# User Experiences on Network Testbeds

Jelena Mirkovic
sunshine@isi.edu
USC Information Sciences Institute
USA

Portia Pusey
edrportia@gmail.com
Portia Pusey LLC
USA

## ABSTRACT

Network testbeds are used by researchers to evaluate their research products in a controlled setting. Teachers and students also use network testbeds in classes to facilitate active learning in authentic settings. However, testbeds have scarce human resources to develop documentation or support users one-on-one. Therefore, using testbeds can be difficult, especially for novice users. A user's lack of experience, coupled with user support deficiencies, can turn into research or learning obstacles.

In this paper we report on two surveys we administered to investigate and document possible obstacles in user interaction with network testbeds. In the first survey we conducted *interviews* with 13 students that used a network testbed in class. Informed by their answers, we created the second, more comprehensive *online* survey and circulated it to both research and education users of network testbeds. We received 69 responses. User responses indicate three broad sources of usability challenges: *orientational* – learning a new environment, *implementational* – setting up and running experiments and *domain-specific* – monitoring experiments and diagnosing failures. Responses further show that most users overcome their initial orientational obstacles, but that implementational and domain-specific obstacles remain and should be addressed by testbeds through significant new developments. Overall, users regard network testbeds as a positive and useful influence on their learning and research.

## CCS CONCEPTS

• **Social and professional topics → Information technology education**; • **Human-centered computing** → *Empirical studies in HCI*.

## KEYWORDS

testbeds, learning, usability, user experience

**ACM Reference Format:**
Jelena Mirkovic and Portia Pusey. 2021. User Experiences on Network Testbeds. In *Cyber Security Experimentation and Test Workshop (CSET '21), August 9, 2021, Virtual, CA, USA*. ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3474718.3474728

## 1 INTRODUCTION

There are many large network testbeds today, such as Cloudlab [7], Chameleon [9], Deterlab [2, 6], and COSMOS [16]. These testbeds are extensively used for networking, operating systems, and cybersecurity research. They provide access to specialized hardware and allow users to configure the nodes, the operating system and the network substrate in myriad ways. This is useful to evaluate research products in a wide range of settings, via controlled, repeatable experiments. Network testbeds are also widely used in education, to facilitate active learning, train students in useful skills and teach adversarial thinking [3, 11–15, 18].

While there are numerous potential benefits of testbeds in supporting research and education, little has been done to date to investigate and quantify potential *obstacles* users face when interacting with testbeds. Anecdotally, operators have discussed that these obstacles may arise from the mismatch between a specific user's knowledge or skills and the testbed interfaces (e.g., a testbed offers Linux command-line interface and the user has no experience with Linux), or they may arise from a testbed's lack of documentation and staff time to support users.

In this paper we report on two surveys we conducted in 2020 to understand the obstacles users face in network testbed interactions, and to map out a range of possible solutions. We first conducted a series of online interviews with students who used testbeds in education, to understand in depth the obstacles they faced and their thoughts on how such obstacles could be addressed. Informed by these findings, we launched an online survey and circulated it widely to different testbeds' user groups. This resulted in 69 responses from a mix of research and education users. Most of these users are students using testbeds for their research or using them in a class.

Our main findings include:

**Many users face obstacles around learning a new environment** (testbed) and how to interact with it. These obstacles arise regardless of a user's skill level, and are quickly overcome as users become more proficient.

Even after a long tenure with testbeds, **users continue facing obstacles in experiment design, monitoring and failure diagnostics**. Addressing these issues requires a large shift in how testbeds support experiments, extending the support to the entire experiment lifecycle. Experiment design could be better supported through worfklows, sample experiments and building a shared experiment repository. Experimentation tools (building blocks for design stage), monitoring and diagnostics are likely to be domain-specific. Building and supporting user communities around the same research domain, and providing shared spaces where they can exchange tools and datasets would help address these issues.

**Users are generally positive about their experience with testbeds**, and how helpful testbeds are to their research and education. Interactions with testbed staff are also rated very positively.

To support reproducible research, we release digitized responses to our online survey, as well as accompanying processing scripts at: https://steel.isi.edu/LearningWithTestbeds/.

## 2 OBSTACLES IN TESTBED INTERACTION

A user may face multiple obstacles when interacting with a testbed. We observe them roughly as three broad categories, depending on where we believe interventions may be most effective: user-specific, task-specific and testbed-specific obstacles.

### 2.1 User-specific obstacles

User-specific obstacles stem from a misalignment between a user's knowledge/skills and testbed interfaces. For example, a user may lack familiarity with the testbed's interface, such as Linux command-line interface. User-specific obstacles can usually be overcome with an instructional intervention; for example additional user training, better documentation, tutorials and workshops.

### 2.2 Task-specific obstacles

Task-specific obstacles arise when a user attempts to accomplish a task that exceeds the capability of the given testbed. This often happens with research tasks; when a user requires a specific support (e.g., storage of large OS images) that the testbed does not regularly provide. Since task-specific obstacles are usually unique to a single researcher or research team, overcoming them most often involves one-on-one interaction between the user and testbed staff. In some cases an obstacle cannot be overcome and the user leaves the testbed.

### 2.3 Testbed-specific obstacles

When building a testbed its team must make many decisions ranging from which hardware and software to buy, to which user interfaces to support, and how to develop and maintain documentation and code. Some of these decisions can inadvertently create obstacles for many users (see 2-hop login in Section 2.4), and warrant changing the testbed itself.

### 2.4 Surveying testbed users

Network testbeds are very different, thus designing a common survey that applies to most or all of them is hard. First, testbed technologies are very broad. Some testbeds allow users access to virtual machines, while others offer access to physical machines. Yet other testbeds focus on non-standard hardware, such as SDN switches, GPU nodes, or wireless nodes. This diversity of resources necessarily leads to diversity of user interfaces, and diversity of use cases and user experiences. Along with testbed offerings, their outreach programs and user recruitment strategies also lead to diversity in and uniqueness of each testbed's user population. All these factors make it hard to investigate testbed users as a homogeneous population. Ideally, one would design a variety of studies for each testbed, to understand how well it meets the needs of its different user populations, e.g., novice vs expert users, research vs education users, majority vs minority users, US vs international users, users in each scientific discipline, etc. Such segregated surveys are very hard to conduct, since testbeds usually have hundreds to thousands of active users, and only a small fraction of them may respond to a volunteer survey. For instance, while our survey was distributed widely to testbed-specific as well as discipline-specific mailing lists, we only had 13 participants in our interview survey and 69 participants in our online survey. We estimate that this is well under 1% of the total testbed user population.

We address the issues around different testbeds and user populations by looking for common features across testbeds to design broadly applicable user surveys. These common features are described below.

**SSH** – Many testbeds offer resources that are accessible via SSH. In the past we have observed that students in classes struggle to learn SSH and `scp` (SSH version of file copy command) syntax.

