WHYNET: A Hybrid Testbed for Large-Scale, Heterogeneous and Adaptive Wireless Networks

Zhengrong Ji, Mahesh Marina, Maneesh Varshney, Zhiguo Xu, Yi Yang, Junlan Zhou and Rajive Bagrodia
Computer Science Department
University of California, Los Angeles
http://pcl.cs.ucla.edu/projects/whynet/

Abstract

We present an overview of the WHYNET (Wireless HYbrid NETwork) testbed, currently being developed for realistic and scalable evaluation of next-generation wireless network protocols and applications. WHYNET framework enables seamless integration of physical, simulation and emulation components in a single framework, and allows the use of any combination of those components when evaluating a target wireless network scenario. In this article, we describe the rationale behind our hybrid testbed approach, and give an overview of the architectural components of the hybrid testbed and key technical challenges addressed in its design. Further, we present several case studies to demonstrate the value of the hybrid testbed for realistic and scalable evaluation of a broad range of wireless network scenarios, focusing on cross-layer interactions, heterogeneous and large-scale wireless networks.

1 Introduction

The value of testbeds for networking research in general is widely recognized, and wireless networking is identified as a key area that can greatly benefit from research testbeds [4]. The need for wireless testbeds is motivated by the increasing use of wireless devices in networked applications together with the difficulty in accurate modeling of the behavior of various aspects of wireless networks (e.g., channel, traffic, mobility). Technology trends indicate that mobile/wireless access (potentially over multiple wireless hops and diverse radio technologies) will dominate the future Internet (while also significantly increasing the number of users and connectivity), and networked embedded sensor devices will be widely deployed for spatial & temporally dense monitoring of the physical world in diverse application domains. However, a number of significant technical challenges lie ahead before the performance and reliability of wireless access matches that of wired alternatives, and truly long-lived sensor networks become a reality. Emerging radio technologies such as MIMO and UWB promise very high physical layer data rates, but translating those rates at the application layer remains a big challenge. It is widely recognized that understanding cross-layer protocol interactions, employing adaptation mechanisms spanning multiple layers and exploiting physical layer flexibility are key to addressing this challenge, thereby achieve vastly improved user-perceived performance [20]. Another issue of concern is dealing with heterogeneity and limited coordination between wireless systems arising from the use of different set of networking and radio technologies/standards, each targeted towards a specific usage scenario (application, device, mobility and environment characteristics). Multimode wireless devices and reconfigurable software-based radios offer promising solution for synergistic operation of different wireless systems [16]. Yet another area of active research is the scalable and energy-efficient operation of sensor networks, which are typically densely deployed in large numbers.

Designing testbeds to support wireless network research in addressing challenges such as those mentioned above is a rather difficult problem as it entails simultaneously satisfying the following basic but *conflicting* set of requirements[1]:

- Realistically represent various components of the target wireless network as dictated by the questions being investigated.
- Provide high degree of control over experimental conditions and configuration for reproducible evaluation across a wide range of scenarios.
- Support large-scale evaluations in terms of network size, traffic intensity and node mobility.

[1] In addition, resource sharing issue must be addressed to support multiple users [8].

- Be cost-effective after factoring in costs for testbed hardware & software, deployment & management, model development & validation, experimentation and so forth.

Existing approaches for wireless network evaluation represent distinct regions in a space defined by the above requirement set, each offering a unique set of benefits while neither of them sufficient to meet the diverse experimentation needs of next-generation wireless networks.

- *Physical experimentation* with real wireless systems and channels (typically using small to medium scale testbeds such as MIT Roofnet [7]) is invaluable for characterization of various real-world aspects of wireless networks (e.g., channel, usage patterns, traffic, mobility) and validation of research ideas in real-world settings. But the inherent difficulty of controlling wireless channel behavior limits this approach in terms of experimental control to support repeatable experiments under diverse channel conditions, an important requirement for comprehensive and fair evaluation of cross-layer techniques; this problem becomes more severe as the target wireless network gets larger and more heterogeneous. Besides, this approach can be expensive for evaluation of large-scale or mobile scenarios and those involving emerging radio technologies (e.g., flexible and high-performance MIMO radios).

- *Simulation* is an alternative and widely used approach that supports flexible and controlled experimentation of arbitrary wireless network scenarios. It is especially useful for gaining insight into the efficacy of design alternatives at early stage of research involving new networking and radio technologies (ahead of their implementation in real systems) as well as studying the impact of scaling to larger and more stressful configurations, all in a cost-effective manner. However, intrinsic to this approach is the need to balance between accuracy of models via minimal assumptions and abstractions, on one hand, and lower execution times (scalability) and modeling related costs on the other; this limits its use for realistically studying system-wide interactions among real applications, operating system, hardware and channel dynamics.

- *Emulation* is an intermediate approach between physical experimentation and simulation in that it uses a combination of real and virtual components to realize a target wireless network scenario. Among existing wireless network emulators, some emulate only the wireless channel [21, 17] whereas others also emulate the radio device [28, 15]. All emulators operate in a lab-scale setting with most of them emulating mobility using a fixed set of nodes. Compared to simulation, the emulation approach provides better realism without incurring additional modeling costs through the use of real implementations for applications and protocols running in a real operating system and hardware environment. Relative to physical experimentation, emulation can provide greater experimental control in exchange for some realism. For these two reasons, it is an attractive approach for cross-layer studies involving adaptive applications and protocols. Besides, it allows perceptual evaluations of media applications. However, the scalability of this approach is limited by the number of nodes used for emulation.

We are developing a Wireless HYbrid NETwork (WHYNET) testbed that embodies the benefits of physical experimentation, simulation and emulation. By providing an integrated hybrid testbed environment that spans multiple evaluation approaches, WHYNET testbed offers the experimenter the flexibility to choose from a wide range of experimentation modes. The WHYNET testbed supports seamless inter-working of simulated subnets with physical subnets for scalable and realistic evaluation of heterogeneous wireless networking scenarios. WHYNET infrastructure consists of diverse set of physical testbeds including 802.11-based networks (wireless LAN, mesh, MANET), sensor networks, and novel SDR and MIMO radio platforms. In addition, WHYNET features a novel high fidelity wireless network emulator that allows running real applications and protocols on top of simulated radio devices for repeatable studies of adaptive applications/protocols and perceptual evaluations; this emulator can be seamlessly integrated with simulation for increased scalability. Not only does the WHYNET testbed provide the benefits from individual or combined use of different evaluation approaches, it allows validation across them and also permits smooth transition from design to deployment.

The purpose of this article is to provide an overview of the WHYNET testbed and show its usefulness for research on next-generation wireless networks and applications. Section 2 outlines WHYNET testbed components and elaborates on the benefits of hybrid testbed approach. Section 3 presents several case studies using the WHYNET testbed aimed at demonstrating its value and versatility for studying a wide range of interesting wireless networking scenarios. We briefly review related research on wireless testbeds in Section 4 and summarize in Section 5.

