A Distributed Simulator Coordination Platform and its Application for Integrating an IEEE 802.11 Effective Capacity-based Admission Control Algorithm

Emmanouil Kafetzakis
Institute of Informatics &
Telecommunications
NCSR "Demokritos"
Ag. Paraskevi, Greece
mkafetz@iit.demokritos.gr

Kimon Kontovasilis
Institute of Informatics &
Telecommunications
NCSR "Demokritos"
Ag. Paraskevi, Greece
kkont@iit.demokritos.gr

Lambros Sarakis
Institute of Informatics &
Telecommunications
NCSR "Demokritos"
Ag. Paraskevi, Greece
sarakis@iit.demokritos.gr

ABSTRACT

Efficient coordination and management of radio resources in heterogeneous wireless networks is a key requirement for the realization of the Beyond 3G vision. In this context, radio resource optimization algorithms as well as infrastructures that allow for their performance evaluation are of great value. Towards these directions, the goal of the paper is twofold: first to present a testbed that enables integration of geographically distributed modules simulating different radio access technologies with independent realizations of inter-system optimization functions, and second to present the integration of an algorithm for admission control in IEEE 802.11 networks into the described platform. The algorithm, which is based on the notion of Effective Capacity and provides statistical overflow probability-related QoS guarantees, is validated using both a simulator in a stand-alone fashion and a simulator attached to the presented testbed. The integration of the algorithm on the testbed validates the testbed's capability to serve as a facility for optimization studies in simulated composite radio networks.

Categories and Subject Descriptors

I.6.7 [**Simulation and Modeling**]: Simulation Support Systems; C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design; C.4 [**Performance of Systems**]: Modeling techniques; G.3 [**Probability and Statistics**]: Queueing theory

General Terms

Algorithms, Performance, Design, Experimentation, Theory, Verification

Keywords

Distributed Virtual Testbed, Event-based middleware, Admission control, IEEE 802.11, QoS, Effective Capacity

1. INTRODUCTION

The convergence of circuit-switched and packet-switched networks and, at the same time, the increase of available data rates of wireless links has led to the rapid evolution of the so-called Beyond 3G networks integrating heterogeneous wireless access technologies. In such environments, terminals are equipped with multiple radio interfaces and are capable of attaching to different wireless access networks either alternatively or concurrently. Furthermore, services with strict Quality of Service (QoS) requirements can be allocated to different networks (either at the session establishment phase or during the session progress) so as to fulfill cost and QoS restrictions.

In this regard, efficient coordination and management schemes must exist to allocate available resources among the different networks in an optimized manner. This is the idea of inter-system optimization addressed by joint (or common, or multi-access) Radio Resource Management (RRM) schemes [9]. What is defined by joint RRM (JRRM) is a set of algorithms usually running on a centralized network entity that take decisions on the optimal access network selection or the reallocation of resources between different wireless access networks.

Many inter-system optimization schemes have been proposed in the literature for initial access selection, joint admission control and resource scheduling, inter-system handovers and load sharing for heterogeneous networking environments. Validation of such algorithms under close-to-reality conditions requires platforms that are able to integrate (possibly) disparate simulators (and even emulators and physical testbeds) of wireless technologies and, at the same time, provide for hooks that allow for smooth integration of individual implementations of optimization algorithms. Another requirement for such a platform is the ease of usability: the end-user should be able to identify available simulators and algorithms, construct simulation models and submit them for execution, monitor the simulation status and get simulation results.

Many experimental testbeds exist for wireless network research (e.g., WHYNET [14], Emulab [16], Panlab [5]) addressing simulated, emulated, physical and hybrid systems. However, although these represent significant efforts towards providing versatile experimentation platforms, they mostly

SIMUTools 2009, Rome, Italy.
Copyright 2009 ICST, ISBN 978-963-9799-9799-45-5.

Digital Object Identifier: 10.4108/ICST.SIMUTOOLS2009.5598
http://dx.doi.org/10.4108/ICST.SIMUTOOLS2009.5598

target at providing large-scale experimental infrastructures that can be used to evaluate new end-to-end approaches for next generation networking environments (like Panlab) or at providing platforms that interconnect well-established simulators (like ns-2) with a special focus on supporting cross-layer optimization studies (like WHYNET). In this regard, they constitute testbeds in which smooth integration of new simulators and JRRM algorithms can be difficult, as vital parts of the infrastructure need to be modified.

Motivated by the need for an open and flexible testbed that allows for simulations of different radio access technologies and, at the same time, provides facilities for testing optimization algorithms, the UNITE project [13] aims to deliver a simulator coordination platform (Virtual Distributed Testbed - VDT) that interconnects possibly physically isolated software simulators and independent implementations of optimization algorithms. Each one of the interconnected simulators may correspond to a different radio technology, constructing, in this way, a VDT for simulating heterogeneous wireless networks found in a Beyond 3G setting. By analyzing these requirements, it becomes apparent that a federated simulation system is a natural choice for such an experimental testbed.

Distributed systems organized in a federated fashion have received a lot of attention during the last years (see, for example, [11]) and the High Level Architecture (HLA) has been specified as a standardized way for the interconnection and coordination of federates [6]. However, the HLA implies strict rules that the federates must adhere to in order for them to communicate. For this reason, a customized mechanism for simulator coordination including data distribution and time management has been adopted for the UNITE VDT.

In this paper we describe the proposed simulator coordination platform, illustrate its application for integrating admission control algorithms and provide results from simulations governed by a particular admission control algorithm for IEEE 802.11, so as to validate the architecture and the algorithm integration (extending in this way the work presented in [15]).

Towards this end, we present the overall architecture of the platform, we discuss the functionality of the key building blocks and present the events through which communication and synchronization of the different simulation clusters are achieved. The main strengths of this architecture include a) support for distributed simulations of composite networks where different radio technologies are simulated in disparate simulators, b) integration of virtual and physical clusters, and c) integration of inter-system (and cross-layer) optimization algorithms. The design of the VDT is modular in the sense that new wireless network simulators and realizations of joint admission control (and handover) algorithms can be attached to the testbed (expanding in this way the overall testbed's capabilities) without major modifications.

To illustrate the flexibility and potential of the UNITE VDT framework, the paper discusses a recently developed algorithm for admission control, designed for providing tail-related QoS guarantees in IEEE 802.11 WLANs [7]. The theory underlying the admission control captures the service rate variability in 802.11 stations through the notion of Effective Capacity. The resulting algorithm can be used for controling the admission of new stations in the WLAN, or new traffic flows over already active stations. In the second case, the algorithm could be used as a trigger for a vertical handover to another network (of the same technology, or even a different one, in case the station has multimodal capability). The fact that the admission control scheme could be integrated in the VDT with relatively minor effort, by making use of VDT constructs devised independently from the algorithm's conception, provides reasonable evidence for the potential of the simulator coordination platform.

