Lowering the Barrier to Wireless and Mobile Experimentation

Brian White Jay Lepreau Shashi Guruprasad
School of Computing, University of Utah
{bwhite,lepreau,shash}@cs.utah.edu www.flux.utah.edu www.netbed.org

ABSTRACT

The success of *ns* highlights the importance of an infrastructure that enables efficient experimentation. Similarly, Netbed's automatic configuration and control of emulated and live network environments minimizes the effort spent configuring and running experiments. Learning from the evolution of these systems, in this paper we argue that a live wireless and mobile experimental facility focusing on ease of use and accessibility will not only greatly lower the barrier to research in these areas, but that the primary technical challenges can be overcome.

The flexibility of Netbed's common abstractions for diverse node and link types has enabled its development from strictly an emulation platform to one that integrates simulation and live network experimentation. It can be further extended to incorporate wireless and mobile devices. To reduce the tedium of wireless and mobile experimentation, we propose automatically allocating and mapping a subset of a dense mesh of devices to match a specified network topology. To achieve low-overhead, coarse repeatability for mobile experiments, we outline how to leverage the predictability of passive couriers, such as PDA-equipped students and PC-equipped busses.

1. INTRODUCTION

Instruments can catalyze an entire field. In the hard sciences the right instruments are crucial, and are frequently objects of research in their own right. Closer to home, we've seen the broad impact of the *ns* simulator [1], which is used in a high fraction of network research. The benefits to the networking community of a common experimental environment have been discussed in the simulation context [1]. These include improved protocol validation, a rich infrastructure for developing and testing new protocols, a controlled experimental environment, and easier comparison of experimental results. Many of these benefits are the byproduct of a community-endorsed environment and are not unique to a simulation environment.

We have already seen impressive results with Netbed [23], an infrastructure supporting emulation, simulation, and live network experimentation. However, the nature of mobile and wireless devices both exacerbates existing problems of sound network evaluation and raises new challenges to testbed design. We believe that community-accessible wireless and mobile testbeds will have even greater impact than a wired testbed and that the technical challenges can be overcome.

This paper addresses the following limitations of most wireless and mobile experimental environments:

Lack of validation: Simulation's abstraction of low-level detail may come at the expense of accuracy. The tradeoff between accuracy and efficiency is particularly acute in a wireless network. For example, network simulators [1, 9, 24] typically incorporate idealized radio propagation models which are inadequate to model interesting indoor scenarios. Unfortunately, an experimenter is forced to make the tradeoff between accuracy and efficiency without any systematic means of validating the choice of abstraction [5].

Tedious experimental setup: Where live network experimentation is burdened by an experimenter's need to obtain remote accounts and configure nodes, mobile experimentation is further complicated by node placement and movement. A number of past [18] and proposed [19] mobile network testbeds, using automobiles, require expensive and non-scalable manual control. In fact, a lack of drivers proved a significant limitation [17]. Even static domains present configuration headaches. Kaba and Raichle interface wireless devices to a wired network with induced attenuation to curb the irreproducibility caused by multi-path effects from nearby people and objects [10]. This approach requires physically configuring the attenuators within the wired network.

Lack of realistic mobile scenarios: Simulation environments introduce randomized mobility models [3] that successfully stress network protocols, but make little attempt to capture real-world dynamics. Johansson *et al.* simulated a number of different scenarios including a conference, event coverage, and a disaster area [7]. However, these remain inaccessible to live experimentation, which is relegated to the artificial motion patterns of the above mobile testbeds. Another approach [25] skirts the issue of realistic mobility patterns by requiring that the experimenter specify a scenario describing node movement. This information is used to emulate movement on a wired network topology by affecting the "connectivity" of wired hosts. Interference effects are modeled off-line (e.g., by the Monarch project's ns extensions [9]) and are reflected in the scenario.

Lack of scale and availability: Unlike desktop machines, which abound throughout industry and academia, mobile and wireless devices are more exotic and hence available in lesser quantities and to much smaller communities. This limits the scale of existing testbeds, such as MIT's 30-node wireless "Grid" testbed [16].