**Linux** – Many testbeds support versions of Linux OS (e.g., Debian, CentOS, Fedora), and a few may support other operating systems (FreeBSD and Windows). For a user familiar only with Windows OS, Linux may represent an obstacle.

**Command-line interface** – Testbeds often offer only a command-line interface or CLI (terminal application) for node access. Users that have used only a graphical interface (e.g., on their own laptop or desktop) may feel challenged when faced with CLI.

**Distributed experimentation** – Users of network testbeds often have to interact with more than one node simultaneously. This can be confusing to users that are used to only interacting with their local laptop or desktop. These users may struggle to keep track which terminal window (on their local machine) corresponds to which remote node.

**Two-hop login** – Some testbeds allow users to log in directly into their experimental nodes, while others require users to first log into a gateway machine, and then from that machine into their experimental nodes. We call this process "2-hop login". The 2-hop login can become an obstacle, because it is a detour in user's path to the experimental nodes.

**Shared directories** – Even though users may access multiple experimental nodes, some testbeds map the user's home directory to a shared folder. Thus on each machine the user's home directory is the same. This may run contrary to the user's expectations and become an obstacle.

**Long experimentation** – Most experiments are complex, requiring users to create tools and set up applications on their nodes. Thus an experiment may take days, weeks or even longer. Coming back to an experiment after a while can create mental burden for a user, because they may forget where they left off. Research experiments are also unique, designed by the user to answer a specific research question. Unless the user takes very detailed notes, they may forget what they did, and whether it worked or not. We call this issue a *mental context switch*.

Another burden caused by long experimentation occurs when a user must return resources to the testbed, e.g., because their allotted time has run out. Most testbeds do not save user state after the experimental resources are returned. Instead, the burden is on the user to save any local files to a persistent storage, and to restore the state when they want to resume experimentation. We call this issue *resource context switch*.

**Empty slate** – It can be daunting for a user to design their experiment from scratch. A novice user may not know where to start or what can be done on a given testbed. Even an experienced user may take a long time to develop their initial idea of an experiment into the final version that works as they intended.

When designing user surveys it is challenging to identify best types of questions to ask. Questions that are too general or open-ended (e.g., "Tell us about your testbed experience") can result in little actionable feedback. On the other hand, questions that are too specific (e.g., asking about very specific obstacles) may miss the opportunity to learn about new issues that the survey did not cover. We addressed this challenge in two ways. First, we conducted minimally-guided open-ended interviews to learn about common obstacles faced by students when they use testbeds in classes. We kept recruiting students for these interviews until the responses we received stopped identifying new obstacles. Using this information we next crafted questions for our online survey. In this survey we primarily used multiple-choice, guided questions to obtain actionable feedback. We added several open-ended questions at the end, to help us learn about any obstacles or interventions we may not have foreseen.

## 3 METHODOLOGY

This paper reports on two user studies conducted to understand obstacles users face when using network testbeds. Our research goals in both studies were to: (1) understand what different obstacles users face, (2) investigate how these obstacles evolve over time, i.e. if users learn to compensate for these obstacles or overcome them, and (3) solicit user input about the possible solutions.

Our *interview* study was designed to investigate obstacles faced by students that use testbeds in classes. Two members of our team – one testbed developer and one evaluation specialist – spent 30 minutes with each user, discussing their experiences. Table 1 shows the list of questions we used to guide our discussion. We started with a list of possible obstacles, but kept our questions open for anything else a user may bring up. Each user was compensated 15$ for a 30-minute interview. We advertised the survey to all educational users of Deterlab and EduRange testbeds – around 1,000 users. Only 13 of them volunteered.

After completing our interview study, we used the findings to design questions for our *online* study. We also broadened the scope from learning about educational users to learning about all testbed users. Table 2 summarizes the questions used. We provide the exact wording for the survey in the Appendix. Participants responded to questions on their own timeline and were not compensated for participation. We advertised the survey on several mailing lists, and we also asked testbed operators to forward the survey to their user base; 69 users completed the survey.

### 3.1 Participant statistics

Table 3 shows respondent's breakdown by age and gender, the testbed they used and how they first came to use a testbed (entrance path), as well as user's prior experience with SSH or Linux. No personally identifiable information was collected about participants. Both user studies were evaluated and approved by our Institutional Review Board.

| num | question |
|---|---|
| 1 | Did you ever use a testbed like Deterlab/Edurange before? Which one? |
| 2 | Did you ever use Linux before? What did you do on Linux? |
| 3 | Did you struggle with some exercises on Deterlab/Edurange? Which ones? |
| 4 | For each exercise ask where they struggled and why |
| 5 | Ask if distributed environment (having to use multiple machines at the same time) was an issue? |
| 6 | Ask if using Linux command line was an issue? |
| 7 | Ask if having to work remotely was an issue? |
| 8 | Ask if working over many days as opposed in one sitting was an issue (context switching)? |
| 9 | Ask what would have helped make the experience better |

**Table 1: Interview study questions.**

### 3.2 Limitations

Our studies have the following limitations that stem from their design and the participant pool size.

**Volunteer bias.** It is known that people volunteering for a study may not faithfully represent the entire target population. In our case, users volunteering for our studies could also exhibit *survivor bias*, i.e., they may come from the population of users who either did not face obstacles or who have successfully overcome them. Our study does not survey users that have attempted to use a testbed and quit, or those that considered using a testbed but decided against it. These are important future research directions.

**Focused on a small number of testbeds.** While we have tried to advertise our studies broadly, the actual volunteers we managed to recruit are biased toward Deterlab users. Our online study has a sizable population of users of other testbeds, but majority of users still come from Deterlab's user population. Our questions were also tailored toward Emulab/Cloudlab/GENI/Deterlab/Chameleon testbeds, and thus may miss obstacles that stem from a different testbed technology or a different environment.

**Small and non-diverse sample of users.** Our participants are just a small sample of the entire user population of the network testbeds which numbers in thousands. Thus our survey likely covers under 1% of all network testbed users. Looking at participant statistics (Table 3), our volunteers were mostly male (76%), majority has used one testbed (66%) and about 80–85% self-rated their experience with SSH or Linux as moderate or expert. Thus our findings may be limited to this user group.

Our volunteers were relatively evenly divided between class and research users (72% have used a testbed for research, and 58% have used it in class, 33% used it for both research and class), allowing us to study experiences of these two modes of testbed use. Volunteers were also evenly divided between those that were novices (0–1 year of use), moderately experienced with testbeds (2–5 years of use) and very experienced (more than 5 years of use).

