

Evaluation von VIRL, GNS3 und Mininet als Virtual Network Testbeds in der Hochschullehre

Christoph Seifert,¹ Sven Reißmann,² Sebastian Rieger,¹ Christian Pape¹

Abstract: In den letzten Jahren wurden verschiedene Frameworks zur Implementierung virtueller Computernetzwerkumgebungen entwickelt. Typischerweise können diese in Simulations- und Emulationsansätze kategorisiert werden. Letztere zeichnen sich dabei häufig durch eine im Vergleich zu Simulationsansätzen höhere Praxisnähe und einen realitätsnahen Funktionsumfang aus. Die Einrichtung und Wartung dieser Tools, z.B. in Laborumgebungen von Universitäten, ist jedoch komplex. In diesem Papier werden moderne Netzwerk-Emulations-Tools für Computer-Networking-Kurse und Forschungsprojekte im Netzwerklabor (NetLab) der Hochschule Fulda evaluiert. In erster Linie werden GNS3, VIRL und Mininet ausgewertet, die derzeit in Lehrveranstaltungen in Bachelor- und Masterstudiengängen im NetLab eingesetzt werden. Besonderes Augenmerk wird auf die Skalierbarkeit für große Studierendenzahlen, geringen administrativen Aufwand und Kosten, hohe Praxisnähe der unterstützten Laborübungen und entsprechende didaktische Anforderungen gelegt.

Keywords: Netzvirtualisierung, Hochschullehre, VIRL, GNS3, Mininet

1 Einleitung

Die Paradigmen im Bereich der Lehre haben sich im Verlauf der vergangenen Jahre stark verändert. E-Learning und Blended Learning erlauben zeit- und ortsungebundenes Lernen und relativieren teils sogar die Erfordernis der Präsenz von Studierenden in Vorlesungen. Lehrveranstaltungen in Laboren im Allgemeinen, insbesondere in Netzwerklaboren, lassen sich jedoch häufig nicht ohne weiteres auf diese neuartigen Lehrmethoden übertragen. Beispielsweise ist für Trainings an physischen Geräten oder zur Stärkung von Teamwork häufig eine körperliche Präsenz der Studierenden in Praktika sinnvoll oder gar notwendig. Während Vorlesungen in der Regel 90 Minuten dauern, erfordern Laborveranstaltungen zusätzliche Vor- und Nachbereitung. Zum Beispiel müssen in Netzwerklaboren für Übungen zu Beginn jeder Veranstaltung die erforderlichen Hard- und Software-Umgebungen aufgebaut und konfiguriert werden. Dies bedeutet insbesondere bei steigenden Teilnehmerzahlen in Laborkursen einen erhöhten Aufwand. So bleiben von den regulären 90 Minuten häufig nur 60 Minuten für die Durchführung der eigentlichen Praktika.

Virtuelle Labore können dabei helfen eben diese Probleme zu adressieren, indem sie einerseits schnell auf einen definierten Zustand vorbereitet werden können und andererseits

¹ Hochschule Fulda, Angewandte Informatik, [vorname.nachname]@informatik.hs-fulda.de

² Hochschule Fulda, Rechenzentrum, sven.reissmann@rz.hs-fulda.de

erlauben, den aktuellen Zustand zu speichern, um später genau an diesem Punkt fortzufahren. Virtuelle Labore können durch Simulatoren oder Emulatoren realisiert werden. Da die Simulation jedoch oft nicht vollständig dem Verhalten von realen Geräten entspricht, wie [Ha12] und [YM17] zeigen, bieten Emulatoren häufig die bessere Wahl. Abbildung 1 zeigt einige Charakteristiken von Emulatoren, Simulatoren und Testbeds.

	Simulatoren	Testbeds		Emulatoren
		Shared	Custom	
Realitätsnahe Funktionen		✓	✓	✓
Realistisches Timing	✓	✓	✓	(eingeschränkt)
Realistischer Traffic		✓	✓	✓
Flexibilität der Topologien	✓	(eingeschränkt)		✓
Einfache Nachvollziehbarkeit	✓	✓		✓
Geringe Kosten	✓			✓

Abb. 1: Lösungen für reproduzierbare Netzexperimente aus [Ha12].