2 WHYNET Testbed Overview

WHYNET testbed is designed with the goal of providing a realistic, scalable, flexible and cost-effective evaluation environment for next-generation wireless technologies and applications. In particular, the focus is on accurate prediction of the collective impact of innovative technologies at

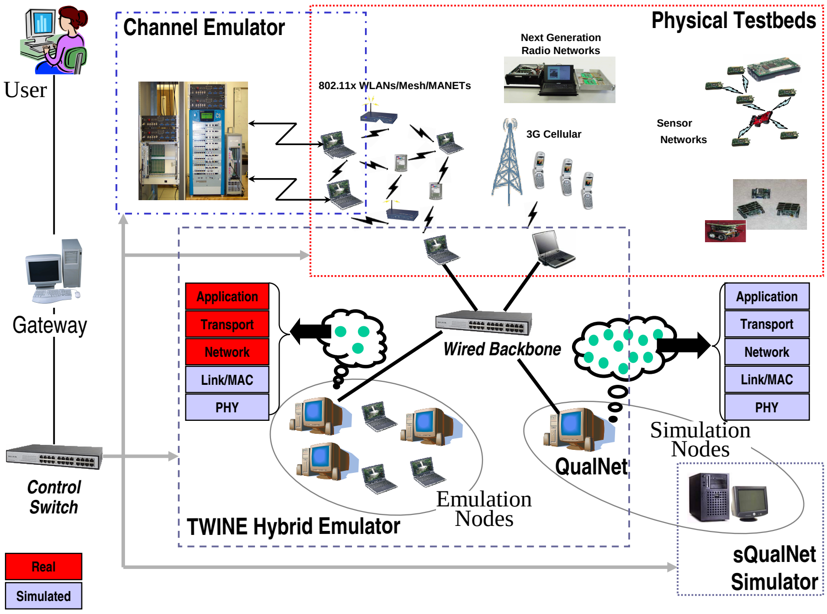


Figure 1. WHYNET Testbed Framework.

different protocol layers on application-level performance in large-scale and heterogeneous wireless networks.

WHYNET is a *hybrid* wireless network testbed environment that integrates physical, simulation and emulation components into a common framework. Beyond the traditional methods of physical experimentation and simulation, WHYNET framework supports several *hybrid* modes of experimentation (including emulation) that use physical and simulated elements in different combinations. These additional experimentation modes are enabled by a hybrid emulation framework called TWINE [28] that seamlessly integrates emulation, simulation and physical components. From a testbed user's viewpoint, WHYNET framework provides the flexibility of choosing from a wide range of experimentation modes for realizing a target wireless network scenario — the scenario can be realized entirely using either a physical testbed, simulation or emulation; alternatively, it can be partitioned into interconnected subnets with each subnet mapped to an instance of either physical, simulation or emulation components. As these different experimentation modes are well-suited for distinct purposes (as will be elaborated below), together they can better meet diverse experimentation needs of wireless network research. Further, this flexibility permits the user in making an appropriate tradeoff between realism, experimental control, scalability and cost depending on the evaluation requirements and available testbed resources. Such tradeoffs are made possible in part due to the ability of hybrid experimentation modes to naturally exploit the heterogeneity that usually exists in wireless network scenarios in terms of channel environments, radio/networking technologies, scale, fidelity requirements etc. Beyond its use as a flexible evaluation framework, the hybrid testbed facilitates smooth transition between design and deployment within a single framework. For instance, a new application/protocol can be studied at early stages of design using an abstract model in simulation. Afterwards, the prototype implementation can be tested using emulation under repeatable conditions prior to real world deployment. Similarly, WHYNET framework allows validation of radio/channel models against real measurements in small-scale configurations, which can be later used in large-scale evaluations with greater confidence.

Figure 1 illustrates the WHYNET framework consisting of physical, emulation and simulation components interconnected by a high-speed (gigabit) network switch for internal communication during an experiment. In addition, these components are externally connected to a separate control switch through which the user can control experiments (including experiment setup, execution, trace collection and online visualization of statistics) and monitor the status of testbed nodes from a console-based interface. In the following, we briefly discuss key aspects of physical, emulation and simulation components in the WHYNET framework along with their usage in wireless network evaluations.

The physical component consists of real systems with radio devices communicating over real wireless channels. WHYNET infrastructure consists of a diverse and geographically distributed set of physical testbeds spanning 802.11-based networks (wireless LANs, mesh networks, MANETs, VANETs), sensor networks, CDMA2000 cellular system, novel radio testbeds (SDR, MIMO and UWB). These testbeds are being used in a number of measurement-based characterization and real-world performance studies. In addition, any of these testbeds can be used along with emulated/simulated components in the WHYNET framework to realize heterogeneous wireless network scenarios in a cost-effective way. TWINE [28] facilitates such interaction (more on this below) via common gateway nodes interfacing physical testbeds with emulated/simulated subnets. This type of hybrid experimentation is especially useful for evaluating the impact of real-world channels in large-scale settings or to study the real-world performance with a physical testbed under diverse scenarios by subjecting it to effects such as host mobility.

The emulation component in the WHYNET framework consists of two different emulators: a hardware-based channel emulator (Propsim [17]) and a novel mobile wireless network emulator (based on TWINE [28]). Both these emulators provide complete control over the wireless channel conditions while allowing use of real application and protocol implementations. Thus, they are good candidates for repeatable and realistic cross-layer evaluations, especially those involving adaptive applications and protocols. In addition, they are useful for perceptual evaluations and providing realistic workloads for protocol evaluations. However, these two emulators differ in some key aspects, making them appropriate for different situations. Propsim provides detailed and real-time signal-level channel emulation capability that can be used by real radio devices from the physical testbeds for controlled yet highly realistic experimentation. But it permits only small-scale network configurations due to the limited number of channels.

TWINE-based emulator, on the other hand, is a more flexible and scalable alternative. With this emulator, radio device (data link and physical layers) of a node in the target wireless network scenario is emulated by a corresponding emulation node in the testbed using detailed models. A set of emulation nodes emulate the wireless channel in a distributed fashion over a wired ethernet. Commodity laptops or PC workstations can be used as emulation nodes. Each emulation node implements the emulation functionality in software as an in-kernel "emulation layer" between the IP layer and the ethernet device. TWINE emulation layer is designed with flexibility and efficiency in mind. It follows a modular design for flexible support for wide range of novel radio technologies, link layer techniques and mobility scenarios. Currently, detailed and realistic models for 802.11 MAC/PHY and propagation (path loss/fading) are supported. Efficiency is key to real-time execution of the emulation layer transparently to higher layers. Extensive evaluations show that the emulation layer is very efficient with a low CPU overhead ($< 3.5\%$) and a small memory footprint ($<$ 100KB). High fidelity emulation of channel behavior with a distributed set of emulation nodes requires them to have consistent state and timing information at all times, which in turn depends on their robustness to communication related delays and clock drifts. TWINE employs a mechanism similar to optimistic simulation to maintain consistency of temporal order among transmitted radio signals by reverting to a correct state whenever a violation is detected. For time synchronization, the master node (not explicitly shown in the figure) serves as a common time reference for the testbed nodes by issuing beacons periodically (resulting in a very effective solution that keeps measured timing errors under 5 microseconds at 4 beacons per second). Validation results for TWINE emulation layer show close match between performance of emulated and real 802.11b links for both TCP and UDP ($<$ 5% difference). For scalability, TWINE emulation layer supports multiple emulated wireless devices per emulation node. Evaluations show that it can emulate up to 4 wireless devices on a commodity PC. Even greater scaling can be achieved from integration with simulation (as discussed below).