Though mobile and wireless communication impose unique constraints on an experimental infrastructure, many of the above issues

First Workshop on Hot Topics in Networks (Hotnets-I), Princeton, NJ, USA, October 28–29, 2002.
This work was largely sponsored by NSF grants ANI-0082493 and ANI-0205702, Cisco Systems, and DARPA grant F30602-99-1-0503.

are not specific to these domains. We believe that many of the benefits Netbed has brought to emulation, simulation, and live network experimentation will be transferable to a mobile and wireless testbed. Section 2 describes how Netbed leverages shared abstractions despite diverse underlying hardware and implementations. It outlines how these techniques, originally designed for an emulation environment, have been successfully employed in the simulation and live network context and how they might similarly impact wireless and mobile experimentation.

The barriers to experimental evaluation in the wireless and mobile domains are much higher than in typical wired networks. This paper explores testbed designs that should dramatically lower these barriers. Section 3 discusses how interesting wireless scenarios can be realized by a dense mesh of wireless devices deployed indoors and outdoors. An automated mapping algorithm will select the subset of nodes that best match an experimenter's specified network topology and relieve the experimenter of strategically and painstakingly placing wireless devices.

Section 4 presents the notion of passive couriers: PDA- or PC-equipped mobile agents, such as students or busses, exhibiting predictable motion patterns. While their motion patterns are not completely reproducible, the dictates of their schedules mean that passive couriers capture realistic mobile scenarios with coarse-grain repeatability (e.g., within a few minutes and a few meters) at predictable times (e.g., every morning at 8 AM) without the manual tedium that is typical of mobile experimentation. Scenarios such as classroom interaction are exactly those that have motivated ad hoc networks. Finally, section 5 concludes.

2. NETBED

Netbed is a direct outgrowth of Emulab, a network experimentation platform that focused on efficient setup and control over *emulated* topologies. Key goals were to make the facility both universally available to any external researcher and extremely easy to use, without administrative, technical, or other obstacles. It pushes *automation* of testbed configuration and control to a qualitatively new level, allowing both interactive and programmatic exploration of a large space of experimental conditions.

Netbed extends the original platform by introducing simulated and distributed nodes and allowing their simultaneous use alongside emulated nodes in mixed, virtual topologies. Netbed's design generalizes resources and mechanisms into common abstractions applicable across the diverse realizations of emulation, simulation, and live network experimentation.

Netbed configures a set of distributed, emulated, or simulated nodes to realize a virtual topology specified either graphically or via an *ns* script. An experiment is defined by its configuration and any run-time dynamics (e.g., traffic generation) specified via the general-purpose *ns*/Tcl interface. Netbed's automated configuration includes managing experimenter accounts, setting emulated link characteristics, mapping the virtual nodes and links to physical resources, downloading clean disk images on emulated nodes, setting up network interfaces and IP addresses, and optionally configuring a primitive virtual machine to "jail" [11] each user of a shared machine. Once an experiment is configured, an experimenter may interactively log into emulated or distributed nodes.

A Netbed experiment may last from a few minutes to many weeks, giving researchers time to make multiple runs, change their software and parameters, or do long-term data gathering. Netbed's Web interface allows experimenters to create, pause, and terminate experiments remotely. All aspects of the experiment can be controlled via the web interface. When used in conjunction with Netbed's batch experiment system, a researcher is able to submit an *ns* file over the web and, when enough hardware resources become available to run the experiment, the user is notified that the experiment has started.

2.1 Supporting Heterogeneous Resources

Links in the user-specified, virtual topology may be emulated by traffic-shaping Dummynet nodes interposing experimental nodes, may be simulated via *ns*, or may be realized by wide-area links. *ns*'s emulation facility, *nse*, acts as transducer for packets crossing the simulation/live network boundary. An important feature of Netbed is the consistent interface it provides to control nodes and links regardless of their realization. For example, the same command uses a distributed event system to start and stop traffic generators on any type of node, be it distributed, emulated, or simulated.

The integration of heterogeneous resources is largely enabled by a database. The database serves as a level of indirection between front-end, general-purpose tools and interfaces and back-end, domain-specific implementations. It presents a consistent abstraction of heterogeneous resources to higher layers of Netbed and to experimenters. For example, the database representations of distributed and emulated nodes differ only in a type tag. Thus, in many cases, experimenters can interact with them using the same commands, tools, and naming conventions regardless of their implementation. As an example, nodes of any type can host traffic generators, despite the fact that traffic may flow over links simulated by *ns*, emulated by delay nodes, or provided by a distributed testbed.