We acknowledge these limitations, and hope that future studies of testbed users can recruit a larger and more representative set of volunteers. One way to do so would be for testbeds to adopt a common set of survey questions and present them periodically (e.g., once a year) to a user upon login. The user could be required to answer the questions before moving on to the next screen. While such a study would not be voluntary, it would ensure that all user groups are well represented among respondents.

| num | question | resp. type |
|---|---|---|
| 1 | age | multi-choice |
| 2 | gender | multi-choice |
| 3 | testbed used | free text |
| 4 | research or edu | multi-choice |
| 5 | knowledge of Linux | Likert 5 pt |
| 6 | knowledge of SSH | Likert 5 pt |
| 7 | first year of usage | free text |
| 8 | for that first year of use, score various testbed activities by ease of use | Likert 5 pt |
| 9 | last year of usage | free text |
| 10 | for that last year of use, score various testbed activities by ease of use | Likert 5 pt |
| 11 | select obstacles you have encountered in your early testbed usage | multi-select |
| 12 | select among possible interventions those you believe would help | multi-select |
| 13 | any other help suggestions | free text |
| 14 | overall how would you rate your experience with testbeds | Likert 5 pt |
| 15 | overall how much did testbeds help your learning or research | Likert 5 pt |

**Table 2: Online study questions (summarized).**

| feature | interview | online |
|---|---|---|
| **age** | | |
| 18-24 | n/a | 11 (16%) |
| 25-34 | n/a | 18 (26%) |
| 35-44 | n/a | 19 (27%) |
| > 45 | n/a | 16 (23%) |
| prefer not to say | n/a | 5 (7%) |
| **gender** | | |
| male | 8 (62%) | 56 (81%) |
| female | 5 (38%) | 9 (13%) |
| other | 0 | 1 (1%) |
| **testbed used** | | |
| Deterlab | 12 (92%) | 53 (76%) |
| EDURange | 2 (15%) | 0 |
| Emulab/Cloudlab | 0 | 18 (26%) |
| Chameleon | 0 | 8 (11%) |
| Other | 0 | 18 (26%) |
| **num testbeds used** | | |
| one | 12 (92%) | 43 (62%) |
| two | 1 (7%) | 14 (20%) |
| three and more | 0 | 13 (19%) |
| **length of use** | | |
| 0-1 years | n/a | 23 (33%) |
| 2-5 years | n/a | 24 (35%) |
| > 5 years | n/a | 19 (28%) |
| not specified | 13 (100%) | 3 (4%) |
| **entrance path** | | |
| research only | n/a | 27 (39%) |
| class only | 12 (92%) | 17 (25%) |
| class then research | n/a | 10 (15%) |
| research then class | 1 (7%) | 13 (19%) |
| **SSH experience** | | |
| none or little | 6 (46%) | 14 (20%) |
| moderate and above | 7 (54%) | 55 (80%) |
| **Linux experience** | | |
| none or little | 3 (23%) | 11 (16%) |
| moderate or expert | 10 (77%) | 58 (84%) |

**Table 3: Participant statistics (some groups do not add to 100% due to missing or overlapping responses).**

## 4 FINDINGS

In this section we summarize our findings from the interview study (Section 4.1) and the online study (Section 4.2)

### 4.1 Interview study

Our interviews identified multiple obstacles for our participants. All participants in the interview study talked about difficulties around SSH-based access to experiment nodes. Some participants experienced issues themselves, while others recounted their classmates' experiences. Users mostly had trouble navigating the testbed. The issues they mentioned were: finding out names of machines to access, performing 2-hop login, remembering to run commands on the experimental node and not on the gateway machine (i.e., perform the full 2-hop login), understanding shared directories and understanding SSH command syntax. Working on a remote node as opposed to working on their laptop did not pose issues for most users. Only one participant indicated that this was a problem, and two more said it was an obstacle at first, but that they quickly got used to it. The rest of participants had prior experience in working remotely.

Distributed experimentation, accessing multiple nodes in the same experiment at the same time, posed issues for all but one participant that had prior system administration experience. Participants detailed strategies they developed to cope with this issue, such as using the command prompt to remind themselves on which node they were on, looking up the IP address of their current node and arranging terminal windows on the screen in a way that corresponded to their experimental topology.

Another common difficulty reported by the interview subjects was transferring files into their experiments. This included the issues around scp syntax, understanding shared directories and installing special software on Windows machines to support file transfer over SSH (pscp).

A significant issue for almost all participants was resource context switching: losing experiment state when physical resources are returned to the testbed and having to restore it to continue experimentation. The interviewees stated they developed strategies to deal with this issue, such as writing down which commands they executed so they could restore the state, creating scripts to run to restore state, and forcing themselves to complete their class assignment in one sitting to avoid context switching.

During the interview, we also discussed what could have helped to make the interviewee's experience smoother. Participants suggested improving documentation and user error messages, and creating video tutorials showcasing common testbed interactions. They also suggested enabling users to access terminal windows of their experiment nodes from the Web UI, thus bypassing SSH altogether. Further, to address the resource context switch, participants suggested developing an ability for users to save and restore experiment state. They also suggested that testbeds should log user actions and make these logs available to users, to help remind them of their prior interactions and ease mental context switch.

We noted that in our interviews with study participants they were overwhelmingly positive about their interactions with testbeds. Even though these interactions were frustrating at first, users found ways to overcome difficulties, and felt that this improved their skills and aided their learning.

We used both the obstacles identified in this study, and the suggestions for improvement to devise questions for our online study.

## 4.2 Online study

Online study allowed us to further quantify the obstacles the users face, and how they cope over time.

**Difficult activities: design, diagnostics and monitoring.** We first investigated which activities users found difficult. Figure 8(a) shows the user ratings on a 5-point scale with options (1-Very Easy, 2-Easy, 3-OK, 4-Difficult and 5-Very Difficult) based on the user's *early* experiences with testbeds. Most activities were rated as either very easy, easy or OK by a majority of users, such as creating an account, interacting with Web-based user interface, creating experiments and SSH-ing into them, and various file transfer operations. The fact that such activities are easy for novice users indicates that testbed documentation is sufficient to support new users in these tasks. A few activities stand out as more challenging: diagnosing experiment creation failures (such as when no suitable nodes are found), designing experiments, experiment monitoring and diagnosing/fixing experiment failures at runtime. These activities are naturally more challenging for users, for two reasons. First, diagnosing failures requires a deep understanding of testbed infrastructure and an expert level of OS/networking knowledge. Second, testbeds mostly focus on experiment provisioning (node allocation and setup) and offer poor support for experiment design, monitoring and diagnostics at runtime.

**Improvement over time for most activities.** Figure 8(b) shows the user ratings on the same 5-point scale, for the same activities (minus account creation) based on the user's *recent* interactions with testbeds. Comparing these recent experiences with early ones, we can learn which activities become easier for users over time, and which remain challenging. To compare the early and recent ratings, we use the Mann Whitney U test [4]. This test compares means of measurements within two populations, and evaluates if differences in means stem from differences between populations, or from variance within each population. In our case, the test helps us evaluate if differences between early and recent experiences by the same user population indicate true change in ease of interaction with testbeds. Since we perform many comparisons on the same population, we apply Benjamini and Hochberg [1] correction to ensure that our tests still have statistical significance. If the corrected $p$-value for Mann-Whitney U test is below 0.05 we say that two groups of ratings are significantly different from each other. We show the $p$-values that indicate statistically significant change in Figure 8(b), with labels on the right.