The simulation component consists of one or more simulation nodes, each capable of simulating the complete protocol stack of a set of wireless nodes. For this purpose, a simulation node can use any existing wireless network simulator (e.g., QualNet). PC workstations or high-end multiprocessor machines can be used as simulation nodes. As mentioned already, simulation can be used in conjunction with emulation and physical testbeds for realizing large-scale (heterogeneous) wireless network scenarios where it is acceptable for simulated subnets to be modeled at a lower level of fidelity compared to emulated/physical subnets. To support such hybrid experimentation, TWINE provides an in-kernel "simulation layer" between the IP layer and the ethernet device on a simulation node for seamless interaction between the user-level simulator process on the simulation node and other emulation/physical testbed nodes. For efficiency, the simulation layer directly communicates with the simulator using file system calls. Further, the simulation layer implements a clock/synchronization functionality that uses the global time reference (from the master node) to synchronize the simulator's execution with other testbed nodes involved in the experiment (by controlling the advancement of the simulation clock and blocking the simulator if necessary). Note that the utility of simulation in the above hybrid mode of experimentation for large-scale evaluations depends to a large extent on simulator's ability to support scalable and real-time simulation with adequate fidelity. In this regard, accurate and efficient simulation models that exploit characteristics of radio and channel behavior are quite promising [11]. In addition, multi-paradigm simulation modeling [29] and parallel simulation techniques [5] can be leveraged. Evaluations show that up to 60 node wireless subnets can be simulated in real-time with high fidelity even on a commodity PC using simulation in hybrid mode. Alternatively, simulation can be used in a stand-alone mode to fully realize very large-scale wireless network scenarios by relaxing the real-time execution constraints.

As part of the WHYNET project, a number of realistic simulation models are being developed for evaluation of emerging wireless network scenarios and technologies (e.g., sensor networks, SCTP, UWB). In particular, a novel sensor network simulation framework called sQualNet [3] has already attracted a large user base in the research community. sQualNet is based on the well-known Qualnet simulator; it benefits from QualNets greater scalability, realistic and detailed propagation models, and support for easing model development. sQualNet features a rich suite of detailed and accurate sensor network specific models, including: sensing and radio channels, sensor protocols (MAC, routing), battery and power consumption models, support for multi-tiered sensor network evaluations. Besides, sQualNet provides real code simulation capability for motes. Specifically, it allows the use of unmodified TinyOS applications (written in NesC) and SOS applications (written in C), thereby enabling easy transition between simulation and real experimentation on a deployed sensor network. In addition, hybrid simulation capability is being added in sQualNet to support varying degrees of integration of physical and simulated network components, which can aid in system development and enable large-scale evaluations in a cost-effective manner.

3 Case Studies

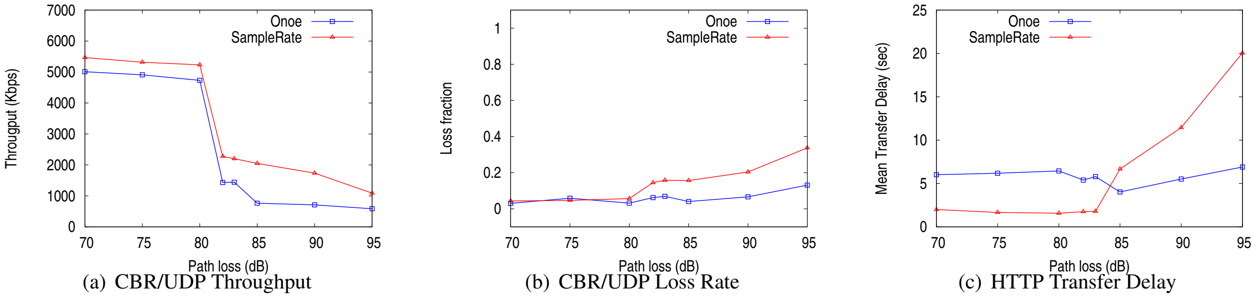
In this section, we present several case studies to demonstrate the use of WHYNET testbed components for realistic and scalable wireless network evaluations. In particular, these case studies highlight two important features of the WHYNET framework: (i) applicability for a broad range of wireless network evaluation studies (cross-layer interactions, heterogeneous and large-scale wireless scenarios) and contexts (wireless LANs, mesh, MANETs, sensor networks and cellular networks); (ii) high degree of flexibility available in selecting an appropriate mode of experimentation (i.e., physical, emulation and simulation modes individually or in a combination) depending on the experiment needs and available resources. Besides the studies below, a number of other experimental studies showcase additional uses of the WHYNET testbed infrastructure, including real-world performance (e.g., [22]) and characterization (e.g., [10]) studies via physical experimentation. We omit detailed discussions of these additional studies for brevity.

3.1 Cross-Layer Interactions

Here we discuss two studies: one using the Propsim channel emulator and the other showing the utility of TWINE emulator.

Characterizing the Interaction between 802.11 PHY Rate Adaptation and Real Applications. IEEE 802.11 is a de facto MAC/PHY standard for wireless LANs and emerging mesh networks. The 802.11 PHY provides several widely different data rates for use by higher layers — 802.11b rates range from 1 to 11Mbps, whereas 802.11a/g extend this range to 54Mbps. These rates are used by PHY rate adaptation mechanisms (usually implemented in the MAC layer) to adapt to time-varying channel conditions for improved throughput and reliability. Many mechanisms for PHY rate adaptation in 802.11 networks have been proposed recognizing its importance in determining higher layer performance. Recently, experimental evaluation of such mechanisms in real-world settings has gained much attention. These studies focus mainly on measured link layer throughput performance with backlogged UDP traffic. However, this metric is not sufficient to predict application layer performance in general as it may also depend on additional metrics such as frame loss rate; these metrics in turn are affected by the interactions among rate adaptation, MAC ARQ mechanism, frame length etc.

Our goal is to study interactions between applications and 802.11 PHY rate adaptation mechanisms. We use the Propsim channel emulator for this purpose for the following reasons. The channel emulator allows controlled experimentation over a wide range of channel conditions. In addition, it is as close to reality as possible due to the use of real applications running in a real operating system environment on real radio hardware. We consider a diverse set of common application workloads including web browsing, video streaming and file transfer. We also consider CBR/UDP traffic to relate to previous studies. We setup a simple wireless LAN scenario consisting of two Linux laptops equipped with widely used commodity 802.11b cards based on atheros chipsets; these cards are made to communicate via the channel emulator by connecting their external antenna ports to the emulator using RF cables. We configure the channel emulator to create different channel environments using a subset of TGn channel models. For brevity, we only present results with TGn channel model D (a typical office environment). Our evaluations consider Onoe [1] and SampleRate [6, 7] as two representative rate adaptation mechanisms. We choose these two specific mechanisms as



