] NUTTCP-Network performance measurement tool, http://www.lcp.nrl.navy.mil/nuttcp/nuttcp.html
[33] Erinc Arikan, "Attack Profiling for DDoS Benchmarks," MS Thesis, University of Delaware, August 2006.
[34] Philip levis, Arsalan Tavakoli, Stephen Dawson-Haggerty, "Overview of Existing Routing Protocols for Low Power and Lossy Networks," draft-ietf-roll-protocols-survey-07, October 17, 2009.
[35] 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.
[36] Wolfgang Kiess, "On Real-world Experiments With Wireless Multihop Networks," PhD dissertation, 2008.
[37] Orbit Management Framework, http://omf.mytestbed.net
[38] WEX Toolbox, http://planete.inria.fr/software/Wextool/
[39] Wireless Tools for Linux, http://www.hpl.hp.com/personal/Jean_Tourrilhes/Linux/Tools.html
[40] Wi-Spy 2.4x, http://www.metageek.net/products/wi-spy-24x
[41] Kismet Spectrum Tools, http://www.kismetwireless.net/spectools/
[42] 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.
[43] Bilel Ben romdhanne, Diego Dujovne, and Thierry Turletti, "Efficient and Scalable Merging Algorithms for Wireless Traces", ROADS'09, October 14, 2009, Big Sky, Montana, USA.
[44] GNU Radio, http://gnuradio.org/redmine/wiki/gnuradio
[45] TCPDump, http://www.tcpdump.org/
[46] Wireshark, http://www.tcpdump.org/
[47] Confidence Interval, http://en.wikipedia.org/wiki/Confidence_interval
[48] Stable compat-wireless releases, http://wireless.kernel.org/en/users/Download/stable/
[49] PlanetLab MyPLC, https://svn.planet-lab.org/wiki/MyPLCUserGuide
[50] vserver capable kernel, http://svn.planet-lab.org/wiki/VserverCentos
[51] Sun Grid Engine, http://gridengine.sunsource.net/
[52] Packet Injection and Sniffing using Raw Sockets, http://security-freak.net/raw-sockets/raw-sockets.html
[53] Multi-Generator(MGEN), http://cs.itd.nrl.navy.mil/work/mgen/index.php
[54] Glenn Judd, and Peter Steenkiste, "Repeatable and Realistic Wireless Experimentation through Physical Emulation", 2003
[55] 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. 7882.
[56] Internet Measurement Data Catalog, http://www.datcat.org/
[57] A Community Resource for Archiving Wireless Data, http://crawdad.cs.dartmouth.edu/
[58] D. G. Andersen and N. Feamster, "Challenges and opportunities in Internet data mining", Technical Report CMUPDL06102, Carnegie Mellon University Parallel Data Laboratory, Jan. 2006.
[59] Network Emulation Testbed, http://www.emulab.net/
[60] Manolescu et al. "The repeatability experiment of SIGMOD 2008", SIGMOD Record, 37(1):3945, Mar. 2008.
[61] E. B. Hamida, G. Chelius and G. M. Gorce, "Impact of the physical layer modelling on the accuracy and scalability of Wireless Network Simulaltion", Simulation, September, 2009.
[62] Ettus Research LLC, http://www.ettus.com/