2.2 Improving Wireless and Mobile Experimentation

Just as Netbed's notion of nodes and links has evolved to encompass distributed and simulated nodes and links, we believe the infrastructure is sufficiently flexible to incorporate wireless and mobile virtual node and link types. This will bring important practical benefits to experimentation in this domain, including: automated and efficient realization of virtual topologies, efficient use of resources through time- and space-sharing, increased fault-tolerance through resource virtualization, an ability to leverage existing tools, and easier validation across experimental techniques.

Ease of use and automation are not a mere convenience; they enable qualitatively new approaches. Our user experiments show that after learning and rehearsing the task of manually configuring a 6-node "dumbbell" network, a student with significant Linux system administration experience took 3.25 hours to accomplish what Netbed accomplished in less than 3 minutes. This factor of 70 improvement and the subsequent programmatic control over links and nodes encourage "what if" experiments that were previously too time- and labor-intensive to even consider. Experiment setup cost is even more acute in wireless and mobile domains, which require careful measuring of interference effects and "walk-through" experiments. Thus, the savings afforded by automated mapping of a virtual topology to physical devices removes a significant experimentation barrier.

Efficient use of scarce and expensive infrastructure is also important and a sophisticated testbed system can markedly improve utilization. For example, analysis of 12 months of the wired Netbed's historical logs gave quantitative estimates of the value of time-sharing (i.e., "swapping out" idle experiments) and space-sharing (i.e., isolating multiple active experiments). Although the behavior of both users and facility management would change without such features, the estimate is still revealing. Without Netbed's ability to time-share its 168 nodes, a testbed of 1064 nodes would have been required to provide equivalent service. Similarly, without space-sharing, 19.1 years would be required. These are order-of-magnitude improvements. The importance of resource efficiency is heightened for wireless and mobile devices since they are less prevalent than commodity PCs.

Netbed virtualizes node names and IP addresses such that nodes and links form equivalence classes. For example, when an experiment is "swapped in" (i.e., reconstituted on physical resources from database state), it need not execute on the same set of physical nodes. Any nodes exhibiting the same properties and interconnection characteristics are suitable candidates. While virtual nodes may be explicitly bound to specific physical hosts, the flexibility to allocate from an equivalence class adds a measure of fault tolerance. If a node or link fails, an experimenter need not wait until the node or link partition is healed, but may instead re-map the experiment to an equivalent set of machines. This approach is valuable wherever node or link failures are anticipated: large-scale clusters, wide-area networks, or unstable wireless environments.

Incorporating wireless and mobile devices under the Netbed umbrella brings a mature set of tools and features to these domains. For example, once domain-specific resource mapping is provided, experimenters will use the existing *ns* or graphical interfaces to allocate wireless and mobile nodes. This ensures that any topology generator that produces *ns* syntax may be used to configure an experiment, while *ns* virtualization tools may be used to view it. The infrastructure will extend consistent control and specification of traffic generators across all experimental methodologies. The familiar user account management, including hierarchical authorization, key distribution, and account establishment will translate directly to these new domains. Applications running on wireless nodes will be controllable via the current event system and standard commands.

The common *ns* interface makes it easier to compare experimental environments, thereby facilitating and encouraging validation. Extending this capability to wireless and mobile nodes will provide an automatic way to compare simulated radio propagation models with real devices. The same *ns* script could then be used to instantiate either an *ns* simulation or a live experiment, leading to an approach in which the simulated model is iteratively refined according to empirical results.

We believe a new hybrid style of experimentation, incorporating resources from multiple experimental environments, can simultaneously leverage the particular strengths of each. More specifically, an experimenter can leverage the greater scalability of simulation without surrendering confidence in the accuracy of the results. A radio propagation model is sufficiently accurate *for a given application*, if the simulation it drives is indistinguishable from the interaction between live, wireless nodes. To this end, we suggest replacing a small "island" of simulated nodes with physical nodes. This subset of nodes will use live, wireless communication amongst themselves and simultaneously transmit packets within the simulated world. This allows an experimenter to apply an emulation "microscope" to achieve highly accurate introspection of an "island" of nodes. Moving or removing this "island" from the experiment will yield similar results if the analytic model of the simulator and the real instantiation of those models are equivalent. Further, if sets of nodes exhibit symmetrical interactions, the behavior of a cluster of interacting simulated nodes may be compared to a corresponding set of live wireless nodes.