(a) CBR/UDP Throughput

(b) CBR/UDP Loss Rate

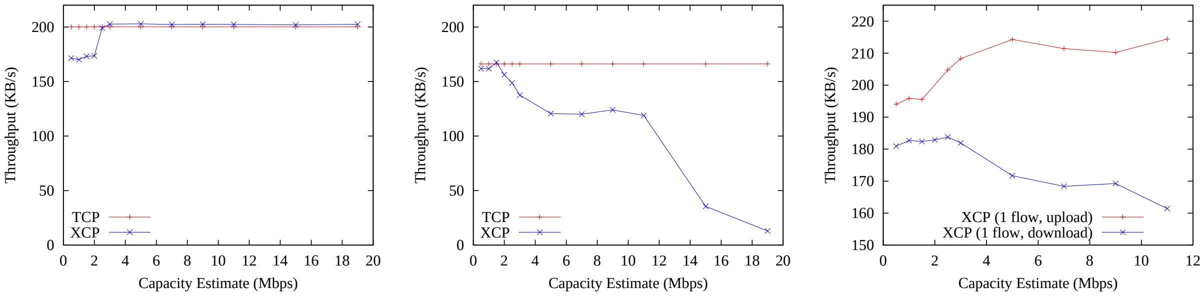
(c) HTTP Transfer Delay

Figure 2. Application performance (backlogged UDP and web traffic) with two 802.11 PHY rate adaptation mechanisms (Onoe and SampleRate) using the Propsim channel emulator.

they were not only shown to be the most effective among different mechanisms that can be readily implemented with commodity hardware [6] but also are sufficiently different in their design. Onoe is the default rate adaptation mechanism in 802.11 cards based on atheros chipsets. Onoe uses a credit-based approach to shift to higher rates, whereas SampleRate uses the average frame transmission time metric to guide rate selection. Both Onoe and SampleRate select rates in an application-oblivious manner, but differ in their aggressiveness with SampleRate being more aggressive.

Figure 2(a) shows the relative performance of Onoe and SampleRate with respect to the commonly used throughput metric using CBR/UDP traffic (generated using the well-known MGEN tool). These results correspond to a traffic load of 7Mbps with 1000 byte packets. Figure 2(a) shows that SampleRate has better or similar throughput for all channel conditions (path loss values), a consequence of the conservative strategy adopted in Onoe. This observation matches with prior results reported in literature. The corresponding packet loss rate (not considered in earlier studies) results in Figure 2(b) show opposite behavior. Again, Onoes conservative use of lower rates relatively improves its ability to provide higher reliability of frame transmissions, hence fewer frame losses that go unrecovered by the MAC ARQ mechanism. The above results essentially reflect the performance behavior of link layer throughput and frame loss rate respectively. The performance of a common application such as web browsing depends on both these metrics (more generally, multiple metrics). Specifically, web (HTTP) application runs on top of TCP whose performance is dependent on the interplay between sending rate and loss rate. Figure 2(c) shows the mean transfer delay performance for web (HTTP/1.1) traffic generated using widely used SURGE tool with default parameter settings. These results clearly show that neither rate adaptation mechanism is able to provide superior performance throughout, which is rooted in the inability of both mechanisms to tune their adaptation strategy in response to the channel quality *and* application characteristics. Our experiments with other applications also lead to similar observations. Thus, application-aware PHY rate adaptation is key to best overall user-perceived performance. More importantly, the above results clearly demonstrate the utility of channel emulator for better understanding of cross-layer interactions.

Impact of Bandwidth Estimation Errors on XCP Performance. XCP [13] is a recently proposed Internet congestion control protocol that has received considerable attention in the research community. XCP adopts a cross-layer approach in that uses explicit, precise feedback from the network about the level of congestion and adapts the rate at the sender (in the transport layer) accordingly. This is in contrast to the end-to-end approach in TCP where the sender probes for available bandwidth by gradually increasing the sending rate and infers congestion implicitly via packet loss. Certain properties of XCP are well-suited for the wireless environment even though it was originally intended as an efficient alternative to TCP over high bandwidth-delay product networks (e.g., high-speed optical networks, large delay satellite links). For example, XCP enables identification of non-congestion related wireless losses through its ability to decouple rate control from error control via precise feedback. XCP, however, requires accurate available bandwidth estimation support at each node on the path between sender and receiver for accurate feedback calculation, a challenging issue over shared and lossy wireless channels. While underestimation of available bandwidth clearly leads to poor utilization of network capacity, the inflated feedback from overestimation can also create inefficiency by causing congestion and buffer overflows. Recent experimental work on XCP [26] has highlighted the negative impact of bandwidth estimation errors on XCP performance in the wired network context with shared Eth-



(a) Single Wireless Link (TWINE emulator)

(b) Single Wireless Link (ns2)

(c) Wired/Wireless Scenario (TWINE emulator)

Figure 3. Impact of bandwidth estimation errors over wireless links on XCP performance using the TWINE-based emulator.

ernet, whereas our focus is on the more complex wireless medium.

Our goal is to experimentally evaluate XCP's sensitivity to bandwidth estimation errors over wireless links. TWINE-based emulator is an ideal evaluation environment for this study because it allows reproducible evaluations with diverse channel conditions (e.g., interference, channel related losses) while being realistic (as indicated by the validation results in [28]). Compared to the channel emulator, it permits extending the evaluation to complex scenarios (e.g., larger number of wireless devices and multihop communication) as well as studying novel bandwidth estimation techniques that may require changes to the MAC protocol (usually implemented in the firmware with commodity radio devices). In this study, we focus on the popular 802.11-based wireless LAN scenario. We use a Linux implementation of XCP from Zhang and Henderson [26]; this implementation is based on standard TCP implementation and uses TCP option fields to allow exchange of information such as congestion feedback between XCP end hosts and routers. Rather than use an actual bandwidth estimation capability, we run experiments over a wide range of static values for the "estimated bandwidth" to reflect a wide range of estimation errors (covering underestimation, accurate estimation and overestimation cases).

We first consider the simplest scenario of a single wireless link with no wireless losses. For this scenario, our testbed configuration consists of two laptops (Dell Latitude D600) running XCP over an emulated 802.11 link with TWINE-based emulator. The MTU is set to 512 bytes and 802.11b is used with PHY data rate fixed at 11Mbps. For comparison, we also present ns-2 simulation results with identical settings with XCP model used in [13] (XCP model is currently available only in ns-2 simulator). Figure 3(a) and (b) show the throughput results obtained using XCP as a function of capacity estimate for a large file transfer (80MB) with TWINE emulator and ns-2 respectively. TCP results are included for reference. Note that there is a marked difference in behavior between the two set of results especially in the overestimation case. In the bandwidth overestimation case, the XCP router provides inflated feedback to the sender (both on the same node in this scenario), causing the latter to inject data at higher than optimal rate. With TWINE emulator, bandwidth overestimation does not have any effect on XCP performance in this single link scenario because the sender host is stopped from sending more data beyond what can be handled by its device, thereby preventing any packet loss due to buffer overflow. We observed similar behavior when we turned off emulation and configured the hosts to use the built-in 802.11 interface instead. Results with ns-2, however, show that XCP performance degrades with increasing amounts of bandwidth overestimation. This we found was due to the modeling of interaction between network (IP) and link layers in the wireless protocol stack in ns-2 that differs from reality. Specifically, ns-2 allows packets to be transferred to the interface buffer (implemented as a droptail queue with a default size of 50 packets) regardless of the current buffer occupancy, which in turn causes IP layer to overflow the network interface and drop packets.