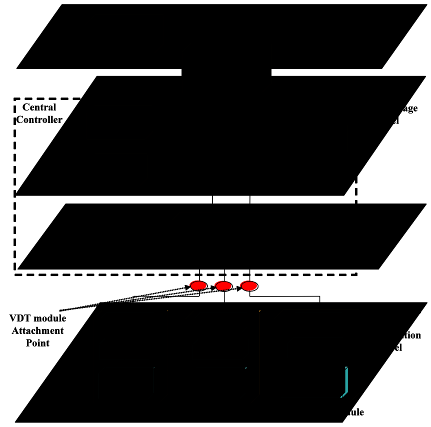

Figure 1: The architecture of the simulator coordination testbed.

The rest of the paper is organized as follows: Section 2 describes the architecture of the simulator coordination platform, while Section 3 presents the main functional blocks (as well as the events that are used for the communication between remote entities) that are required to support the testbed's goals. Section 4 presents the admission control algorithm for IEEE 802.11 networks and provides model validation and performance evaluation results using the ns-2 simulator in a stand-alone fashion. The integration of the algorithm into the simulator coordination platform, in which simulation of the IEEE 802.11 technology is addressed as a federate service managed remotely, and its evaluation in this setting are presented in Section 5. Finally, Section 6 concludes the paper.

2. ARCHITECTURE OF THE SIMULATOR COORDINATION PLATFORM

The aim of the described platform is to integrate independent physically/geographically distributed simulation clusters and realizations of optimization algorithms so as to allow for inter-system optimization studies in heterogeneous networking environments. Towards this end, two main functional requirements need to be addressed. The first relates to the derivation of a simulation facility for composite wireless networks by interconnecting existing simulators of diverse Radio Access Technologies (RATs) running on different platforms. The second has to do with the incorporation of facilities for JRRM functions. These requirements can be realized by the functional architecture depicted in Figure 1. This architecture relies on the identification of four conceptual levels for the simulator coordination testbed and the definition of discrete building blocks residing at these levels. At the topmost level, the end-user interface provides access to the Testbed Controller, which is responsible for

Digital Object Identifier: 10.4108/ICST.SIMUTOOLS2009.5598
http://dx.doi.org/10.4108/ICST.SIMUTOOLS2009.5598

connecting the VDT platform with the external world. Using this interface and the facilities provided by the Testbed Controller (e.g., identification of available simulators and optimization algorithms, exposure of simulator parameters and list of supported events, etc.), end-users compile simulation plans and submit them for execution (plans are created through a graphical tool called VDT Editor). Additional functionalities of this component include management of user authentications and permissions, retrieval of past simulation plans from the repository, validation of simulation plans, scheduling of the simulation instance execution, monitoring of the simulation state and storage of simulation plans to the repository for future use (the repository is implemented as part of the UNITE Data Base–UDB).

When a simulation is eligible for execution the Testbed Controller hands over control to the Central Controller. The latter is comprised of three modules: a) the UDB/Repository module responsible for storing simulation results and simulation plans, b) the Statistics module that performs statistical processing on the data of the UDB, and c) the Scenario Manager that undertakes functions like terminal management, service and traffic stream management and simulator clusters time management. The Scenario Manager is the core component of the simulation platform and its functionality will be discussed in the next section.

At the lowest level of the VDT architecture, there are a number of components (federates) responsible for simulating radio technologies and implementing JRRM algorithms. These components are attached to the testbed by forming VDT modules (RAT VDT modules and UNITE RRM – URRM– VDT modules, respectively). These modules are comprised of three entities: a) the actual simulator or optimization algorithm (in general the module/cluster service), b) the Federated Gateway (FG) which manages the simulation cluster or the execution of the algorithm and translates cluster-specific messages to VDT messages and vice versa, and c) the VDT module API which provides the interface (functions and parameter definitions), through which VDT modules communicate with the components of the Central Controller.

The cluster service calls specific functions from the FG to send VDT Events. In the other direction, a specific function from the cluster service is called each time a VDT Event is received from the VDT Central Controller, so that the event can be processed. The communication framework is supported by an event-based middleware implemented using SOAP messages over HTTP. The control of the events and their routing towards the distributed modules registered for them is undertaken by the Central Controller.

The VDT module is the basic building block of the VDT architecture and it can be used for any function that needs to be implemented in a separate hardware/software environment. In general, the VDT module can be considered as an abstract implementation of a desired function serving a specific purpose. In this regard, the notion of the VDT module can perfectly apply as well to the simulation control functions of the Central Controller. Additionally, the federated service of a VDT module can also correspond to an emulated or physical networking testbed.

3. FUNCTIONS AND EVENTS FOR SIMULATION CONFIGURATION AND EXECUTION

Simulation configuration and execution control is undertaken by the Scenario Manager, which is in charge of logically integrating VDT modules corresponding to simulators of different radio technologies and realizations of inter-system optimization algorithms, thus presenting to the end-user a virtual distributed simulator of a composite network consisting of heterogeneous segments. It takes input from the Testbed Controller in the form of simulation plans edited by the end-users. Each simulation plan contains information about the simulators and the optimization algorithms that will be used and their configuration parameters, the number of terminals that will participate in the simulation, the connectivity capability of these terminals with respect to the chosen RATs, the service types, QoS requirements and traffic configuration parameters, and the preferred RAT for service initiation. The information contained in the simulation plans corresponds to configuration parameters that do not change during the simulation run-time (e.g., it is assumed that session arrival events are independent of the state of the simulation).

In the following we briefly describe the functions of the Scenario Manager and the operations of the VDT modules that are responsible for the simulation of radio technologies and the implementation of joint admission control (in general initial access selection) algorithms. The discussion of the functions is complemented by a set of events that have been used to implement the corresponding functionality. Detailed description of the functions of the Scenario Manager and the events needed for the integration of VDT modules implementing handover algorithms can be found in [15].

3.1 Terminal and Service Management

The VDT platform addresses two types of terminals: single-mode terminals which are connected to and managed by only one simulator and multi-mode terminals that have connectivity to several simulators (each one simulating the radio technology of a given interface). At the system level (composite simulator level) every mobile terminal is allocated a unique global identifier (ID). For multimodal terminals, however, there is an additional need to identify the mobile in all systems it is supposed to have access to. This introduces a second level of mobile indexation inside RAT VDT modules. The mapping between the global mobile ID and the simulator-specific ID is based on the concept that (virtual) multi-mode terminals are identified by many instances, each one of which applies to a different simulator. The global IDs are provided by the VDT Editor and are managed by the Scenario Manager in a centralized manner while the local ones are assigned by the corresponding simulators upon terminal registration in the simulated radio access system.