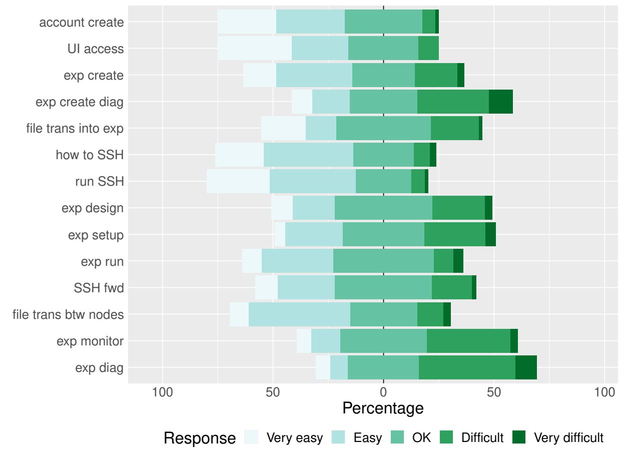
We find statistically significant improvements in experiment creation, diagnostics of experiment creation failures, file transfer both into the experiment and between experimental nodes, experiment setup and running, and setting up of SSH forwarding. All these are routine activities that should become easier over time with more practice and as users become more familiar with a testbed environment. Our results confirm this – the recent ratings are visibly shifted to the left, compared to the early ones. Other routine activities, such as accessing testbed UI, discovering how to SSH into the experiment and performing SSH did not become much easier, but they were already easy in early interactions, so there was little room for improvement. Similarly, experiment design, monitoring experiments and diagnosing run failures did not change. They remained difficult for around 25–30% of respondents. These are the areas that would most benefit from interventions.

**Few differences between class and research users.** We compared the ratings for early and recent experiences between the groups of research-only and class-only respondents (we removed those participants that used testbeds both in their research and in their classes). In most cases, there was no statistically significant difference in ratings. Exceptions were experiment setup (early experiences) and experiment running (recent experiences). Both activities were on the average harder for research (setup.pre=3.2, run.post=2.5) than for class (setup.pre=2.7, run.post=2) users. We attribute this to higher difficulty of research tasks compared to class tasks. We conclude that entry path does not affect a user's experience with testbeds for most activities.

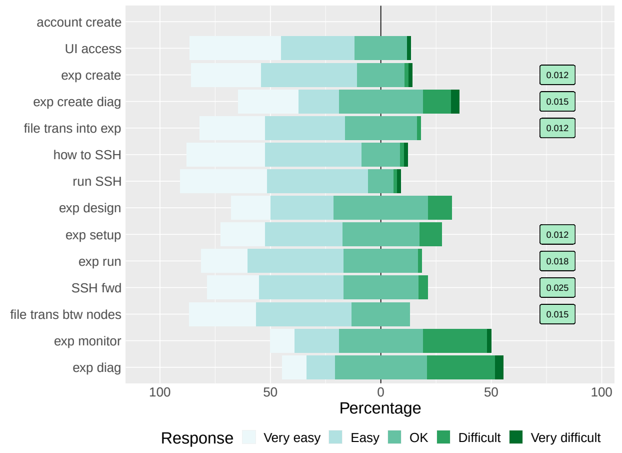
**Some difference between users proficient in Linux/SSH and novices.** We compared the ratings for testbed experiences between those users that rated their SSH or Linux expertise as 3, 4 or 5 indicating medium and above expertise, and users that were novices (rating 1) or beginners (rating 2). There was no statistically significant difference in early or recent experiences for most tasks between these user groups. The only significant differences showed in *early experiences* with four tasks related to use of SSH and shared file system – understanding how to run SSH, running SSH, transferring files between nodes (but not transfer into experiment) and SSH forwarding. Thus most obstacles affect all users similarly, irrespective of their previous knowledge and skills.

**No difference in experiences among testbeds.** We investigated if users on different testbeds may have faced different obstacles and show these results in the Appendix. There was no statistically significant difference between testbeds.

**No difference based on length of experience.** We further compared the ratings for early and recent experiences between those respondents who used a testbed for less than two years and those that used a testbed for five years or longer. There was no statistically significant difference in ratings. We conclude that length of interaction with a testbed may not significantly affect a user's experience. In other words – what can be learned is learned quickly,



(a) Early usage experiences



(b) Later usage experiences

**Figure 1: Online survey: user rating of selected activities (see 8.1) during their early and recent testbed usage.**

and truly difficult obstacles remain difficult. From our survey results, experiment design, monitoring experiments and diagnosing failures at run time remained difficult regardless of the length of a user's interaction with testbeds.

**Common obstacles: node allocation, file transfer, SSH, 2-hop login and context switching.** Figure 2 shows the percentage of respondents that indicated they faced a given obstacle. A third of respondents had trouble allocating nodes for their experiment. Around a quarter had trouble transferring files to and from an experiment, learning how to use SSH and `scp` , performing two-hop login/file transfer and dealing with context switching (restoring state after long pauses in interaction). Around a fifth of respondents struggled with finding out names of their nodes, and understanding shared directories. Less common were issues relating to creating experiments, setting up passwordless SSH access, dealing with multiple nodes in an experiment, and transferring files between nodes. The common obstacles represent a good opportunity for testbeds to improve user experience, since many of them can be addressed through improvements in user interfaces and documentation.

**Testbed staff was generally helpful.** Figure 3 summarizes participant responses concerning testbed staff. Almost half of the respondents indicated that testbed staff was helpful. Only 10% of the respondents said testbed staff was slow to respond to their requests for help – this is likely due to small size of testbed staff. Only one participant indicated that testbed staff was rude and two participants said staff was uninterested in helping. This level of user satisfaction is noteworthy, given challenges that testbed staff faces in allocating their time.