Such close interaction between live traffic and simulation requires careful synchronization between simulated and real time. We intend to leverage the work of Ke *et al.* [12] to reduce timing discrepancies between the two worlds. The relative expense of wireless simulation exacerbates such concerns. A parallelizable simulation architecture, such as GloMoSim, would be more efficient in a multiprocessor environment, and is worth exploring. We have already done some work in this area, including cutting CPU usage in half by making some modest changes to *nse*'s scheduler.

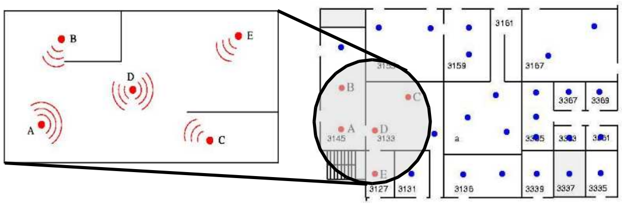

Figure 1: Wireless Virtual to Physical Mapping

3. WIRELESS TESTBED

Broadcast and interference effects are difficult or impossible to emulate on a wired testbed, although centralized simulation of all links [12] can provide some degree of realism. However, the lack of good propagation models, particularly for indoor environments, severely limits the accuracy of this approach. Trace modulation [20] finesses concerns over broadcast and interference issues as it captures their effect within a trace. However, the traces are bound to the sampled environment; if the scenario changes, new traces must be collected.

Our approach is to overcome these difficulties with a large pool of physically distributed test nodes that have real RF devices. When the number of nodes under test is significantly fewer than the total number of available nodes, it should be possible to provide a reasonable approximation of the intended virtual environment. Given the costs of the devices involved, the benefit to researchers, and the multi-institutional stake of the facility, we believe that sufficient device density will become available. As an example of the scale under consideration, we are considering 500 devices within a four-story building.

The wireless Netbed will include a dense deployment of wireless devices throughout one or more campus buildings. [1] Devices will typically be attached to control machines (e.g., low-end PCs or similar; UCLA is using PC-104 "Octopus" machines each of which controls 8 RF devices), enabling independent, reliable control by Netbed tools. Many devices will have permanent—and to the degree economical, controllable—sources of power and power monitoring hardware. This avoids the maintenance problem of battery replacement while facilitating studies of power-related issues.

For an indoor wireless mesh, 900 Mhz devices have the appropriate range and popularity. We will select devices with a programmable MAC layer, such as "motes" [6] from Crossbow/Intel/-Berkeley, giving experimenters flexibility in that important dimension. The pervasiveness of 802.11b makes it an obvious choice to populate the indoor mesh. The higher data rates and larger number of frequency channels also make 802.11a an attractive option. To allow closer packing, we will reduce the transmit power of the devices [10]. Bluetooth devices could be deployed very densely, but their long-term importance and popularity are uncertain. Sensors, including those for temperature, light, and probably motion, will be included on nodes due to their low marginal cost and importance to real-time applications.

[1] In a wild possibility for researchers or others, Aerie Networks acquired Metricom's Ricochet 900 Mhz system, but has "abandoned in place" huge numbers of operational transceivers and access points in many U.S. cities. [4].

The success of an automated, wireless testbed hinges on two primary challenges: providing a high fidelity mapping of a virtual topology to wireless nodes and ensuring that interference does not pollute experimental results. These considerations are examined in the following subsections.

3.1 Mapping Wireless Resources

Our experience mapping complex virtual requirements to physical resources in Netbed showed that approaches based on combinatorial optimization can be practical with sufficient attention to abstraction, data representation and, sometimes, local search. Currently, a randomized heuristic algorithm, simulated annealing [8], underlies allocation of Netbed's local resources (nodes, links, and switches), efficiently targeting the NP-hard problem of mapping virtual resources to physical ones. We improved its computational behavior by grouping physical resources into equivalence classes based on their physical attributes; it finds solutions in a few seconds.