The VDT testbed supports several types of services, all of which comply with a common service description. Services correspond to sessions initiated between mobile nodes in different RATs and fixed nodes (i.e., nodes that are supposed to belong to a wired infrastructure outside the wireless simulators). The description of the VDT services and the configuration of the associated sessions are given in a combined structure (in the form of records provided by the VDT Editor) including service description (identified by service ID

Digital Object Identifier: 10.4108/ICST.SIMUTOOLS2009.5598
http://dx.doi.org/10.4108/ICST.SIMUTOOLS2009.5598

and corresponding parameters), service QoS requirements (QoS identifier and parameters), service binding (source and destination terminal IDs), service start (and possibly stop) time and description for the session initiation (inter-session idle time distribution and parameters) and session duration processes (distribution and parameters). Examples of services include Voice over IP, HTTP, Near Real Time Video, FTP (provided they are supported by the simulators) as well as more abstract services corresponding to CBR and Poisson traffic models. Services are characterized by QoS requirements (e.g., minimum rate, tolerable packet loss/delay, etc.) and are bound to source and destination terminals identified by unique global IDs. Session arrivals are managed by the service start time and intersession idle time, while session termination takes place in accordance to session duration (and possibly a service stop time).

When a new session (traffic stream) is setup, it is assigned a unique ID. Several sessions (applications) can run on a mobile using any of the available radio access technologies. The mapping of the sessions to source/destination mobile terminals and corresponding RAT VDT modules is handled by the Scenario Manager.

3.1.1 Supporting Events

When a new mobile must be added to a simulator, an `AddMobile` event is sent by the Scenario Manager. Since the insertion of a new mobile has always to do with imminent service activation, the characteristics of the service (including QoS requirements) are incorporated in the body of the message. The newly added mobile will just appear in the simulator as an idle node; it will not yet produce or receive any traffic.

Service configuration is performed by the `SendStreamToMobile` event sent by the Scenario Manager to the RAT VDT module. Traffic activation and deactivation, on the other hand, takes place through the `StartTxToMobile` and `StopTxToMobile` events (the `StartTxToMobile` event is not combined with the `SendStreamToMobile` event in order to avoid multiple configurations of the same service in cases of frequent handovers). The `StartTxToMobile` is sent by the Scenario Manager to the RAT VDT module, indicating that a specific mobile should start transmitting traffic according to the profile that had been previously configured. The `StopTxToMobile` event indicates that a specific mobile should stop transmitting traffic. An additional `RemoveMobile` event has been specified to indicate that a specific mobile should be completely removed from a simulator in order to free resources. All of the previously described events are complemented by appropriate replies.

3.2 Initial Configuration and Time Management

Each VDT module has specific configuration parameters that must be initialized before the simulation starts. For RAT VDT modules, these configuration parameters are separated in two types. The first includes parameters that are specific to the simulated RAT (e.g., physical and MAC layer parameters) and the second includes parameters (common for all simulators) for adding background traffic to the simulator (in this case the number of terminals and their service ID are provided). For URRM VDT modules, on the other hand, the parameters are algorithm-dependent.

Time management in the VDT platform is achieved by a simple yet efficient scheme that guarantees that all connected simulators are synchronized at specified points in time (i.e., they have simulated up to a given time). With the aim of facilitating the validation of inter-system optimization algorithms, these points in time have been selected to correspond to instances when inter-system optimization functions are executed. This means that for the cases of initial network selection and inter-system handover operations the simulators need to be synchronized both when new sessions are setup and when handover algorithms are executed. The requirement to address synchronization at session setup points results in an "irregular" set of synchronization points (in the sense that the intervals between synchronization points are not constant). Handover optimization algorithms, on the other hand, can be reasonably assumed to be executed on a regular basis (e.g., every few seconds). If, however, this assumption is not acceptable it is the responsibility of the URRM VDT module to predict the time in future it will require service by the time manager and communicate this information to the Scenario Manager. This prediction can be assisted from the fact that handover decisions are usually not based on single snapshots of the system state but rather on observations taken during some time period.

3.2.1 Supporting Events

VDT module initialization parameters are provided by the Scenario Manager through the `ProcessConfiguration` event (this is the only event sent by the Scenario Manager, the format of which is simulator-dependent). After the initial configuration of the simulators, the Scenario Manager sends a Start event to all RAT VDT modules to start the simulation.

Time synchronization in the VDT platform is achieved via the `ProcessTime` and `ProcessTimeReply` events. The former is sent by the Scenario Manager to the simulators and allows them to simulate for the specified time interval (i.e., simulation step delineated by the start time and stop time parameters). The stop time can coincide with the next time in future that the optimization algorithms will be executed. The `ProcessTimeReply` event is sent as a reply to the Scenario Manager, as soon as the specified interval has been simulated and update of measurements in the UDB has taken place. At the end of the simulation duration, a Stop event is sent to all VDT modules, indicating that the simulation is completed.

3.3 Management of Simulators' Results

Each simulator is responsible for producing simulation results for the time interval it was instructed to simulate. After the end of the simulation step and before sending the permission for time advancement, each RAT VDT module updates the UDB with new measurements. These results are of interest to both the Statistics entity of the Central Controller and the URRM VDT modules.

3.3.1 Supporting Events

Simulator results for the most recent time interval are propagated to the UDB through the `UpdateValues` event. In response to this event, the UDB module sends an `UpdateValuesReply` event, after the reception of which the RAT VDT module sends permission for time advancement through the `ProcessTimeReply` event.

Digital Object Identifier: 10.4108/ICST.SIMUTOOLS2009.5598
http://dx.doi.org/10.4108/ICST.SIMUTOOLS2009.5598

To achieve proper communication between the URRM and the UDB modules two events have been specified. The `GetSystemInformation` event is sent from the URRM to the UDB, requesting "system level" simulation results for the whole composite network (e.g., total load, number of connected terminals, simulator configuration parameters, terminal and session mapping to different RAT VDT modules, etc.), while the `GetMobileInformation` event requests detailed information (e.g., throughput, packet error rate, signal strength, etc.) for specific mobiles.

3.4 Management of Admission Control VDT modules

When an admission control VDT module is included in a simulation, the default session allocation strategy, according to which sessions are allocated to different networks on the basis of preferences defined by the end-user, is bypassed and the allocation of new sessions to simulators takes place according to the decision of the algorithm. In such cases, the Scenario Manager takes the necessary actions to call the admission control algorithm at appropriate time instances and implement the decision.

3.4.1 Supporting Events

At session initiation times an `AdmissionRequest` event is sent by the Scenario Manager to the admission control module to let it decide on the admission of a new service in the system. Apart from the terminal ID, service's description and QoS requirements, the message also includes a list of simulated networks that are candidates for accepting the new service (this list may be utilized by joint admission control algorithms). After the execution of the algorithm, an `AdmissionReply` event is sent to the Scenario Manager, indicating the acceptance or rejection of the service. In the former case, the network in which the service will start is indicated.