In our next experiment, we added a gigabit wired link to the scenario to create a typical wireless LAN scenario, where the wireless host connects to a wired backbone network via the AP. For this wired/wireless scenario, we consider file transfer from wireless host to the wired host via the AP ("upload") as well as in the reverse direction ("download"). Figure 3(c) shows the results obtained with TWINE emulator for this scenario. Impact of bandwidth estimation in the upload case is similar to the previously discussed isolated wireless link scenario because the wireless hosts sends directly over the bottleneck wireless link (11Mbps). The download case behaves differently with overestimation hurting XCP throughput. This is because the bottleneck is now away from the sender (wired host) at the AP where incoming packets coming over the wired link can get dropped due to lack of buffers. Thus, the impact of bandwidth estimation errors on XCP performance depends on bottleneck location on the path.

The above results demonstrate the utility of TWINE emulator for realistic evaluations of cross-layer protocols by accurately capturing real-world interactions between different protocol layers. Additional studies in [28] show the use of another TWINE capability to seamlessly integrate emulation and simulation components in the WHYNET framework for studying the performance with real applications in diverse and large wireless network scenarios. In particular, evaluation of an adaptive video streaming application in a large MANET environment (via combined use of emulation and simulation) shows that traditional quantitative metrics do not correlate well with user-perceived performance, highlighting the importance of perceptual evaluations for media applications.

3.2 Heterogeneous Wireless Networks

We now present a study to demonstrate the use of hybrid testbed for evaluating heterogeneous wireless network scenarios. In particular, we explore the use of cellular networks in conjunction with 802.11-based mesh networks for Internet access. Cellular networks, despite their wider coverage and good support for mobile voice applications, have lagged until recently in providing high data rates needed for Internet applications. This is changing with the roll out of 3G wireless data services such as CDMA-1xEVDO that offer peak rates around 2Mbps (comparable to wired broadband Internet access solutions such as DSL and CATV) and future enhancements promising much higher data rates up to 46Mbps. On the other hand, the success of 802.11 technology for indoor wireless LANs has led to its use in newer scenarios, notably community wireless mesh networks for wider and low-cost Internet access. In a typical mesh network, a set of access routers form a multihop backhaul network with a subset of them with a wired Internet connection (e.g., T-1, DSL, CATV) acting as gateways to provide Internet access to mobile clients associated with one of the access routers. The use of dedicated wired T-1/T-3 lines for Internet connectivity in a mesh network is expensive, whereas other wired broadband access solutions such as DSL and CATV are limited to highly populated metropolitan areas (also due to economic reasons). As an alternative access approach, the use of wireless wide-area network (WWAN) or cellular links offers a low-cost solution for ubiquitous broadband Internet connectivity by leveraging vast amount of extant cellular infrastructure. Also, the synergistic use of both cellular and 802.11 networks can better meet the future needs for increased performance and coverage as future advances in both cases are driven by a common set of emerging technologies (e.g., dynamic spectrum access, smart antennas) with the potential to outpace wired alternatives.

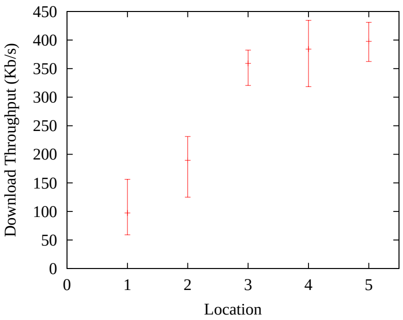


Figure 4. Spatio-temporal behavior of CDMA-1xEVDO link performance based on measurements.

We therefore consider a heterogeneous wireless mesh network architecture with some dual-mode access routers (equipped with both 802.11 and 3G cellular interfaces) serving as Internet gateways, whereas the rest of them single-mode with only a 802.11 interface. This architecture can be seen as a generalization of the two-hop-relay architecture in [24]; it differs from other architectures based on heterogeneous radio technologies such as UCAN [14] in the use of fixed relay nodes (access routers) of which not all of them need to be dual-mode.

Our goal is to evaluate Internet access performance with the above architecture as perceived by a user at a wireless host attached to the mesh network as a function of number of Internet (WWAN) gateways and their relative physical placement. Note that these two factors influence the characteristics of the multihop path in the mesh network between the wireless host and the gateway node, as well as the cellular link connecting the gateway to the Internet; these characteristics together determine end-to-end performance.

Below we describe how we map the above experimental scenario to the hybrid testbed components. To this end, we begin with a discussion on the measured performance behavior of cellular links.

Cellular links can exhibit high spatio-temporal variations in terms of bandwidth, latency and loss characteristics even with stationary nodes. A number of factors contribute to these variations including: long distance links, terrain, larger delay spread due to multipath fading, environmental mobility effects, interference among multiple users,

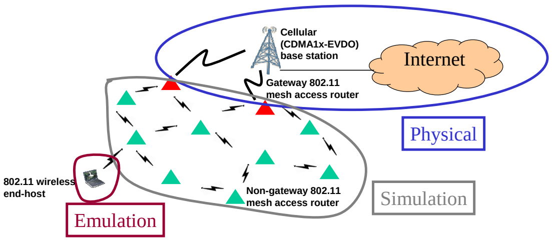


Figure 5. Realization of an integrated cellular/mesh network scenario using TWINE and physical cellular (CDMA-1xEVDO) links.

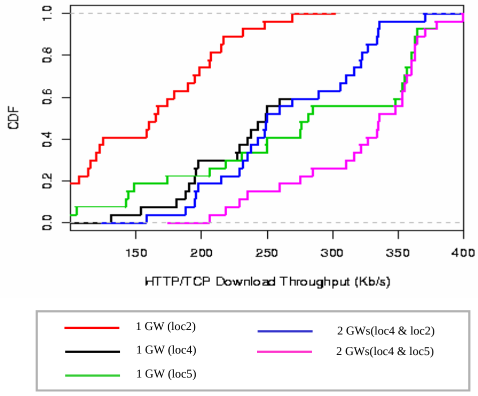


Figure 6. Impact of Internet gateways and their placement on user-perceived download performance in the integrated cellular/mesh architecture.

link adaptation and scheduling mechanisms used at the base station. As shown in Figure 4, the above assertion is supported by our measurements of download speeds using the commercial wireless broadband access service from Verizon based on CDMA-1xEVDO technology. These measurements were taken at nearby indoor/outdoor locations around Boelter Hall at UCLA. The data for each location corresponds to five back-to-back measurements for 1MB image download from the web. Since we obtained this data during early hours of the morning, the effect of interference may not be significant. We can make several observations based on this data. First, download speeds at different locations can be significantly different (up to factor of four). Second, short-term variations are considerable at locations with poor link performance. Third, the measured average throughput is at most around 400Kbps, near the lower end of the range advertised by the service provider. We also observed similar behavior as above from the measurements using the CDMA2000-1X base station deployed at UCSD as part of WHYNET testbed infrastructure.