**Interventions: the more the better.** Figure 4 summarizes participant responses about interventions that may help users overcome the obstacles to testbed use. Participants were asked to rate each suggested intervention as: 1-Probably would not help, 2-Maybe it would help or 3-Yes, definitely it would help. One striking observation is that users rated almost all suggestions as very likely to

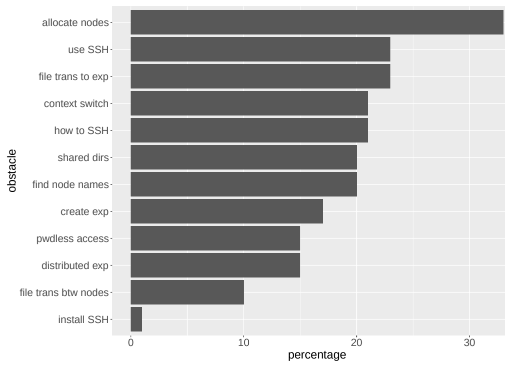


**Figure 2: Percentage of participants that experienced a given obstacle.**

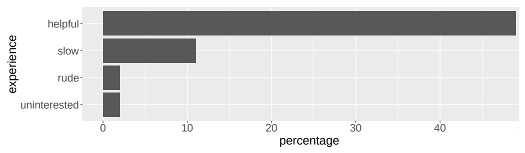


**Figure 3: Percentage of participants that had a given experience with testbed staff**

help. The only suggestion that received less enthusiastic response was development of Linux tutorials. Suggestions with the highest ratings were: (1) better tutorials and documentation (also rated highly was the suggestion to develop more tutorials and documentation), (2) a library of sample experiments (similarly, and highly rated was the suggestion to develop aids for experiment design),

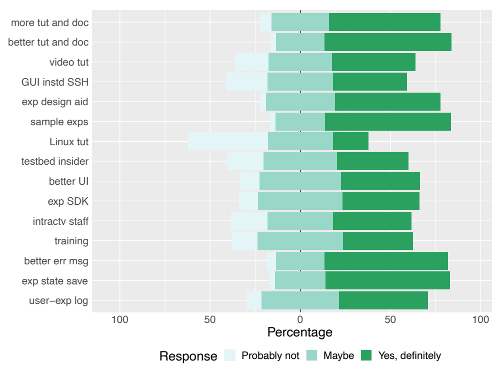


**Figure 4: Participant opinions of possible interventions.**

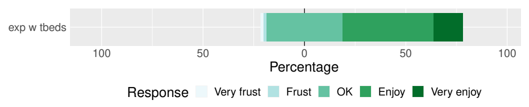


**Figure 5: Participant rating of overall testbed experience**

(3) better error messages and (4) ability to save and restore the experiment state.

**Users are positive about testbed interactions.** Figure 5 summarizes respondent rating about their overall testbed interaction, on the scale: 1-Very frustrating, 2-Frustrating, 3-OK, 4-Enjoyable and 5-Very enjoyable. Surprisingly, 60% of respondents found their interaction with testbed either enjoyable or very enjoyable in spite of the obstacles. Additional 37% found it OK. Only one respondent said their experience with testbed was frustrating, and another one said it was very frustrating.

**Testbeds are very helpful to users.** Figure 6 summarizes the respondents' rating about the helpfulness of testbeds for their their research or learning in class on the scale: 1-Not at all, 2-Slightly, 3-Moderately, 4-Very and 5-Extremely. Respondents were very positive; no one selected the "Not at all" option. There were 55% of the respondents who felt testbeds were extremely helpful for their research or learning, and additional 32% felt they were very helpful. Only 9% of participants felt testbeds were moderately helpful and 4% felt they were slightly helpful.

**Individual suggestions.** Qualitatively, 25 out of the 69 respondents provided suggestions clarifying their response to the helpfulness of the interventions. These suggestions went into more details about interventions from our questions, but did not suggest new intervention approaches. For example, participants suggested adding a simulated allocation to Web UI, to quickly inform users if they were requesting more nodes than are available. They also suggested removing 2-hop login, and organizing testbed documentation better.

## 5 DISCUSSION AND RECOMMENDATIONS

Table 4 summarizes our recommendations. User-specific obstacles can be addressed easily, but should not be a high priority for testbeds. The qualitative data suggested that users consider these

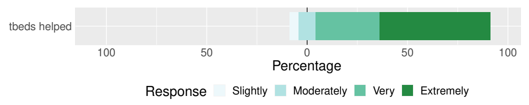


**Figure 6: Participant rating of how much testbeds helped their research or learning in class.**

obstacles as learning opportunities. We identify two categories of testbed-specific obstacles – orientational and implementational – as well as some task-specific obstacles, which we abstract under a larger, domain-specific category.

*Orientational* obstacles come with working in a new environment. Working with testbeds seems akin to learning a new language or moving to a new city. One must learn how to accomplish their goals in this new setting, which tools to use and how to invoke them. Some users struggle because they lack familiarity with some tools, such as SSH, Linux, or even command line interface. Other users struggle because testbed documentation is not well organized, or it is outdated, or it is too long and cumbersome to peruse. Yet others struggle because a testbed's Web UI is not intuitive enough to use, or error messages and allocation failures are difficult to understand. Our findings suggest that as users mature in their testbed use, they overcome orientational obstacles and emerge with a sense of accomplishment. Testbeds can easily address orientational obstacles by improving their UI and their documentation, to improve early user experiences.

*Implementational* obstacles stem from the way network testbeds are implemented today, with most user support for provisioning resources. Users are then left to their own devices to set up and use these resources. Users struggle with experiment design, starting from this empty slate. They also struggle with working with multiple nodes simultaneously using a terminal interface, which makes it difficult to determine which node the user is currently on. Finally, testbeds that do not use virtual machines make it difficult for users to save state when their resources are reclaimed, and to restore state when resources are provisioned again. Even if the state were easily saved and restored (resource context switch) users would still struggle to remember where they left off and what to do next (mental context switch).

Testbeds can address implementational obstacles by providing more sophisticated support for experiment setup and running. These obstacles require a substantially higher effort than addressing orientational obstacles – they require development of new front-end and back-end functionalities to support a completely different use model than in the past. Jupyter notebooks or logs of user interaction with the experiment could be used to ease mental context switch. They would also facilitate sharing between users, and would help with empty-slate design. We note that sharing needs additional support in terms of creating shared spaces where users can upload data, scripts and OS images to be used in the experiment they share. Similarly, logs of user activity (Jupyter notebooks or other kinds of logs) that a user has used for their own experimentation need to be adjusted to use these shared data/scripts/images, and they need to be tested by other users to ensure the shared experiment is usable and self-contained. These functionalities need additional testbed support. Finally, if users could interact with their nodes from some

| obstacle type | recommendation |
|---|---|
| orientational | improve UI and documentation |
| implementational | more support for exp. design and running |
| | design UIs that address human cognitive needs |
| | provide support for sharing and reuse |
| domain-specific | develop basic monitor./diag. tools |
| | provide support for sharing and reuse |
| | provide support for building user communities |
| | provide pathways for user tool maturation |

**Table 4: Summary of recommendations**

graphical UI instead of SSH-based terminals, working with multiple nodes simultaneously would become easier.

We abstract the task-specific obstacles into a larger category of *domain-specific* obstacles, because multiple users may be working to accomplish similar tasks in the same experimentation domain (e.g., congestion control). These users will likely have similar needs for experiment design tools, experiment monitoring and failure diagnostics. Domain-specific obstacles remain challenging even after using testbeds for a long time.