4. EFF. CAPACITY-BASED ADMISSION CONTROL FOR IEEE 802.11 WLANS

In this section we briefly describe a recently developed Admission Control (AC) algorithm for IEEE 802.11 DCF environments. The algorithm is based on the notion of Effective Capacity and can be used to provide tail-related QoS guarantees. The ability to enforce tail-related QoS guarantees is a considerable merit of the algorithm given that virtually all other results for estimating and/or enforcing QoS for IEEE 802.11 are limited to mean-value quantities (e.g., throughput, mean packet delay, mean queue length). We limit ourselves to only the material necessary for understanding and for applying the AC scheme. For details, proofs and further implications, the reader is referred to [7]. Section 5 will address the details of incorporating this admission control scheme within VDT.

4.1 Theoretical foundations

Effective Bandwidth theory offers a linkage between source characteristics, system resources (server capacity and buffer size) and QoS tail-constraints. It was developed by a great number of contributions from various researchers (see [8] for a survey in the field). The notion of Effective Bandwidth encapsulates the details of bandwidth time-varying bursty sources in a single function, the *Effective Bandwidth function*, that can be used to express the minimum constant server capacity required to satisfy a given overflow probability-related constraint.

The theory was originally developed for queueing systems with constant server capacity. When the server's capacity is time-varying independently from the input, the theory can be generalized, by defining an *Effective Capacity function* to capture the server's burstiness. Although this generalization has been studied for some years (see, e.g., [3, 4]) it did not attract much attention until recently when the importance of wireless systems grew considerably. This is because most such systems feature a variable service rate and the notion of Effective Capacity is ideal for modeling such settings. In the literature, application of Effective Capacity to wireless systems has focused on the modeling of rate fluctuations at the physical layer [17, 18]. Instead, here and in [7] Effective Capacity theory is employed to Medium Access Control (MAC) layer modeling.

For a quick review of the Effective Bandwidth/Capacity theory, consider a single-server queue with time-varying capacity, where the capacity fluctuations are independent from the input. Let the input traffic that produces an amount of data $V(t)$ within the time window $(0, t]$ feed the queue and denote by $C(t)$ the amount of data that can be served within the time window $(0, t]$. Assuming stationary and ergodic increments for the input and output processes and some additional mild technical conditions[1], the probability that the queue size Q exceeds a certain threshold x has at all times an asymptotic exponential upper bound of rate θ, viz.,

$$a_B(\theta) \leq a_C(-\theta) \iff \lim_{x\to\infty} \frac{-\log \Pr\{Q > x\}}{x} \geq \theta, \qquad (1)$$

where

$$a_B(s) \triangleq \frac{1}{s} \lim_{t\to\infty} \frac{1}{t} \log E[e^{sV(t)}], \quad \forall s > 0,$$

and

$$a_C(s) \triangleq \frac{1}{s} \lim_{t\to\infty} \frac{1}{t} \log E[e^{sC(t)}], \quad \forall s < 0,$$

stand for the Effective Bandwidth function and the Effective Capacity function, respectively.

Assume now that the queue has a finite size x, and that we want to provide stochastic QoS by limiting the overflow probability to a value $\leq e^{-\epsilon}$. By using the tail percentile $\Pr\{Q > x\}$ of the respective infinite queue as a proxy for the overflow probability, (1) suggests that the QoS is respected (asymptotically, for large ϵ and x, maintaining a finite ratio) if and only if

$$a_B(\theta^\star) \leq a_C(-\theta^\star), \quad \text{for} \quad \theta^\star = \epsilon/x. \qquad (2)$$

Application of the admission control in (2) is eased further by noting that by definition, the Effective Bandwidth function is additive, i.e., the Effective Bandwidth of a superposition of N independent traffic flows, $i = 1, \ldots, N$ (each possibly of different traffic profile) is simply the sum of the Effective Bandwidths $a_B^{(i)}(\theta)$ of the constituent flows, namely

$$a_B(\theta) = \sum_{i=1}^{N} a_B^{(i)}(\theta), \quad \forall\theta. \qquad (3)$$

As an example, consider the case where the AC scheme is applied for deciding if a terminal with two running services

[1] Always satisfied in the setting addressed in this paper.

Digital Object Identifier: 10.4108/ICST.SIMUTOOLS2009.5598
http://dx.doi.org/10.4108/ICST.SIMUTOOLS2009.5598

is admissible or not. Assume that the first service generates constant rate traffic stream with bit rate $\mathcal{R}_{cbr}$ and that the second service generates a Poisson traffic stream with a constant packet size D and a mean bit rate $\mathcal{R}_{pois}$. Then $a_B^{(1)}(\theta) = \mathcal{R}_{cbr}$ and $a_B^{(2)}(\theta) = \mathcal{R}_{pois}\frac{e^{\theta D}-1}{D\theta}$ and the Effective Bandwidth of the aggregate traffic is governed by (3) with $N = 2$. Note that if the two services have different QoS requirements, ϵ_1 and ϵ_2, then the test in (2) should be applied with $\theta^\star = \max_{i=1,2} \epsilon_i/x$ in order to satisfy both in a shared buffer.

As with Effective Bandwidths, the form of the Effective Capacity function depends on the details of the server process. When the latter is of an On/Off type, alternating between On periods at peak server rate $\hat{r}$ and Off periods at zero rate, the Effective Capacity function can be explicitly determined [7] as

$$a_C(\theta) = u(\theta)/\theta, \quad \forall \theta < 0, \tag{4}$$

where $u(\theta)$ is the unique negative solution of

$$\log \gamma_{\text{on}}(\hat{r}\theta - u) + \log \gamma_{\text{off}}(-u) = 0. \tag{5}$$

The functions $\gamma_{\text{on}}(\cdot)$ and $\gamma_{\text{off}}(\cdot)$ are the moment generators corresponding to the distributions of the On and Off sojourns, respectively.

If one is interested in applying the AC test (2), rather than in evaluating the Effective Capacity function itself, it is not necessary to numerically solve (5). Using the monotonicity of the related functions [7], it can be proven that the inequality in (2) is equivalent to

$$\log \gamma_{\text{on}}\big(-\hat{r}\theta^\star + \theta^\star a_B(\theta^\star)\big) + \log \gamma_{\text{off}}\big(\theta^\star a_B(\theta^\star)\big) \le 0. \tag{6}$$

This condition simplifies greatly the computational aspects of the AC scheme.