The preceding discussion points to the difficulty in modeling of the cellular link behavior for realistic experimentation, whereas accurate radio/channel models for 3G cellular links are not readily available for use in wireless network evaluations. Since we would like to realistically capture real-world behavior of cellular link characteristics in our evaluation without incurring the high modeling costs, we use physical mode of experimentation for the cellular link part of the scenario. Mesh networks, on the other hand, are larger in scale with typical deployments ranging from several tens to a hundred nodes. In addition, mesh network links tend to be relatively stable due to dense deployments and detailed models for such links are widely available. Given that this is a feasibility study with focus centered around effects of using cellular gateways, we find it convenient to use simulation to represent the mesh network of access routers with adequate fidelity. This choice avoids the high cost and tedious setup issues associated with physical experimentation of large scale scenarios, while providing adequate fidelity. Moreover, it allows us to more easily identify the effect of cellular link dynamics and to study alternative routing/gateway selection strategies[2]. Finally, the 802.11 mobile host attached to the mesh network in our scenario is emulated to experiment with real applications. In our study, we assume that the host is stationary and use web (HTTP) traffic as the application. However, it is quite straightforward to extend the evaluation to other application scenarios such as mobile clients running VoIP. Figure 5 illustrates the realization of above scenario using a combination of emulation, simulation and physical testbeds that are integrated using TWINE.

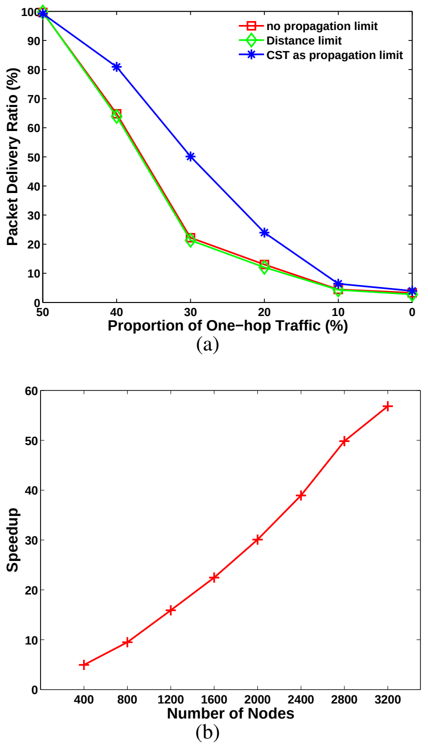
Figure 6 shows the results obtained from using the hybrid testbed as described above to realize a heterogeneous 802.11 mesh and cellular network scenario. The scenario consists of one wireless host and a 30 node 802.11b mesh network with either 1 or 2 gateway nodes at different randomly chosen locations. In Figure 6, the impact of number of gateways and their placement is shown as a CDF of average throughputs (for download of a 2MB image from the Internet) observed from a host associated with every non-gateway router in the mesh network. The locations

[2] Alternatively, access routers in the mesh network can be emulated when testing with real implementations of routing protocols prior to their use in a real deployment.

for the gateway nodes indicated in the legend correspond to the locations shown in Figure 4. In fact, the measurements for each location in Figure 6 were taken immediately after the corresponding measurements in Figure 4. The staircase pattern of the curves for single gateway case indicates the performance observed at various points in the mesh network differing in their distance (hops) to the gateway node — performance degrades with increasing hops because of greater inter-hop interference and higher likelihood of channel-related losses with longer paths. The curves for the two gateway case are obtained by taking the best throughput observed with either gateway. These results clearly show that choice of locations for the gateway nodes has significant impact on performance because of the high spatial dependence of the cellular link performance. Moreover, judicious placement of multiple gateways can yield more uniform performance in the mesh network. From an evaluation viewpoint, this study shows that hybrid testbed (through combined use of physical, simulation and emulation components) can lead to useful insights when evaluating heterogeneous wireless scenarios, and that it is a realistic, scalable and cost-effective approach for evaluating such scenarios in a lab scale setting.

3.3 Large-Scale Wireless Networks

In this subsection, we show the value of simulation for large-scale wireless network evaluations, and highlight the importance of using accurate and efficient simulation models for such evaluations. In particular, we consider two cases: (i) interference modeling in mobile ad hoc networks (MANETs); (ii) battery modeling for sensor networks.



(a)

(b)

Figure 7. Impact of accurate and efficient interference model on data delivery performance in MANETs (a) and simulation scalability (b).

Interference Modeling for Large-Scale MANET Performance Studies. The shared nature of the wireless channel makes the effect of interference among multiple concurrent transmissions an important issue to consider when studying network performance. Although a wireless transmission signal can potentially affect the successful reception of every other overlapping transmission in the network, wireless network simulators typically place a limit on signal propagation when simulating interference for smaller execution times by reducing number of events required per signal transmission. This limit is usually set arbitrarily to carrier sensing threshold (CST) and events corresponding to signals weaker than this threshold are not scheduled. However, this simplifying assumption can cause inaccuracy (especially in large networks) because the cumulative impact of weaker signals on packet errors is ignored. Using no propagation limit, on the other hand, provides utmost accuracy but can be computationally very expensive, hindering study of large-scale scenarios. In [11], the relationship between the propagation limit and ensuing inaccuracy is analytically studied by taking wireless channel propagation characteristics into consideration. Based on this analysis, a better propagation limit (termed "distance-limit") is obtained that is lower than the network size (corresponding to the "no-propagation-limit" case) while keeping inaccuracy in interference calculation below a negligible level.

Figure 7(a) compares the data delivery performance in a MANET obtained with the above three alternatives: CST-propagation-limit, no-propagation-limit and distance-limit. We use the QualNet simulator for this study. These results correspond to a MANET scenario with 400 nodes uniformly distributed in a 4km x 4km terrain; each node in this scenario runs AODV routing over a 802.11b wireless interface. For this scenario, CST propagation limit and distance limits are 679m and 2500m respectively. The traffic consists of 120 CBR sessions with sender in each session generating 512 byte packets at the rate of 8 packets/s. The traffic pattern consists of a mix of one-hop and multi-hop sessions (between randomly chosen pairs of nodes for both types). Near-identical results between the distance-limit and no-propagation-limit cases (Figure 7(a)) show the effectiveness of a carefully chosen propagation limit in accurately predicting network performance. Commonly used CST-propagation-limit, in contrast, overestimates the performance in most cases by as much as 150%. In terms of the execution time, distance-limit provides substantial speedups compared to no-propagation-limit case (ranging from 1.5 and 9 depending on the network size). However, the distance-limit increases execution time by a factor of 1.5 compared to CST-propagation-limit case. To enable scalable evaluations without compromising accuracy, we have designed several additional optimizations to improve the execution time with the distance-limit technique including: Lazy Event Scheduling with Corrective Retrospection (LSCR), greedy signal evaluation and partitioning. These optimizations together with the distance-limit technique provide impressive speedups over the no-propagation-limit case with speedups as high as 55 for a 3200 node network (Figure 7(b)).

Battery Modeling for Network Lifetime Prediction in Large Sensor Networks. In many sensor network applications, battery-driven small form-factor wireless sensor devices are densely deployed in large numbers. While these devices are severely resource constrained especially in terms of energy, battery replacement post-deployment is not an option due to the enormous scale of these networks; so they must run unattended for long periods of time. Motivated by this need, many protocols have been designed to optimize lifetime of sensor networks while meeting application needs. Note that network lifetime in turn depends on the lifetime of energy source (battery) at individual nodes. Most evaluation studies of energy conserving sensor network protocols assume ideal batteries for which battery lifetime corresponds to the time it takes to exhaust its full theoretical capacity. In reality, however, actual capacity delivered by a battery can be lower than the theoretical capacity, and depends on how the battery is discharged over time (load profile) — high load (discharge rate or current) increases the amount of unavailable capacity, some of which can be recovered with intermittent idle periods or low loads.