It is possible that domain-specific obstacles should be a focus of longer-term research and development in the testbed community. However, it is debatable if testbed developers can or should develop domain-specific tools, and support domain-specific monitoring and diagnostics. After all, such tools and support would vary greatly from user to user, and would require domain-specific knowledge as well as knowledge of the testbed infrastructure. It is hard to envision a large return on investment from such complex undertaking, serving small user communities.

On the other hand, testbeds today lack even the rudimentary monitoring and diagnostics, beyond node liveness and reachability. A viable path forward may consist of testbed developers surveying their users and developing some basic monitoring and diagnostics tools to support wide classes of experiments, e.g., computation vs networking vs storage experiments. If testbeds then also provide strong support for sharing and reuse, users in the narrower science domains can contribute their own tools for domain-specific experimentation, monitoring and diagnostics. This way small user communities could grow their own tools and support environments that meet their needs well. Testbed experts could support the domain-specific communities in order to help transition these tools and environments from research into production, and to maintain them in longer term. This would require building pathways for user-contributed software maturation and adoption, similar to other open science communities [5].

## 6 RELATED WORK

We are not aware of prior attempts to survey network testbed users about obstacles in their testbed interaction. However, related works exist in other fields that use shared infrastructure, such as robotics [10] and scientific experimentation [8, 17].

Manzoor et al. survey testbeds for ubiquitous robotics [10] and identify several ways of improving usability: having a GUI, a remotely accessible testbed, providing simulators and tools for experiment programming, logging and monitoring. Manzoor et al. further find that advanced testbeds implement basic functionalities that can be reused by users in experiments, thus allowing users to focus on higher-level, more relevant tasks.

Da Silva et al. report on a workshop on needs of the scientific workflows community [8]. Scientific workflows are different than testbed experimentation in a sense that they focus primarily on computation and data transfer, and that workflow managers offer good support for the entire experimentation lifecycle (running, analysis, sharing, etc.) Still, this report identifies several areas for improvement around usability: (1) FAIR workflows (Findable, Accessible, Interoperable, and Reusable), (2) training and education for users, (3) APIs and interoperability support for users to switch between workflow management systems and (4) building a community of users and developers. The first two usability findings align well with our findings around orientational, implementational and domain-specific support needs. The last two findings are also pertinent to testbeds, but questions around those issues were not included in our survey. We hope to explore these in the future.

Towns et al. [17] report on their extensive efforts to support scientific computing community through the XSEDE project. Efforts include extensive user training activities for many different categories of users (e.g., minority students, general student populations, faculty, etc.) as well as developer support in experiment design and optimization for individual users and teams. Similar activities would certainly be useful for network testbed users, but they require significant investment.

## 7 CONCLUSIONS

Network testbeds offer great opportunities for users to experiment on hardware and at scale that may not be available to them at their home institutions. But testbed interfaces and services can also present obstacles to research or learning. In this paper we reported on two studies we undertook to quantify obstacles in users' interaction with testbeds. Our findings offer a message of hope and appreciation for testbeds, and suggestions for further improvements. Our respondents were generally happy with their interaction with testbeds, and believed it helped them improve their research and learning. Testbed staff was regarded as helpful, and interaction with testbeds was mostly enjoyable.

We find that users overcome many obstacles that are present early in their interaction, as they become more familiar with the testbed environment. We also find that several obstacles merit deeper investigation and investment to address them. First, testbeds should develop tools that support the entire experiment lifecycle. Second, testbeds should create spaces for users that work in the same science domain to exchange their experiment designs, monitoring and diagnostic tools, and to learn from each others' experiences. There should also be a path to adopt popular user-developed tools into the testbed.

## ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation under Grant Nos. 1723717 and 1835608.

## REFERENCES

[1] Yoav Benjamini and Yosef Hochberg. 1995. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal statistical society: series B (Methodological)* 57, 1 (1995), 289–300.

[2] T. Benzel, R. Braden, D. Kim, C. Neuman, A. Joseph, K. Sklower, R. Ostrenga, and S. Schwab. 2006. Experiences With DETER: A Testbed for Security Research. In *2nd IEEE TridentCom.*

[3] Jack Cook, Richard Weiss, and Jens Mache. 2020. Refactoring a full stack web application to remove barriers for student developers and to add customization for instructors. *Journal of computing sciences in colleges* 36, 1 (2020).

[4] JFC De Winter and Dimitra Dodou. 2010. Five-point Likert items: t-test versus Mann-Whitney-Wilcoxon. *Practical Assessment, Research, and Evaluation* 15, 1 (2010), 11.

[5] Ewa Deelman, James Blythe, Yolanda Gil, Carl Kesselman, Scott Koranda, Albert Lazzarini, Gaurang Mehta, Maria Alessandra Papa, and Karan Vahi. 2003. Pegasus and the Pulsar Search: From Metadata to Execution on the Grid. In *Applications Grid Workshop, PPAM 2003.* http://pegasus.isi.edu/publications/ewa/agw_ppam2003.pdf

[6] DETER. [n.d.]. DeterLab Web page. http://www.deterlab.net.

[7] Dmitry Duplyakin, Robert Ricci, Aleksander Maricq, Gary Wong, Jonathon Duerig, Eric Eide, Leigh Stoller, Mike Hibler, David Johnson, Kirk Webb, Aditya Akella, Kuangching Wang, Glenn Ricart, Larry Landweber, Chip Elliott, Michael Zink, Emmanuel Cecchet, Snigdhaswin Kar, and Prabodh Mishra. 2019. The Design and Operation of CloudLab. In *Proceedings of the USENIX Annual Technical Conference (ATC).* 1–14. https://www.flux.utah.edu/paper/duplyakin-atc19

[8] Rafael Ferreira da Silva, Henri Casanova, Kyle Chard, Dan Laney, Dong Ahn, Shantenu Jha, Carole Goble, Lavanya Ramakrishnan, Luc Peterson, Bjoern Enders, Douglas Thain, Ilkay Altintas, Yadu Babuji, Rosa Badia, Vivien Bonazzi, Taina Coleman, Michael Crusoe, Ewa Deelman, Frank Di Natale, Paolo Di Tommaso, Thomas Fahringer, Rosa Filgueira, Grigori Fursin, Alex Ganose, Bjorn Gruning, Daniel S. Katz, Olga Kuchar, Ana Kupresanin, Bertram Ludascher, Ketan Maheshwari, Marta Mattoso, Kshitij Mehta, Todd Munson, Jonathan Ozik, Tom Peterka, Loic Pottier, Tim Randles, Stian Soiland-Reyes, Benjamin Tovar, Matteo Turilli, Thomas Uram, Karan Vahi, Michael Wilde, Matthew Wolf, and Justin Wozniak. 2021. Workflows Community Summit: Bringing the Scientific Workflows Community Together. https://doi.org/10.5281/zenodo.4606958

[9] Kate Keahey, Jason Anderson, Zhuo Zhen, Pierre Riteau, Paul Ruth, Dan Stanzione, Mert Cevik, Jacob Colleran, Haryadi S Gunawi, Cody Hammock, et al. 2020. Lessons learned from the Chameleon testbed. In *2020 {USENIX} Annual Technical Conference ({USENIX}{ATC} 20).* 219–233.