4.2 The Eff. Capacity of IEEE 802.11 mobile stations

Because of the CSMA/CA access algorithm used by the IEEE 802.11 protocol, a mobile station behaves as an On/Off server [7]. The server is On, at rate equal to the channel bit rate $\hat{r}$, when transmitting successfully the payload of a packet. In all other states of the CSMA/CA protocol (terminal backing-off, colliding with other terminals, or doing overhead operations before or after a successful transmission, e.g., RTS/CTS or ACK), the server is Off.

On the basis of these observations, the moment generator function of the On period is

$$\gamma_{\text{on}}(\omega) = E[e^{\omega\frac{P}{\hat{r}}}], \tag{7}$$

where P is the payload size of the packet being transmitted. For packets of constant payload, the On period reduces simply to random variable of constant length.

The moment generator function of the Off period reads

$$\gamma_{\text{off}}(\omega) = e^{\omega t_{\text{over}}}\Big(B_o + (1-B_o)e^{\omega t_{\text{slot}}}\gamma_{\text{bo}}(\omega)\Big). \tag{8}$$

This last equation reflects that, if the backoff counter drawn at stage 0 (immediately after a successful transmission) is zero (an event of probability B_o), then the Off period simply lasts the deterministic time t_{over}, required for the overheads before and after the successful transmission. In the complementary event, with probability $1 - B_o$, the Off period additionally includes the deterministic time slot t_{slot} required for initially decrementing the backoff counter by one, plus the time spent by the terminal in backoff mode. The moment generator for this backoff time is

$$\begin{aligned}\gamma_{\text{bo}}(\omega) =& \frac{g_o\big(\gamma_s(\omega)\big) - B_o}{\gamma_s(\omega)(1-B_o)} \\ &\times \Bigg[\sum_{l=0}^{m-1}\Big((1-p)p^l e^{l\omega t_{\text{coll}}}\prod_{j=1}^{l} g_j\big(\gamma_s(\omega)\big)\Big) \\ &+ \frac{(1-p)(pe^{\omega t_{\text{coll}}})^m \prod_{j=1}^m g_j\big(\gamma_s(\omega)\big)}{1 - pg_m\big(\gamma_s(\omega)\big)e^{\omega t_{\text{coll}}}}\Bigg].\end{aligned} \tag{9}$$

In (9), $g_i(z)$ stands for the generator function of the backoff counter at the i^{th} backoff stage. The equation reflects the fact that, beyond stage m, the backoff windows maintain the same distribution. Eq. (9) allows for general backoff window distributions. For the special uniform backoff window distribution described in the IEEE 802.11 standard,

$$g_j(z) = \sum_{l=0}^{W_j-1}\frac{1}{W_j}z^l = \frac{1}{W_j}\frac{z^{W_j}-1}{z-1},$$

where

$$W_j = 2^{\min\{j,m\}}W_o, \quad j \ge 0$$

and $W_o - 1$ denotes the upper margin of the backoff window at the 0^{th} stage. For any backoff window distributions, we always have $B_o = g_o(0)$.

The quantity p in (9) denotes the probability of a collision seen by a packet being transmitted on the channel (conditional collision probability). The value of p is obtained by solving the relations

$$1 - p = (1-\tau)^{n-1}, \tag{10}$$

and

$$\tau = \frac{1}{1 + (1-p)\big(\frac{\overline{W}_o}{1-B_o} - 1 + \sum_{i=1}^{m-1} p^i\overline{W}_i + \frac{p^m\overline{W}_m}{1-p}\big)}, \tag{11}$$

between the conditional collision probability p and the transmission probability τ [1]. In (10) and (11), n is the number of competing stations and $\overline{W_i}, i = 1\ldots m$ is the mean backoff window at the i^{th} stage. Eq. (11) assumes saturation conditions, where every mobile station has always a packet to send.

Finally, the function $\gamma_s(\omega)$ appearing in (9) denotes the moment generator function of the time needed for the reduction by one of the backoff counter viz.,

$$\begin{aligned}\gamma_s(\omega) =& P_{\text{coll}}e^{\omega t_{\text{coll}}} + P_{\text{empty}}e^{\omega t_{\text{slot}}} \\ &+ P_{\text{succ}}\frac{(1-B_o)\gamma_{\text{on}}(\omega)e^{\omega t_{\text{over}}}}{1 - B_o\gamma_{\text{on}}(\omega)e^{\omega t_{\text{over}}}}e^{\omega t_{\text{slot}}},\end{aligned}$$

where

$$\begin{aligned}&P_{\text{succ}} = (n-1)\tau(1-\tau)^{n-2}, \quad P_{\text{empty}} = (1-\tau)^{n-1}, \\ &P_{\text{coll}} = 1 - P_{\text{succ}} - P_{\text{empty}},\end{aligned} \tag{12}$$

are the probabilities with which a successful transmission, an empty slot and a collision, respectively, are observed by a station backing-off (which observes $n-1$ other independent stations) and where t_{coll} represents the deterministic time for detecting a collision. Eq. (12) again relies on the assumption of saturation conditions.

All quantities involved in the On/Off server characterization just described, either depend on the probabilities p and

Digital Object Identifier: 10.4108/ICST.SIMUTOOLS2009.5598
http://dx.doi.org/10.4108/ICST.SIMUTOOLS2009.5598

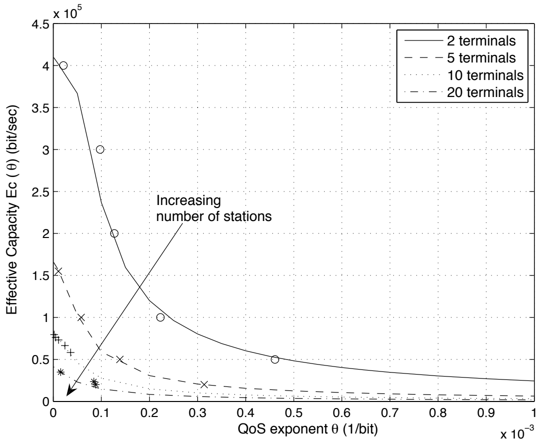

Figure 2: Curves of $a_C(-\theta)$ vs θ, for different values of the number of stations n.

τ (each of which can be expressed in terms of the other through (10)), or are directly available as parameters specified by the standard. Therefore, it is straightforward to compute values of the moment generators of the On and Off periods (through (7) and (8), respectively), towards, either the computation of Effective Capacity values (through (4) and (5)), or testing for the AC through (6).

Note that, since the Effective Capacity model of the IEEE 802.11 station employs the assumption that all other competing stations are saturated, the Effective Capacity of the station does not depend on the details of traffic through these competing stations; the only information required is the total number of stations n. Besides simplicity, in most cases the saturation assumption provides a conservative upper bound to the overflow probability (equivalently, a conservatively strict AC test), which becomes all the more accurate when the IEEE 802.11 WLAN approaches congestion.