Whereas the specification of a wired topology is fairly straightforward, a faithful mapping of an experimenter's intent to wireless resources is highly dependent on the level of detail provided by the configuration interface. Such an interface must avoid a circular dependency: if Netbed were to rely on an existing simulation model to map a virtual topology to physical resources, the system's reliance on models incorporates the potential inaccuracies that live experimentation seeks to avoid! It may be possible to use offline analysis that would be intolerably slow in a simulation. Other possibilities, outlined below, avoid analytic models in favor of more intuitive and efficient interfaces.

Configurable wireless experimentation will allow manual or automatic selection of a subset of the wireless nodes, chosen to match certain characteristics, as represented in Figure 1. We plan three different user interfaces. First, and simplest to develop, we will provide an annotated 3-D map of the building, with nodes colored green, yellow, or red, indicating whether they are available, assigned to an idle experiment ("swappable"), or busy. Experimenters simply select the nodes they prefer, inferring link characteristics from the map and its annotations.

Next, we will develop a more abstract interface that allows a user to specify a scenario graphically, based on spatial arrangement, as in Figure 1. The mapping code will select the set of physical nodes that best match. Our algorithmic approach is uncertain at this stage, but seems to have an intuitive mapping to subgraph isomorphism. Graph edges correspond to some abstracted characterization of the configuration that affects attenuation, such as distance, number of obstructing walls, their orientation—but not to any concrete metric such as attenuation itself. However, this still implies estimating a propagation model, with all its assumptions. A better approach may be simply to match the spatial layout as closely as possible.

Experimental results from these first two approaches are clearly dependent on the RF idiosyncrasies of the particular building and environment. However, this downside is outweighed by the building's becoming a "reference platform" across multiple experiments and between experimenters—which has heretofore not been available.

Lastly, the experimenter could supply a desired configuration of node radio connectivities or higher-level properties such as bit error rate and the system would choose the set of real nodes that most closely match that configuration. This will require prior measurements of the NxN inter-node signal strength or link characteristics, ideally while selected combinations of other traffic is flowing. Solving this problem in discrete optimization should be feasible. This approach offers better hope of precision, but has the drawback

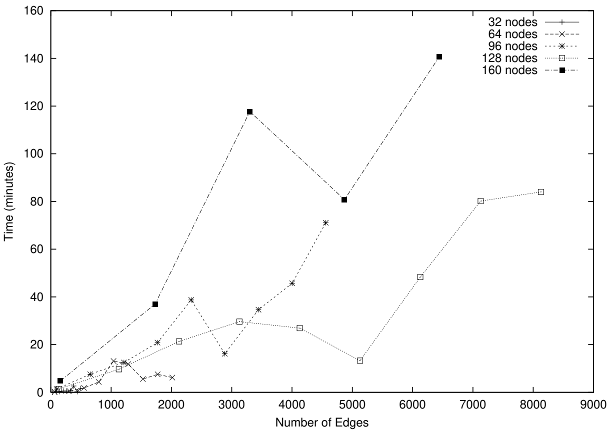

Figure 2: Time taken by genetic algorithm to map a variety of wide-area topologies

that experimenters typically don't know the connectivity characteristics of their target environment.

This last approach is analogous to the way we currently map wide-area resources. In our system, when an experimenter requests actual wide-area nodes and specifies desired inter-node path characteristics, we use a genetic algorithm to find the best-matching set of physical nodes and their corresponding Internet paths. Input to this algorithm consists of NxN latency and loss data, updated frequently. The wireless resource allocation problem is similar to this wide-area IP-path-matching problem, since the broadcast medium fully connects all nodes (in principle). Our experience with the wide-area algorithm suggests that the wireless problem will be challenging, but—with appropriate optimizations—tractable, well into the hundreds of nodes.

Our results on a simulated physical topology of 256 nodes are shown in Figure 2. All experiments with 32 nodes, as well as all sparse topologies, mapped in a few minutes. For larger and more dense topologies, up to 256 nodes and approximately 40 edges/node, mapping time ranged from 10 minutes to 2 hours.