[10] Sarah Manzoor, Raza Ul Islam, Aayman Khalid, Abdul Samad, and Jamshed Iqbal. 2013. Testbeds for ubiquitous robotics: A survey. *Robotics and Autonomous Systems* 61 (2013), 12.

[11] Jelena Mirkovic, Aashray Aggarwal, David Weinmann, Paul Lepe, Jens Mache, and Richard Weiss. 2020. Using Terminal Histories to Monitor Student Progress on Hands-on Exercises. In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education, SIGCSE 2020, Portland, OR, USA, March 11-14, 2020.* ACM, 866–872.

[12] Jelena Mirkovic and Peter A. H. Peterson. 2014. Class Capture-the-Flag Exercises. In *2014 USENIX Summit on Gaming, Games, and Gamification in Security Education (3GSE 14).* USENIX Association, San Diego, CA. https://www.usenix.org/conference/3gse14/summit-program/presentation/mirkovic

[13] Jelena Mirkovic, Mike Ryan, John Hickey, Keith Sklower, Peter Reiher, Peter A. H. Peterson, B. Hoon Kang, Mooi Choo Chuah, Daniel Massey, and Gisele Ragusa. 2011. Teaching Security with Network Testbeds. In *ACM Sigcomm Educational Workshop.*

[14] Jelena Mirkovic, Aimee Tabor, Simon Woo, and Portia Pusey. 2015. Engaging Novices in Cybersecurity Competitions: A Vision and Lessons Learned at ACM Tapia 2015. In *2015 USENIX Summit on Gaming, Games, and Gamification in Security Education (3GSE 15).* USENIX Association, Washington, D.C. https://www.usenix.org/conference/3gse15/summit-program/presentation/mirkovic

[15] COSMOS Project. [n.d.]. COSMOS Education Toolkit. https://www.cosmos-lab.org/cosmos-toolkit/.

[16] Dipankar Raychaudhuri, Ivan Seskar, Gil Zussman, Thanasis Korakis, Dan Kilper, Tingjun Chen, Jakub Kolodziejski, Michael Sherman, Zoran Kostic, Xiaoxiong Gu, Harish Krishnaswamy, Sumit Maheshwari, Panagiotis Skrimponis, and Craig Gutterman. 2020. Challenge: COSMOS: A City-Scale Programmable Testbed for Experimentation with Advanced Wireless. In *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking (MobiCom '20).* Association for Computing Machinery, New York, NY, USA, Article 14, 13 pages. https://doi.org/10.1145/3372224.3380891

[17] John Towns, Timothy Cockerill, Maytal Dahan, Ian Foster, Kelly Gaither, Andrew Grimshaw, Victor Hazlewood, Scott Lathrop, Dave Lifka, Gregory D. Peterson, Ralph Roskies, J. Ray Scott, and Nancy Wilkins-Diehr. 2014. XSEDE: Accelerating Scientific Discovery. *Computing in Science Engineering* 16, 5 (2014), 62–74. https://doi.org/10.1109/MCSE.2014.80

[18] Richard Weiss, Jens Mache, Elizabeth Hawthorne, Ambareen Siraj, Blair Taylor, Siddharth Kaza, and Ankur Chattopadhyay. 2021. Integrating Hands-on Cybersecurity Exercises into the Curriculum in 2021. In *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education.* 1358–1358.

## 8 APPENDIX

### 8.1 Survey Questions

These are the questions we asked in our online survey:

- **Background questions**

(1) What is your age?
  (a) 18-24
  (b) 25-34
  (c) 35-44
  (d) 45 and older
  (e) Prefer not to say
(2) Which gender do you identify as?
  (a) Male
  (b) Female
  (c) Other
(3) List all testbeds that you have used (e.g., Cloudlab, Deterlab, GENI, etc.) If you don't recall the name of the testbed please give us anything that you can remember, that may help us identify the testbed.
(4) In which scenario did you use a testbed or testbeds?
  (a) Only for my research
  (b) Only for my class
  (c) First for research then for my class
  (d) First for class then for my research
(5) Please rate your knowledge of Linux prior to using any testbed (on a scale from 1 - novice, I had no exposure to 5 - expert, I helped others learn Linux)
(6) Please rate your knowledge of SSH prior to using any testbed (on a scale from 1 - novice, I had no exposure to 5 - expert, I helped others learn SSH)

- **Testbed experience**

(7) What year was the FIRST time you used a testbed? (approximate is fine)
(8) Please think back to your EARLY experience with the testbed or testbeds when answering the following questions (rating on a scale Very difficult, Difficult, OK, Easy and Very Easy):
  (a) Creating an account on the testbed was (**account create**)
  (b) Accessing testbed's Web interface was (**UI access**)
  (c) Creating an experiment was (**exp create**)
  (d) Diagnosing problems with experiment creation was (**exp create diag**)
  (e) Transferring files to and from my experiment was (**file trans into exp**)
  (f) Finding out how to SSH into my experiment was (**how to SSH**)
  (g) Running SSH to log into my experiment nodes was (**run SSH**)
  (h) Designing my experiment was (**exp design**)
  (i) Setting up my experiment nodes (e.g., installing software) was (**exp setup**)

(j) Running my experiment was (**exp run**)
(k) Doing SSH forwarding on my experiment was (**SSH fwd**)
(l) Transferring files between experiment nodes was (**file trans btw nodes**)
(m) Monitoring my experiment and detecting failures was (**exp monitor**)
(n) Diagnosing and mitigating failures in my experiment was (**exp diag**)

(9) What year was the LAST time you used a testbed? (approximate date is fine)

(10) Please think back to your LATEST experience with the testbed or testbeds when answering the following questions (rating on a scale Very difficult, Difficult, OK, Easy and Very Easy):
(a) Accessing testbed's Web interface is now (**UI access**)
(b) Creating an experiment is now (**exp create**)
(c) Diagnosing problems with experiment creation is now (**exp create diag**)
(d) Transferring files to and from my experiment is now (**file trans into exp**)
(e) Finding out how to SSH into my experiment is now (**how to SSH**)
(f) Running SSH to log into my experiment nodes is now (**run SSH**)
(g) Designing my experiment is now (**exp design**)
(h) Setting up my experiment nodes (e.g., installing software) is now (**exp setup**)
(i) Running my experiment is now (**exp run**)
(j) Doing SSH forwarding on my experiment is now (**SSH fwd**)
(k) Transferring files between experiment nodes is now (**file trans btw nodes**)
(l) Monitoring my experiment and detecting failures is now (**exp monitor**)
(m) Diagnosing and mitigating failures in my experiment is now (**exp diag**)