Several high-level models to accurately capture battery behavior while having reasonable computational complexity have been proposed in recent past. Among these models, we consider the analytical model in [18] as it is physically based and requires less configuration effort. Although the computational complexity of this model may be acceptable for portable computing applications for which it was primarily designed, it is very inefficient for network-level simulations. We have developed an optimized version of this model with similar level of accuracy by exploiting the battery load characteristics in sensor networks (i.e., small loads of short duration for communication using low-power radios in sensor nodes) [23].

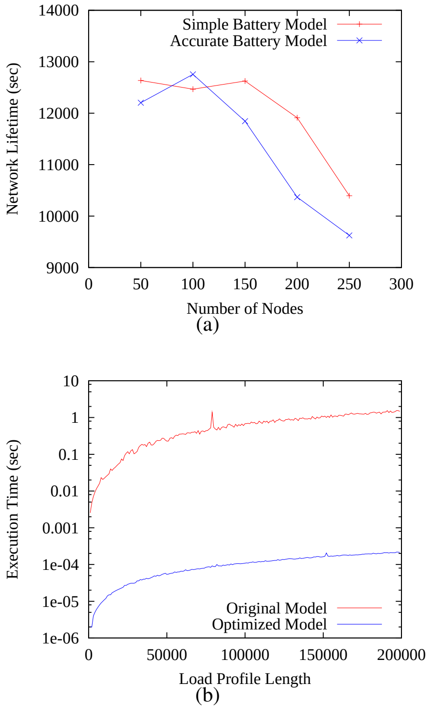


Figure 8. Impact of accurate and efficient battery model on sensor network lifetime (a) and execution time (b).

Figure 8(a) shows the impact of using the aforementioned accurate and optimized battery model on sensor network lifetime prediction as a function of network size. For comparison, we use the commonly used "simple battery model" that assumes ideal battery behavior. We assume Itsy battery with same capacity for both cases. These results were obtained using the sQualNet simulator, which features a rich suite of sensor network related models. The experimental scenario consists of nodes uniformly placed in a 400mx400m terrain. Directed diffusion is used for routing; 25% randomly chosen nodes act as sources and generate data 1 event/s to a sink, which (re)subscribes to the data by sending an interest every hour. We use S-MAC (operating in a full duty cycle mode) as the MAC protocol. Hardware and radio models are based on Rockwell's WINS nodes. Two-ray path loss with constant shadowing is used as the channel model. Network lifetime is calculated by measuring the difference between the time at which the sink loses communication to any source node and the beginning of the simulation (when batteries at all nodes are fully charged). As seen from Figure 8, the accuracy of the battery model has a substantial impact on predicted network lifetime especially for larger networks and node densities.

The observed differences in lifetime are due to the relationship between node density and radio states. Note that WINS radios in idle state draw an order of magnitude lower current from the battery compared to busy (transmit/receive) states. Since a radio is more likely to be busy either transmitting or overhearing other transmissions in a dense network, there is little or no opportunity for the battery to recover the lost capacity from applying high loads. Consequently, the battery lifetime for a given traffic load drops with increasing node density, hence the differences between the two battery models.

With regard to the execution time, we observed significant reductions (by a factor of more than 500) with our optimizations compared to the original model in [18] (Figure 8(b)) with negligible difference in accuracy (within 0.1%). This in turn has resulted in faster than real time simulation of the above sensor network scenario with several hundreds of nodes (results not included for brevity).

4 Related Work

Among the related testbed efforts, *Netbed* (a successor of Emulab) [25] is conceptually similar to WHYNET in that it also aims to integrate simulation, emulation and physical experimentation within a single framework. Unlike WHYNET, however, Netbed targets *wired* network experimentation (with main emphasis on automatic experiment setup and efficient use of testbed resources). Recent wireless extensions to Netbed focus on remote control of node software and topologies of a dense in-building deployment of fixed wireless devices, and a mobile version with small set of robots with remotely controllable mobility [2].

In recent past, several *physical wireless network testbeds* have been used for real-world protocol evaluations. These include MANET testbeds at CMU and Uppsala for ad-hoc routing protocol studies, and MIT Roofnet to evaluate mesh network performance (see [8] and references therein). These testbeds are based on commodity (802.11) hardware with limited configurability, and have narrower focus compared to general-purpose research testbeds in terms of support for diverse networking/radio technologies and access to broader research community. In contrast, WHYNET testbed infrastructure features a heterogeneous set of wireless testbeds that also include novel multi-antenna testbeds based on software-defined radios [19] and 3G (CDMA2000), sensor, UWB testbeds. Measurement traces and results from various studies will be made publicly available, and work on providing remote access to the whynet testbed infrastructure is also underway.

In an attempt to address the experiment control (including repeatability) and manageability issues associated with full-scale physical testbeds, some testbed research efforts *"emulate" wireless channels* while still using real radio devices. These efforts follow one of two approaches. In the first approach, which we term *scaled testbeds* (e.g., ORBIT [21], MiNT [9]), the idea is to attenuate radio signals to restrict the range of communication to a smaller space, thus scaling the environment for over-the-air tests. Realism, repeatability (especially in presence of external uncontrolled sources of noise or interference) and ability to support diverse experimental conditions of testbeds following this approach are not yet well-established and remains an active area of research. On the other hand, *hardware-based channel emulators* [17, 12] are highly realistic due to their detailed signal-level emulation of the wireless channel, while at the same time offer high degree of control in terms of experimenting with a wide range of channel conditions in a repeatable manner. In fact, PROPSim C8 wideband multichannel simulator [17] is part of WHYNET testbed infrastructure. On the downside, these hardware channel emulators have high cost and limited scale.

Other research efforts on wireless network emulators take a different approach by emulating wireless device and channel behaviors in software for increased flexibility and scalability at low cost. Earlier work in this category uses statistical (e.g., NIST Net) or empirically derived models (e.g., trace modulation) to subject higher layer protocols to coarse-grain wireless network dynamics in terms of delay and loss behavior. These emulators fail to realistically capture key wireless channel effects such as interference or cannot support experimentation under diverse channel conditions. Other emulators such as MobiEmu emulate only node mobility (based on trace) and are primarily meant for testing ad-hoc routing protocol implementations. More recent work additionally models radio (MAC/PHY) and channel behaviors. MobiNet/ModelNet [15] adopts a centralized emulation approach in which routing, MAC/PHY, channel and node mobility are emulated separately from the emulation hosts on a workstation cluster. Additional processing and propagation delays required to communicate between the emulation hosts and the cluster can prevent this approach from accurate and seamless emulation. MobiNet emulation modules, however, are only validated against ns-2 simulator and not against real measurements. Besides, MobiNet cannot leverage existing routing protocol implementations and requires re-implementing them specifically for the cluster nodes. EMWIN/EMPOWER [27], on the other hand, takes a distributed emulation approach using a collection of emulation nodes. Further, each of these emulation nodes can emulate multiple wireless nodes from the target scenario for scalability. However, EMWIN does not model PHY and only has an approximate (CSMA/CA) MAC model. The novel emulator in the WHYNET framework based on TWINE [28] is conceptually similar to EMWIN, but can emulate radio and channel with high fidelity using detailed models in real time. Furthermore, TWINE supports seamless integration with simulation and physical testbeds for greater scalability or realism.