We expect to improve that by an order of magnitude using the following techniques: less stringent and more clever termination conditions; standard optimization techniques, in particular memoizing; and parallelizing the algorithm, which is practical in either a shared memory multiprocessor or on a cluster [21]. In the "Island Model," mutations and crossovers can be done in parallel, sharing the best solutions periodically; we estimate synchronizing every few seconds to exchange 1–2KB of data. Other algorithmic approaches may also be relevant. For example, constraint programming [22], a method guaranteed to find a feasible solution if one exists or report otherwise, should produce better results for problems tightly constrained by heavy utilization of physical resources.

Finally, we expect major additional improvement to come from "binning" the nodes and links into groups with similar characteristics, which will dramatically reduce the search space. The result should be an algorithm that can map hundreds of nodes and links in a few minutes.

3.2 Interference

To retain Netbed's efficient use of space and resources, wireless experiments should be isolated from one another and from the environment to the greatest extent possible. Interference from unrelated "production" traffic or from devices such as microwave ovens, Bluetooth and cordless phones in the 2.4 GHz band may lead to anomalous experimental behavior. Unfortunately, there is an inherent conflict in the unlicensed spectrum: popular technologies are the most interesting to study, but also the most likely to be incidentally present in the environment. There are three recourses: using buildings or areas that are not used for such traffic, negotiating for a subset of channels to be unused for production, or studying devices that are still on the upswing in popularity. All of these are reasonable approaches.

One of these techniques, the use of multiple channels, should also alleviate potential interference caused when multiple experiments occupy intersecting transmission ranges or collision domains. Wireless devices with overlapping collision domains can retain separate conversations by occupying *non-overlapping* frequencies. To enforce isolation, Netbed would take this additional constraint into account during the mapping phase.

In the wired emulated arena, Netbed leverages VLAN (Virtual LAN) switch technology to enforce isolation of experiments between separate network partitions. While each experiment may be readily mapped to a distinct VLAN, the number of non-overlapping frequencies within 802.11b's available spectrum is more limited. Given its channel width of 22 MHz, only 3 of the 11 channels available (in the United States) are non-overlapping [14]. Since 802.11a supports 8 independent, non-overlapping channels [13], it may be a more suitable technology to support a dense mesh.

An alternate approach to achieving greater density reduces transmission power; this effectively decreases both the range and attendant collision domain. In an analogy to graph coloring, because there are fewer overlapping domains, fewer "colors" or non-overlapping channels are required for isolation. This technique avoids a large physical footprint by simultaneously scaling down both the transmission power and inter-node distance. There is a caveat; reduced transmission power may not be indicative of real scenarios and may suffer some loss of realism.

An active channel may be assigned to a new experiment if the experiments in question do not share overlapping collision domains. To prevent an overly aggressive channel reuse strategy that would lead to co-channel interference, the wireless devices will be placed according to a network plan that aims to reduce interference while retaining high density. After placement, the ranges of each device will be carefully measured to be used later by the mapping algorithm to determine collision domains. Online wireless channel monitoring will be employed in all the nodes and made available to experimenters so that results from experimental runs with unacceptably high interference levels could be discarded or dealt with in a suitable manner by the experimenter.

4. MOBILITY

Configurable mobile environments are critical in the evaluation of many wireless systems, since coping with mobility is often the hardest part in the design of, for example, ad-hoc routing algorithms. To extend Netbed to provide actual mobile nodes, we will deploy a large, dense set of wireless devices via passive couriers that move predictably in time and space. We will use two types of couriers: students moving from class to class with radio-equipped PDAs, and city and campus busses with wireless PCs. Both exhibit predictable movement patterns. We expect that it is the general dynamics of a scenario rather than the tracking of individual nodes that make it interesting. Therefore, experimenters will specify a desired scenario (e.g., students wandering the halls, students eating in the cafeteria, etc.) and the density of its constituent nodes.

For those experiments where the precise relative motions of each node is significant, the Netbed software could manage automated couriers in the form of GPS-equipped radio-controlled cars that navigate a large open space. Differential GPS with RTK can localize to 1 cm when a correction signal is provided every second [17].