(11) Which of the following obstacles did you experience in your EARLY testbed interactions (check all that apply)?
(a) I couldn't find out how to create experiments
(b) I couldn't find out how to SSH into my nodes
(c) I had trouble with two-hop login (SSHing into a gateway node and then into an experimental node)
(d) I had trouble setting up passwordless SSH (when you don't have to type in your password to log in)
(e) I had trouble transferring files between my computer and experiment nodes
(f) I had trouble starting an experiment (allocating machines)
(g) I had trouble finding out node names and/or IP addresses
(h) I didn't quite know how to use SSH and/or scp
(i) I had trouble installing SSH/scp on my computer
(j) I had trouble transferring files between experiment nodes
(k) I had trouble understanding shared directories on the tested
(l) When I had to use multiple nodes in an experiment it was dificult to remember which node I was on during SSH session
(m) When I returned to my experiment I had trouble restoring experiment sate
(n) Testbed staff was slow to reply to my requests for help
(o) Tested staff was uninterested in helping me
(p) Testbed staff was rude in their interaction with me
(q) Testbed staff was helpful

- **Recommendations**

(12) What do you think would help testbed users? (on a scale Yes, definitely, Maybe, Probably not and No opinion)
(a) More tutorials and documentation
(b) Better organized tutorials and documentation
(c) Video tutorials
(d) Tools that let users interact with their experiment nodes via a GUI instead of SSH
(e) Tools that aid experiment design
(f) Sample experiments that one could change to fit their research goal
(g) Linux tutorials
(h) A detailed explanation how the testbed works
(i) A better organized Web UI
(j) Tools that let users design, run and monitor their experiment from a graphical interface
(k) Staff available interactively (e.g., via chat)
(l) Training workshops
(m) Better error messages that point users towards a solution
(n) Ability to save experiment state when machines are reclaimed and restore it later
(o) A log capturing a user's interaction with the experiment

(13) Please use this space to comment on any other improvements that testbeds could make to help users

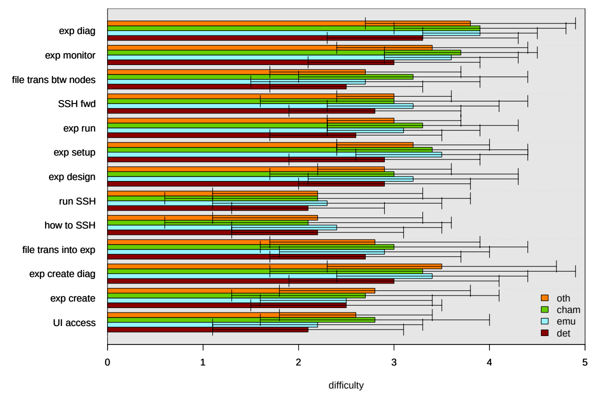
(14) Overall how would you rate your experience with testbeds (on a scale from 1 - Very frustrating to 5 - Very enjoyable)

(15) Overall how much did testbeds help your learning or research (on a scale from 1 - Not at all to 5 - They helped a lot)

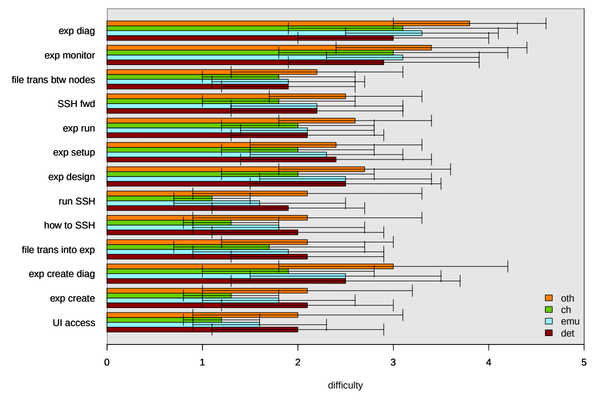
## 8.2 Comparison among testbeds

We grouped users per testbed they used (Deterlab, Emulab, Chameleon or other) and compared the ratings for both early and recent experiences between those groups of users. This comparison is imperfect, since there is a large difference in the group size. Also, many users use multiple testbeds and will appear in multiple groups – thus their experiences cannot be attributed to a single testbed. Figure 7 shows means (bars) and standard deviations (error bars) of ratings per experience and per testbed. There was no significant difference in experiences among users of different testbeds.

To address the group size issue, we further separated those users that have only used Deterlab into one group (37 users), and users that have not used Deterlab into another group (16 users). Figure 8 shows means (bars) and standard deviations (error bars) of ratings per experience and per group. There was no significant difference in experiences among users in these two groups.

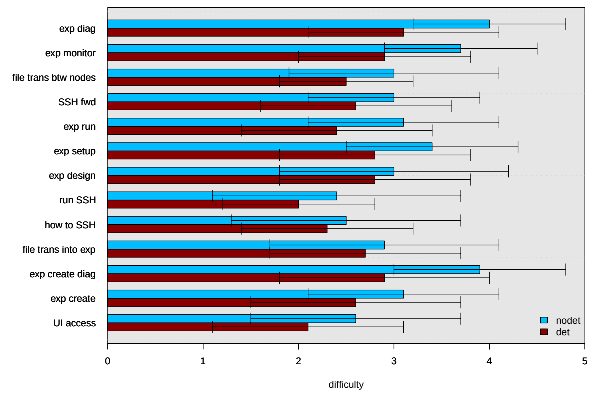


(a) Early usage experiences

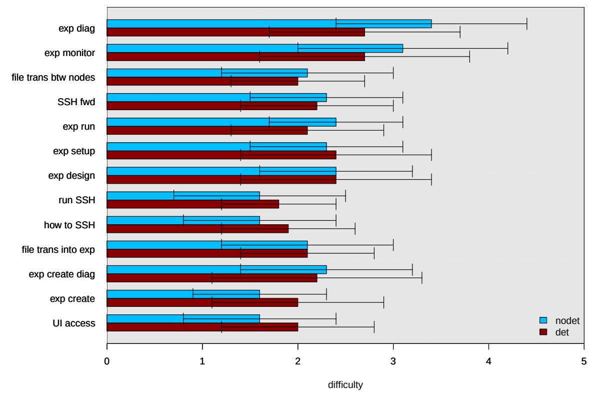


(b) Later usage experiences

**Figure 7: User experiences collected in our online survey, regarding their early and recent testbed usage, disaggregated per testbed. Many users use multiple testbeds and will appear in multiple groups.**



(a) Early usage experiences



(b) Later usage experiences

**Figure 8: User experiences collected in our online survey, regarding their early and recent testbed usage, disaggregated with regard to their use of Deterlab. Users that used both Deterlab and another testbed were excluded.**