Several wireless network simulators exist (e.g., ns-2, GloMoSim, QualNet, OPNET) and are commonly used for wireless network evaluations. Our simulation-oriented efforts in the WHYNET testbed context complement existing simulators in two ways: (i) accurate and efficient modeling of various unique aspects of wireless networks (e.g., interference [11], battery [23]); (ii) developing simulation support for emerging wireless network scenarios and protocols (e.g., sQualNet sensor network simulation framework [3]).

5 Summary

WHYNET is an integrated hybrid testbed environment that allows both individual and combined use of physical, simulation and emulation components for wireless network evaluations. This high degree of flexibility in choice of experimentation method not only makes the WHYNET framework suitable for realistic evaluation of wide range of wireless network scenarios, but also in a scalable and cost-effective manner. To demonstrate these features, we have presented several case studies using the WHYNET testbed components. These case studies collectively attest to the value of hybrid testbed approach in obtaining useful insights when addressing research issues of interest to wireless networking community such as cross-layer adaptation, synergistic use of heterogeneous wireless networks and large-scale wireless ad-hoc/sensor network design. In near-future, we plan to make hybrid testbed software as well as various measurement traces and simulation models available to the wider research community via the WHYNET website (http://whynet.ucla.edu). We also plan to provide limited remote access to WHYNET testbed infrastructure.

Acknowledgments

This work has been funded by the National Science Foundation under the Network Research Testbed grant “WHYNET: Scalable Testbed for Next Generation Mobile Wireless Networking Technologies.” We thank Weijun Zhu, David Browne and Prof. Mike Fitz for help with the use of channel emulator. We thank Yongguang Zhang from HRL and Prof. Medy Sanadidi for XCP Linux implementation and helpful discussions. We thank Salih Ergut, Rajesh Mishra and Ozdemir Akin from UCSD for their help with EVDO cards.

References

[1] Madwifi: Multiband Atheros Driver for WiFi. http://madwifi.sourceforge.net/.

[2] Netbed. http://www.netbed.org.

[3] sQualNet Download and Documentation Website. http://chenyen.cs.ucla.edu/projects/whynet/download.php?v=squalnet.

[4] NSF Workshop on Network Research Testbeds. http://www-net.cs.umass.edu/testbed_workshop/, Oct 2002.

[5] R. Bagrodia et al. Parsec: A parallel simulation environment for complex systems. *IEEE Computer Magazine*, 31(10), 1998.

[6] J. Bicket. Bit-rate Selection in Wireless Networks. Master’s thesis, MIT, Feb 2005.

[7] J. Bicket, D. Aguayo, S. Biswas, and R. Morris. Architecture and Evaluation of an Unplanned 802.11b Mesh Network. In *Proc. IEEE/ACM MobiCom*, 2005.

[8] P. De, A. Raniwala, S. Sharma, and T. Chiueh. Design Considerations for a Multi-hop Wireless Network Testbed. *IEEE Communications Magazine*, 43(10), 2005.

[9] P. De, A. Raniwala, S. Sharma, and T. Chiueh. MiNT: A Miniaturized Network Testbed for Mobile Wireless Research. In *Proc. IEEE Infocom*, 2005.

[10] F. Hui and P. Mohapatra. Experimental Characterization of Multi-hop Communications in Vehicular Ad Hoc Networks. In *Proc. MobiCom 2005 Workshop on Vehicular Ad Hoc Networks (VANET)*, 2005.

[11] Z. Ji, J. Zhou, M. Takai, and R. Bagrodia. Scalable Simulation of Large-Scale Wireless Networks with Bounded Inaccuracies. In *Proc. ACM MSWiM*, 2004.

[12] G. Judd and P. Steenkiste. Using Emulation to Understand and Improve Wireless Networks and Applications. In *Proc. 2nd Symposium on Networked Systems Design & Implementation (NSDI)*, 2005.

[13] D. Katabi, M. Handley, and C. Rohrs. Congestion Control for High Bandwidth-Delay Product Networks. In *Proc. ACM Sigcomm*, 2002.

[14] H. Luo et al. UCAN: A Unified Cellular and Ad-Hoc Network Architecture. In *Proc. IEEE/ACM MobiCom*, 2003.

[15] P. Mahadevan, A. Rodriguez, D. Becker, and A. Vahdat. MobiNet: A Scalable Emulation Infrastructure for Ad Hoc and Wireless Networks. In *Proc. MobiSys 2005 Workshop on Wireless Traffic Measurements and Modeling (WiTMeMo)*, 2005.

[16] P. Mahonen, J. Riihijarvi, M. Petrova, and Z. Shelby. Hop-by-Hop Toward Future Mobile Broadband IP. *IEEE Communications Magazine*, 42(3), 2004.

[17] PropSim. Propsim C8 - Wideband Multichannel Simulator. http://www.propsim.com/.

[18] D. Rakhmatov and S. Vrudhula. An Analytical High-Level Battery Model for Use in Energy Management of Portable Electronic Systems. In *Proc. IEEE/ACM International Conference on Computer Aided Design*, 2001.

[19] R. M. Rao et al. Multi-Antenna Testbeds for Research and Education in Wireless Communications. *IEEE Communications Magazine*, 42(12), 2004.

[20] T. S. Rappaport, A. Annamalai, R. M. Buehrer, and W. H. Tranter. Wireless Communications: Past Events and a Future Perspective. *IEEE Communications Magazine*, 40(5), 2002.

[21] D. Raychaudhuri et al. Overview of the ORBIT Radio Grid Testbed for Evaluation of Next-Generation Wireless Network Protocols. In *Proc. WCNC*, 2005.

[22] Y. Sun, I. Sheriff, E. Belding-Royer, and K. Almeroth. An Experimental Study of Multimedia Traffic Performance in Mesh Networks. In *Proc. MobiSys 2005 Workshop on Wireless Traffic Measurements and Modeling (WiTMeMo)*, 2005.

[23] M. Varshney and R. Bagrodia. Detailed Models for Sensor Network Simulations and their Impact on Network Performance. In *Proc. ACM MSWiM*, 2004.

[24] H. Wei and R. Gitlin. Two-Hop-Relay Architecture for Next-Generation WWAN/WLAN Integration. *IEEE Wireless Communications Magazine*, 11(2), 2004.

[25] B. White et al. An Integrated Experimental Environment for Distributed Systems and Networks. In *Proc. OSDI*, 2002.

[26] Y. Zhang and T. Henderson. An Implementation and Experimental Study of the eXplicit Control Protocol (XCP). In *Proc. IEEE Infocom*, 2005.

[27] P. Zheng and L. Ni. EMWIN: Emulating a Mobile Wireless Network using a Wired Network. In *Proc. ACM MobiCom 2002 Workshop on Wireless Mobile Multimedia (WoWMoM)*, 2002.

[28] J. Zhou, Z. Ji, and R. Bagrodia. TWINE: A Hybrid Emulation Testbed for Wireless Networks and Applications. In *Proc. IEEE Infocom*, 2006. To appear.

[29] J. Zhou, Z. Ji, M. Takai, and R. Bagrodia. Maya: Integrating hybrid network modeling to the physical world. *ACM Transactions on Modeling and Computer Simulation*, 14(2), 2004.