4.3 Standalone validation

The Effective Capacity model has been validated against stand-alone ns-2 simulation results, under various forms of traffic load and number of competing terminals. For details see [7]. Here we limit ourselves in two results, in the interest of further highlighting the concepts already discussed. In both of the results the system parameter values correspond to Frequency Hopping Spread Spectrum (FHSS) PHY layer [12], a configuration which, although somewhat outdated, was intentionally chosen the same as the one in [1], for comparison purposes. Also, in all cases, the payload size (see (7)) was chosen constant and equal to 1/4 of the standard's maximum PDU (i.e., $P = 8184$ bits).

Figure 2 illustrates the accuracy of the Effective Capacity, by comparing graphs of the function (dashed, dotted and solid lines), computed with the use of the saturation-based analytic model, against simulation results (marks). The model results are consistent with known properties of the Effective Capacity function (monotonicity, $\lim_{\theta\to 0} a_C(\theta)$ equal to the mean service rate [7]) and depend on the number of terminals in the network. As intuitively expected, for a given QoS parameter value θ the Effective Capacity decreases as the network size increases.

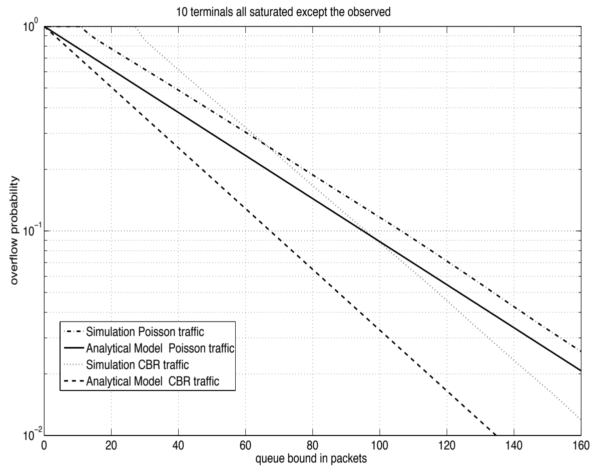

Figure 3: Modeling and simulation results of queue tail probabilites for 10 stations, one unsaturated, for two types of load.

In the simulation runs used for producing Figure 2, the values of the Effective Capacity function were indirectly measured, by feeding a "tagged" IEEE 802.11 station with traffic of known profile, sampling the probability with which the station's buffer exceeded a given threshold and exploiting the linkage (see (1)) between the Effective bandwidth of the input traffic, the Effective Capacity of the server and the probability tail just mentioned. All terminals, besides the tagged one, were operating under saturation conditions.

The remarkable match between theory and simulations validates the model and indicates its suitability for estimating queue tails or, equivalently, for taking AC decisions in IEEE 802.11 WLANs under heavy load. This fact has been reinforced by further extensive comparisons with simulation results, of which one we now present.

Figure 3 depicts curves of the queue tail probabilities versus the tail threshold (in semilog scale) for a network with 10 stations, of which 9 are saturated. The queue of the unsaturated station has been observed under two kinds of traffic load, CBR and Poisson, both featuring the same mean rate of 79.84 kbps. The slope θ of the queue tail for the model-derived curve in each loading case was determined according to the theory, namely as $\theta = \max\{s : a_B(s) \leq a_C(-s)\}$, i.e., as the unique solution of $a_B(s) = a_C(-s)$.

As shown in the figure, the simulation-derived queue tails decays exponentially and the decay rates agree well with the model results. It may also be observed that the model captures the dependence between the slope of decay and the details of the station's input traffic (expressed through the respective Effective Bandwidth function). The decay rate is faster when the unsaturated station is loaded with CBR traffic than with Poisson traffic of the same mean value, due to the smoother nature of the former. The difference in offset between model and simulation curves is attributed to the inherent shortcoming of the asymptotic theory in locating the point where the queue tail starts to decay exponentially. Despite this blemish, the Effective Capacity approach encapsulates in a single function all complicated details of the IEEE 802.11 MAC layer and provides good QoS estimates.

5. INTEGRATION AND EVALUATION OF THE AC SCHEME ON VDT

To address the integration of the AC scheme in the VDT, we now briefly summarize the basic steps of the scheme. When appropriate, we distinguish between tests for admission of a new service on an already active IEEE 802.11 station (NS-test in the sequel) and tests for admitting a newly arrived station (NT-test).

For both types of admission test, the first step consists in calculating the appropriate value $\theta^\star$ and the Effective Bandwidth of the input traffic $a_B(\theta^\star)$. As already discussed (see the paragraph following eq. (3)), $\theta^\star$ is determined by the strictest QoS requirement, among those of the services running on the station under test, and the station's buffer capacity x, namely

$$\theta^\star = \max_{i=1,\dots,s}\{-\log P_i\}/x. \tag{13}$$

In case of an NS-test, (13) includes both the existing services and the newly arriving one, while for NT-tests the equation includes the QoS specifications for all services to be initially engaged, should the terminal be admitted. Once $\theta^\star$ is available, the Effective Bandwidth $a_B(\theta^\star)$ is computed through an application of (3). Note that the VDT uses service IDs to communicate service-related information between modules. These IDs are mapped to the respective traffic and QoS descriptors internally, when needed.

The last step in applying the AC test is checking the inequality in (6). Towards this, the total number of stations n in the WLAN is retrieved (augmented by one to account for the arrival of the new station, in case of an NT-test), equations (10) and (11) are solved, the probability values to the left of (12) are computed and, finally, the test in (6) is applied by invoking (9), (8) and (7).

Note that, in case of NT-tests, normally the test in (6) should be applied to all preexisting stations as well, because the increased number of stations n affects their Effective Capacity as well (see Figure 2). In light of the mostly conservative nature of the saturation-based model, these additional tests are ommited in practice.

Table 1 compiles the input parameters for the AC scheme and the VDT entities from which these parameters are retrieved. The first row in this table refers to the values of the IEEE 802.11 MAC parameters (DIFS, SIFS, backoff window parameters, etc.) that pertain to the network into which admission is requested. These parameters are required for calculating the constant values (including time constants like t_{over}, t_{coll}, etc.) used by the equations in Subsection 4.2. All network-wide (as opposed to station-specific) parameters in the table are distinguished by a mark, and a similar distinction is made for parameters whose values don't change with time.

5.1 Communication between VDT modules

The exchange of messages required to implement session initiation with joint admission control is illustrated in Figure 4. In this figure (which, for completeness, provides also the message exchange for session termination and mobile removal) it is assumed that the admission control algorithm can operate on more than one network (although in this paper only one is used) and that a normal `ProcessTime/` `ProcessTimeReply` round has been performed just prior to the arrival of a new session.