Im Netzwerklabor (NetLab) des Fachbereichs Angewandte Informatik der Hochschule Fulda werden praktische Übungen für die Lehrinhalte des Cisco Networking Academy Programms mit dem Simulationstool Packet Tracer [18g] unterstützt. Diese Software kann für die im Rahmen dieses Programms angebotenen CCNA-Kurse alle erforderlichen Aspekte und Funktionen virtuell abbilden. Packet Tracer erlaubt es jedoch nicht Funktionalitäten zu implementieren, die für fortgeschrittenere CCNA-Sicherheits- oder CCNP-Kurse erforderlich sind. Darüber hinaus ist die Praxisnähe der Software aufgrund fehlender Funktionen (z.B. Anbindung der Simulation an das Internet) und fehlendem realitätsnahem Netzwerk-Traffic bedingt durch den Simulationsansatz eingeschränkt. Aus diesen Gründen sind sowohl Packet Tracer als auch andere Simulatoren kein Bestandteil der Analyse in diesem Paper. Die meisten Emulatoren bzw. virtuellen Netzwerk-Umgebungen lassen sich auf Plattformen wie Linux, Windows und auch Mac OS realisieren. Auch eine Bereitstellung auf virtuellen Plattformen, z.B. VMware-basierte Server-Lösungen, wird unterstützt. In dieser Ausarbeitung konzentrieren wir uns auf Werkzeuge, die die meisten der im Abschnitt 3 genannten Kriterien für den Einsatz im Umfeld des NetLab erfüllen. Der Schwerpunkt wird hierbei auf das Virtual Internet Routing Lab (VIRL)[18i] von Cisco sowie die Open Source Lösungen GNS3 [18d] und Mininet [18e] gelegt, die alle im NetLab für unterschiedliche Einsatzzwecke aktiv eingesetzt werden. VIRL ist mittlerweile nur noch als Einzelplatzlizenz nutzbar. Die auf der gleichen Basis aufsetzende Mehrplatzlösung Cisco Modeling Lab (CML) [18a] wird aufgrund ihrer hohen Lizenzkosten nur der Vollständigkeit halber erwähnt.

2 Verwandte Arbeiten

Moderne IT-Infrastrukturen basieren auf komplexen Netztopologien, die die Grundlage für Skalierbarkeit, Redundanz und Hochverfügbarkeit von IT-Services bilden. Es ist

offensichtlich, dass die Vermittlung theoretischen Wissens alleine nicht genug ist, um Studierenden der Informatik eine gute Grundlage zum Verständnis solcher Topologien zu bieten. Daher sind die Simulation und Emulation von Netzwerktopologien an Hochschulen Gegenstand aktueller Forschung und kontinuierlicher Weiterentwicklung. Ein Vergleich einer rein VIRT-basierten Lernumgebung mit physischen Setups (CCNA-Pods) und der Simulation mit Cisco Packet Tracer wird in [Ta16] vorgestellt. In [Ob14] wird der Einsatz von VIRT für Forschung und Lehre skizziert und bewertet. Die früher im NetLab vorrangig eingesetzte VIRT-Umgebung und deren Skalierbarkeit wurden bereits in [Ri17] und [SRP17] vorgestellt. Ein auf einer deklarativen XML-basierten Modellierungssprache basierendes Framework, das erweiterbare und skalierbare Emulationen/Simulationen von großen Netzwerktopologien realisiert, ist in [MK16] dargestellt. Ein Vergleich mit anderen bekannten Simulations- und Emulationswerkzeugen, z.B. ns-3 oder PlanetLab ist ebenfalls enthalten. Netzwerktopologien mithilfe virtueller Router auf Linux-Computern wurden z.B. in [Ba03] emuliert. Der Echtzeit-Netzwerkemulator EmuLab, der für das Testen von Protokollen und des Anwendungsverhaltens in Netzwerktopologien verwendet wird, ist in [KE04] näher beschrieben. [PR16] zeigt einen didaktisch ausgerichteten Ansatz, der darauf abzielt, reale Netzwerkgeräte mit geringen Kosten und geringem Aufwand zu modellieren und zu emulieren. [Ha12] stellt ein Rahmenwerk für eine reproduzierbare container-basierte Emulation von netzbezogenen Experimenten einschließlich Software Defined Networking (SDN) vor. Reproduzierbare Experimente und Forschungsergebnisse im Netzwerkbereich werden zudem in [YM17], [F117] und [Ba17] thematisiert.

3 Anforderungen und Auswahlkriterien

Für die Durchführung von Laborübungen mit dem Fokus auf Netzwerktechnologien und -protokolle wurden verschiedene Optionen evaluiert. Diese können, wie in Abschnitt 1 erläutert, typischerweise in Simulatoren, physische oder virtuelle Testbeds und Emulatoren kategorisiert werden. Abbildung 2 veranschaulicht die Einordnung dieser Optionen zwischen Praxis und Theorie. Die im Bild gezeigten Gradienten ergeben sich aus Erfahrungen mit verschiedenen Werkzeugen für physische und virtuelle Laborumgebungen, die im NetLab gewonnen wurden. Physische Testbeds verwenden echte Netzwerkgeräte um Testnetzwerke einzurichten, die entweder isoliert oder in das Netzwerk des Labors integriert und an das Internet angebunden sind. Sie wurden für den Bachelor-Studiengang und das CCNA-Programm, das parallel zum Curriculum angeboten wird, eingesetzt. Um die Flexibilität des Setups zu erhöhen und die Vorbereitungszeit zu reduzieren, wurden virtuelle Testbeds unter Verwendung von VMware Workstation und Arista vEOS verwendet, die jedoch individuell erstellt und bereitgestellt werden mussten.