Couriers will present a previously unavailable source of live, realistic scenarios. They will offer the following advances over current approaches:

- Couriers remove the dependence on inaccurate simulation models.
- Couriers, by definition, will provide scenarios that are representative of real world behavior. An understanding of their behavior can guide ad hoc protocols in the same manner that file access characterizations [2] influence file system design. By tracking the location of the couriers we will be able to develop models or traces of their behavior to be used in simulation.
- Since they are governed by bus or class schedules, couriers will provide regular behavior, though not complete repeatability. Movement "noise" that deviates from the intended schedule will enable an accurate study of predictable mobility because of its consistency with real world behavior. Such coarse predictive capabilities were suggested as an extension to Grid's location service (GLS) [16].

Experimenters will have the option of selecting from a subset of students sitting in classrooms, wandering the halls, or eating lunch, which constitute a broad range of mobility. Experimental code will execute on the students' PDAs. Repeatability will be approximated by running experiments to coincide with defining periods in the schedule, for example when students disembark from the bus in the morning or every hour when they leave their classrooms. Such scenarios capture proposed ad hoc network activities, such as classroom interaction. Further, their breadth of movement patterns will provide an opportunity to study a protocol's steady-state behavior and to examine its agility in the face of transient topology changes.

The passive courier approach does not afford the perfect repeatability of simulation. Nevertheless, experience indicates that realism and ease of setup are invaluable, even without complete repeatability. In addition, experiments can be run many times, since the process is fully automated and can be run in batch mode. Simple statistical techniques (read "the average") can help compensate for lack of perfect repeatability.

City busses follow a unique mobility pattern that suggests alternate classes of protocols. Unlike classroom or random motion settings, the number of neighbors reachable by a particular node in a bus transit scenario will change dramatically over time. For example, there are typically 20 busses within a half mile radius of downtown Salt Lake City; more densely populated cities presumably have greater bus density. A cursory glance at bus schedules similarly indicates the potential for interesting multi-hop topologies, assuming antennas are deployed to reinforce a radio's nominal range. After leaving downtown, bus density decreases greatly.

This pattern is conducive to a cooperative long-term store and forward approach based on prediction. The resulting network is a hybrid since it exhibits the dynamism of an ad hoc network and the known (future) topology of a wired network. A bus node might selectively transmit data to a passing bus based on its destination. In such a scenario, sparingly transmitting data prevents pollution of the radio spectrum. For example, busses might engage in highway congestion monitoring [19], wherein an outbound bus only forwards downtown traffic updates to those busses likely to experience congestion.

A network composed of student couriers riding on school bus couriers provides *levels* of coordination, or an ad hoc hierarchy.

While riding on the bus, students are largely stationary and their motion is uninteresting. During this time, a bus node might act as a router for student traffic, perhaps performing aggregation. When students arrive at school and leave the bus, they once again assume the role of first-class peers. Such a scenario has military parallels in which mobile divisions may need to coordinate with upper echelons [15].

5. CONCLUSION

Shared wireless and mobile testbeds would dramatically lower the costs and barriers to emulation and live network experimentation. They will bring the ease of use, configurability, *ns*-compatibility, and transparency of the existing Netbed infrastructure to the wireless and mobile domains. By facilitating and automating large-scale mobile and wireless experimentation, we expect such testbeds to gain widespread adoption, leading to sets of community-accessible reference platforms. Such platforms promote comparable results, encourage validation, and will advance the state of the art in experimental design, setup, and execution.

6. REFERENCES

[1] S. Bajaj et al. Improving Simulation for Network Research. Technical Report 99-702b, USC, March 1999.

[2] M. G. Baker, J. H. Hartman, M. D. Kupfer, K. W. . Shirriff, and J. K. Ousterhout. Measurements of a distributed file system. In *Proc. of 13th ACM SOSP*, pages 198–212, 1991.

[3] J. Broch, D. A. Maltz, D. B. Johnson, Y.-C. Hu, and J. Jetcheva. A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols. In *Proc. of the Fourth Annual ACM/IEEE MOBICOM*, October 1998.

[4] S. M. Cherry. What Went Wrong at Ricochet. *IEEE Spectrum*, 29(3):60–61, Mar. 2002.

[5] J. Heidemann, N. Bulusu, J. Elson, C. Intanagonwiwat, K. chan Lan, Y. Xu, W. Ye, D. Estrin, and R. Govindan. Effects of Detail in Wireless Network Simulation. In *Proc. of the SCS Multiconference on Distributed Simulation*, pages 3–11. USC/ISI, January 2001.