Table 1: Input parameters for the AC scheme ('*'=for entire WLAN,'†'=static configuration).

Input parameters	From where retrieved in VDT
IEEE 802.11 MAC parameters	UDB*,†
Traffic & QoS descriptors	Defined in VDT Editor, provided by Scenario Manager
x	UDB†
n	UDB*

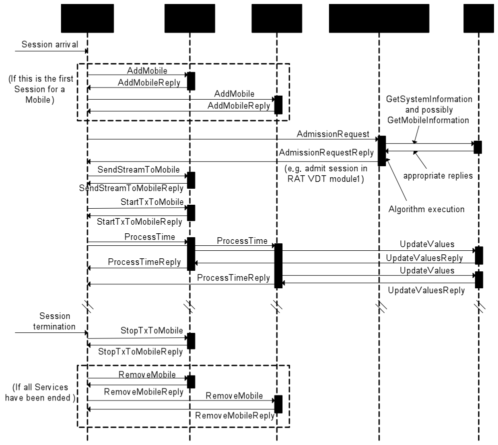

Figure 4: Messages exchange between modules for session initiation with AC.

When a session is initiated for a non-existent station, a new mobile is added to each simulator. Assuming that the new service can be supported in both networks an `AdmissionRequest` event is sent to the joint admission control URRM module. The latter retrieves information from the UDB in order to take decision. The decision is then propagated to the Scenario Manager, which takes care of configuring the traffic profile for the new station and of preparing traffic activation (which will actually take place after the next `ProcessTime` event).

After the traffic configuration is over, the simulators are instructed to continue with normal operation and the Scenario Manager takes care of new session arrivals and time advancement. For each active session, the Scenario Manager keeps track of the sessions' duration, at the end of which it sends appropriate `StopTxToMobile` commands. When all services for a station have been ended, it can be safely removed from the simulators (with the `RemoveMobile` event) in order to free computational and memory resources.

The implementation of the algorithm is addressed in the URRM entity. The algorithm runs through the `AdmissionControl()` function. The IEEE 802.11 MAC parameters and the number of stations already present in the network (see Table 1) are requested and obtained from UDB though events `GetSystemInformation` and `GetSystemInformationReply`, respectively, while the buffer size of the station subjected to the AC is requested and obtained from UDB via the `GetMobileInformation` and `GetMobileInformationReply` events (see Figure 5). The input traffic descriptors

Digital Object Identifier: 10.4108/ICST.SIMUTOOLS2009.5598
http://dx.doi.org/10.4108/ICST.SIMUTOOLS2009.5598

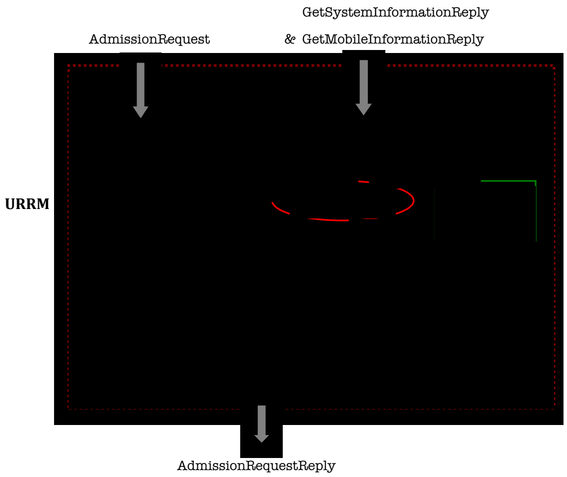

Figure 5: Flow chart representation of the AC algorithm.

and the overflow probability-related QoS requirements are obtained through the `AdmissionRequest` from the Scenario Manager. The URRM computes the required Effective Bandwidth functions on the basis of the traffic descriptors. Finally, according to the AC criterion, i.e., inequality (6), the URRM decides whether to accept the new service or not and this decision is disseminated to the Scenario Manager through the `AdmissionRequestReply` event.

5.2 Scenarios for the AC algorithm evaluation

The integration of the AC algorithm into the VDT was validated through distributed simulations involving the VDT Central Controller (including for the purposes of the tests the Scenario Manager and the UDB), an IEEE 802.11 simulator (called Pythagor) [10] that was extended for interfacing to the VDT, a URRM module implementing the AC, and the graphical application for the VDT Editor. The VDT Central Controller (running on Linux) was hosted in a server located in Lisbon, Portugal, while the other components (running on Windows XP) were hosted on different PCs located in Athens, Greece. The communication between the different modules was over the standard Internet.

In all the simulation tests considered in this subsection, the Pythagor was set to simulate a legacy IEEE 802.11 network with a rate of 1 Mbps. This configuration, although dated, was intentionally chosen to enable comparison with the standalone tests of Subsection 4.3, as well as many other tests in the literature (notable [1]). The first test addressed a setting according to which a number of sessions were initiated on mobile stations connected to an IEEE 802.11, and the AC algorithm was called at session initiation instances to decide whether the sessions could be admitted in the WLAN. The AC algorithm was taking decisions on the basis of the number of terminals that were present in the WLAN and on the QoS requirements of the new sessions. New sessions were allocated on previously inactive stations, and their duration was appropriately set to ensure that sessions never switched-off for the duration of the simulation. The second test involved the same session generation process as the first one, but assessed the performance of the system in the absence of AC.

Table 2: Simulation results provided by testbed's Pythagor VDT.

♯ Type 2 stations + 8 Type 1 stations	Overflow probability $\times 10^{-2}$	Avg. queue length (in pkts)
1	0.11	1.0314
2	0.53	1.7670
3	4.20	3.8420
4	7.34	4.7893
5	9.61	5.9315

The simulation setup assumed two types of sessions with different Traffic and QoS descriptors. Type 1 corresponded to CBR traffic with a mean rate of 100 kbps and with no QoS requirement ($P_1 = 1$). Type 2 was characterized by a Poisson traffic pattern with a mean rate of 60 kbps, and $P_2 = 10^{-2}$. A common buffer threshold was assumed for all involved stations, equal to $x = 20$ packets. Also, a constant packet (payload) size $P = 8184$ bits was used throughout.

According to the simulated scenario, eight Type 1 sessions (allocated to an equal number of stations) asked to join in the WLAN, each one after the other (with constant inter-arrival time of one second). The AC algorithm was called every time a new session was to be added to the system. By virtue of the null QoS requirement (resulting in a value of $\theta^\star = 0$), the AC criterion for Type 1 reduced simply to the stability condition (i.e, mean input rate < mean server rate). Because of the fact that the mean rate (100 kbps) of each Type 1 session was less than the hosting station's saturation throughput (mean server rate, see [1, 2, 7]) for eight competing stations, the eight sessions successively passed the admission criterion and became accepted.

After the sequence just described, five sessions of Type 2 also demanded for service successively, but now the time between session arrivals was longer (800 simulation steps, each one second long) to allow for reliably assessing metrics relevant to the QoS performance. The Type 2 sessions were subjected to two tests: In the first one, the AC algorithm was kept active and its outcome was that only up to nine sessions in total could be admitted (i.e., the preexisting Type 1 sessions plus the first Type 2 session), as inclusion of more sessions violated the Type 2 QoS requirement. The correctness of the AC decision was validated by sampling overflows beyond the threshold at the buffer of the station hosting the session admitted last. Average queue lengths were also sampled, as an additional indicator of congestion. The time averages for these single-run results appear in the first row of Table 2. As it can be observed, the value of the sampled overflow probability is below the Type 2 QoS specification, validating the AC decision.

The second test was targeted at validating the AC algorithm's decision to accommodate only one Type 2 session. The test was realized by deactivating the AC and letting all arriving Type 2 sessions be admitted. The buffers of the hosting stations were sampled as previously (for at least 800 steps) towards obtaining performance metrics. The relevant results appear in the second to last rows within Table 2. It can be seen that the AC algorithm acted conservatively to some (minor) extent, as one more session could be admitted on the basis of the results in the table. As commented in

Digital Object Identifier: 10.4108/ICST.SIMUTOOLS2009.5598
http://dx.doi.org/10.4108/ICST.SIMUTOOLS2009.5598

Subsection 4.3, some deviation from exact answers is due to the asymptotic nature and the saturation assumption underlying the scheme. Overall, however, the AC algorithm guarded against QoS violation quite effectively, as can be observed by the growth of overflow probabilities as the number of sessions increases.

6. CONCLUSIONS

The paper presented a simulator coordination testbed that can be used for optimization studies in composite wireless networks and illustrated how a recently developed algorithm for admission control in IEEE 802.11 networks can be integrated in an effective manner. The functionality of the key building blocks of this testbed was discussed and the events that were used to interconnect and synchronize geographically distributed simulation clusters and realizations of admission control algorithms were presented. The description of the admission control algorithm was complemented by its validation on a stand-alone simulator as well as on the presented simulator coordination platform. The results validated both the capability of the platform to efficiently integrate simulators and implementations of optimization functions, and the capability of the admission control algorithm to provide statistical QoS guarantees for the mobile node's buffer size.

7. ACKNOWLEDGMENTS

The work in this paper has been partly undertaken in the context of the FP6 European research project UNITE (Virtual Distributed Testbed for Optimization and Coexistence of Heterogeneous Systems), Contract No. IST-4-STREP-026906. The authors acknowledge contributions of their colleagues from the UNITE consortium.

8. REFERENCES

[1] G. Bianchi. "Performance Analysis of the IEEE 802.11 Distributed Coordination Function". *IEEE JSAC*, 18(3):535–547, 2000.

[2] G. Bianchi and I. Tinnirello. "Remarks on IEEE 802.11 DCF Performance Analysis". *IEEE Commun. Lett.*, 9(8):765–767, 2005.

[3] C. Chang and J. Thomas. "Effective Bandwidth in High-Speed Digital Networks". *IEEE JSAC*, 13(6):1091–1100, 1995.

[4] C. Chang and T. Zajic. "Effective bandwiths of departure processes from queues with time varying capacities". In *Proc. IEEE INFOCOM*, pages 1001–1009, 1995.

[5] A. Gavras, H. Brüggemann, D. Witaszek, K. Sunell, and J. Jimenez. "Pan European Laboratory for Next Generation Networks and Services". In *Proc. IEEE TRIDENTCOM'06*, Mar. 2006.

[6] IEEE 1516-2000, "Standard for Modeling and Simulation (M&S) High Level Architecture (HLA)—Framework and rules", Sept. 2000.

[7] E. Kafetzakis, K. Kontovasilis, and I. Stavrakakis. "A Novel Effective Capacity-Based Framework for Providing Statistical QoS Guarantees in IEEE 802.11 WLANs". NCSR "Demokritos" Technical Report, available online at `www.iit.demokritos.gr/nel/papers/DEMO-effcap-wlan.pdf`. Submitted for publication, 2007.

[8] F. Kelly. Notes on effective bandwidths. In F. Kelly, S. Zachary, and I. Zeidins, editors, *Stochastic Networks: Theory and Applications*, volume 4, pages 141–168. Oxford University Press, 1996.

[9] P. Magnusson, J. Lundsjö, J. Sachs, and P. Wallentin. "Radio resource management distribution in a beyond 3G multi-radio access architecture". In *Proc. IEEE GLOBECOM*, pages 3472–3477, Dec. 2004.

[10] The "Pythagor" simulation tool. Available at `http://www.icsd.aegean.gr/telecom/pythagor`.

[11] G. F. Riley, M. H. Ammar, R. M. Fujimoto, A. Park, K. Perumalla, and D. Xu. "A federated approach to distributed network simulation". *ACM Transactions on Modeling and Computer Simulation*, 14(2):116–148, Apr. 2004.

[12] "IEEE Standard for Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, Nov. 1997. P802.11".

[13] The UNITE IST-FP6 project. `http://www.ist-unite.org/`.

[14] M. Varshney, Z. Xu, S. Mohan, Y. Yang, D. Xu, and R. Bagrodia. "WHYNET: a framework for *in-situ* evaluation of heterogeneous mobile wireless systems". In *Proc. WINTECH'07*, pages 35–42, 2007.

[15] D. Vassis, L. Sarakis, G. Kormentzas, and C. Verikoukis. "A Distributed Testbed for Performance Evaluation of Inter-system Optimization Schemes in Heterogeneous Wireless Networks". In *Proc. 5th International Workshop on Next Generation Networking Middleware (NGNM)*, pages 109–122, Sept. 2008.

[16] B. White, J. Lepreau, L. Stoller, R. Ricci, S. Guruprasad, M. Newbold, M. Hibler, C. Barb, and A. Joglekar. "An integrated experimental environment for distributed systems and networks". In *Proc. of the Fifth Symposium on Operating Systems Design and Implementation*, pages 255–270, Boston, MA, Dec. 2002. USENIX Association.

[17] D. Wu and R. Negi. "Effective Capacity: A Wireless Link Model for Support of Quality of Service". *IEEE Trans. Wireless Commun.*, 2(4):630–643, 2003.

[18] X. Zhang, J. Tang, H. Chen, S. Ci, and M. Guizani. "Cross-Layer-Based Modeling for Quality of Service Guaratees in Mobile Wireless Networks". *IEEE Commun. Mag.*, 44(1):100–106, 2006.

Digital Object Identifier: 10.4108/ICST.SIMUTOOLS2009.5598
http://dx.doi.org/10.4108/ICST.SIMUTOOLS2009.5598