[6] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, and K. Pister. System Architecture Directions for Networked Sensors. In *Proceedings of the 9th Intl. Conf. on Architectural Support for Programming Languages and Operating Systems*, pages 93–104, Nov. 2000.

[7] P. Johansson, T. Larsson, N. Hedman, B. Mielczarek, and M. Degermark. Scenario-based Performance Analysis of Routing Protocols for Mobile Ad-hoc Networks. In *Proc. of the Fifth Annual ACM/IEEE MOBICOM*, pages 195–206, August 1999.

[8] D. Johnson, C. Aragon, L. McGeoch, and C. Schevon. Optimization by Simulated Annealing: An Experimental Evaluation; Part I, Graph Partitioning. *Operations Research*, 37(6):865–893, 1989.

[9] D. B. Johnson et al. The CMU Monarch Project's Wireless and Mobility Extensions to ns. Technical report, Carnegie Mellon University, August 1999.

[10] J. Kaba and D. Raichle. Testbed on a Desktop: Strategies and Techniques to Support Multi-hop MANET Routing Protocol Development. In *Proc. of the 2001 ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc'01)*, Oct. 2001.

[11] P.-H. Kamp and R. N. M. Watson. Jails: Confining the Omnipotent Root. In *Proc. 2nd Intl. SANE Conference*, May 2000.

[12] Q. Ke, D. A. Maltz, and D. B. Johnson. Emulation of Multi-Hop Wireless Ad Hoc Networks. In *Proc. of the Seventh International Workshop on Mobile Multimedia Communications*, October 2000.

[13] LAN/MAN Standards Committee of the IEEE Computer Society. IEEE Std 802.11a-1999. Technical report, IEEE, 1999.

[14] LAN/MAN Standards Committee of the IEEE Computer Society. IEEE Std 802.11b-1999. Technical report, IEEE, September 1999.

[15] B. M. Leiner, R. J. Ruth, and A. R. Sastry. Goals and Challenges of the DARPA GloMo Program. *IEEE Personal Communications*, December 1996.

[16] J. Li, J. Jannotti, D. S. J. De Couto, D. R. Karger, and R. Morris. A Scalable Location Service for Geographic Ad Hoc Routing. In *Proc. of the 6th Annual Intl. Conf. on Mobile Computing and Networking*, Aug. 2000.

[17] D. A. Maltz, J. Broch, and D. B. Johnson. Experiences Designing and Building a Multi-Hop Wireless Ad Hoc Network Testbed. Technical Report CMU-CS-99-116, Carnegie Mellon University, March 1999.

[18] D. A. Maltz, J. Broch, and D. B. Johnson. Lessons from a Full-Scale Multi-Hop Wireless Ad Hoc Network Testbed. *IEEE Personal Communications*, Feb. 2001.

[19] R. Morris, J. Jannotti, F. Kaashoek, J. Li, and D. Decouto. CarNet: A Scalable Ad Hoc Wireless Network System. In *Proc. of the Ninth ACM SIGOPS European Workshop*, Kolding, Denmark, Sept. 2000.

[20] B. Noble, M. Satyanarayanan, G. T. Nguyen, and R. H. Katz. Trace-Based Mobile Network Emulation. In *Proc. of SIGCOMM '97*, September 1997.

[21] R. Tanese. The Distributed Genetic Algorithm. In *Third International Conference on Genetic Algorithms*, pages 434–439. Morgan Kaufmann, 1989.

[22] P. Van Hentenryck. *Constraint Satisfaction in Logic Programming*. Logic Programming Series, The MIT Press, Cambridge, MA, 1989.

[23] 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 5th Symposium on Operating Systems Design and Implementation*, Boston, MA, December 2002.

[24] X. Zeng, R. Bagrodia, and M. Gerla. GloMoSim: A Library for Parallel Simulation of Large-scale Wireless Networks. In *Proc. of the 12th Workshop on Parallel and Distributed Simulation*, pages 154–161, May 1998.

[25] Y. Zhang and W. Li. An Integrated Environment for Testing Mobile Ad-Hoc Networks. In *Proc. of the 2002 ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc'02)*, June 2002.