Emulatoren wie GNS3 [18d], EVE-NG [18b] / UNetLabv2 [18h], VIRT [18i], CML [18a], Mininet [18e] oder eNSP [18c] lassen sich zur automatisierten Erstellung virtueller Netze und Bereitstellung der benötigten Konfigurationen nutzen. Sie ermöglichen dadurch eine signifikante Reduzierung der Vorbereitungszeit. Eine weitere Abstraktion von realen

Netzwerken führt zu Simulatoren (z.B. Packet Tracer [18g], ns-3 [18f]), die zwar die Realität weniger exakt abbilden, dafür aber einen signifikant geringeren Ressourcenbedarf aufweisen, wodurch auch die Realisierung sehr großer Topologien ermöglicht wird. Theoretische Modelle bieten eine hohe Flexibilität, erfüllen aber nicht die Anforderungen bezüglich der Praxisnähe, die im Labor begleitend zur Vorlesung erreicht werden soll.

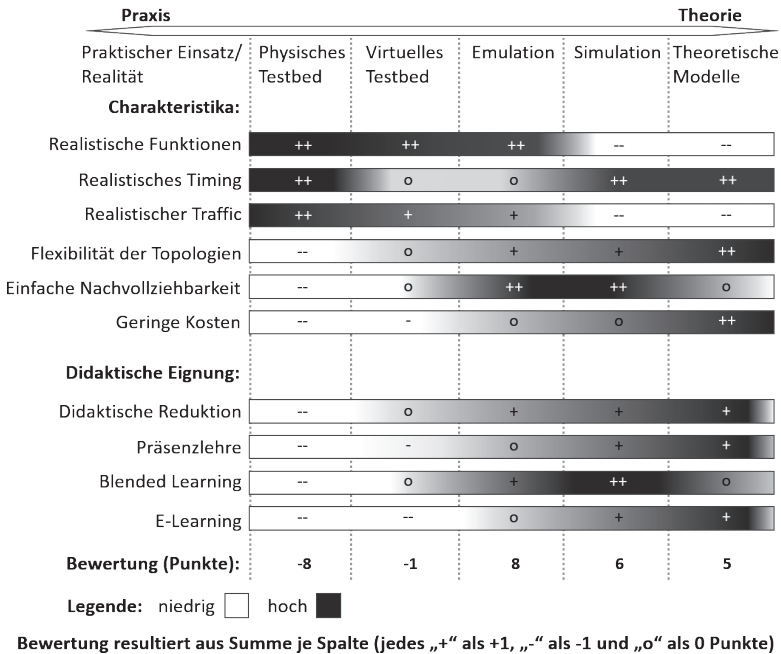


Abb. 2: Klassifizierung unterschiedlicher Ansätze für experimentelle Netzwerkumgebungen.

Zusätzlich zu den in Abbildung 1 aufgeführten Charakteristiken gemäß [Ha12] wurden didaktische Anforderungen an die verschiedenen Ansätze evaluiert, die sich auf die Qualität der Lehrveranstaltungen im NetLab auswirken. Testumgebungen weisen meist komplexe Ansätze auf, die bereits bei der Erstellung des Ausgangszustands eines Versuchs zur Überforderung führen. Durch didaktische Reduktion der Versuchsdurchführung soll der Fokus auf die wesentlichen Lernerfahrungen ausgerichtet werden. Darüber hinaus wurde die Eignung der einzelnen Ansätze für traditionelle und für moderne Lehrmethoden wie E-Learning- und Blended-Learning überprüft. E-Learning und Blended-Learning erfordern einen zeitlich und örtlich unabhängigen Zugang zur Testumgebung und damit gleichzeitig die Möglichkeit, die experimentelle Umgebung leicht und ohne direkte Anleitung verstehen und modifizieren zu können. Integrierte Umgebungen wie sie von Simulatoren und Emulatoren zur Verfügung gestellt werden, besitzen damit eine hohe Eignung für die Realisierung virtueller Netzwerkumgebungen. Sie bieten eine hohe Praxisnähe, realistischen Funktionsumfang, ermöglichen das Pausieren und Abspeichern des aktuellen Zustands

und reduzieren den Administrationsaufwand erheblich. Die in Abbildung 2 dargestellte Bewertung ergibt sich aus Erfahrungen basierend auf regelmäßigen Laborveranstaltungen im NetLab mit einer typischen Dauer von 90 Minuten. Vor allem wegen des manuellen Setups, der damit verbundenen Vorbereitungszeit und des Wartungsaufwands erhalten virtuelle Testbeds -1 Punkt. Folglich erhalten physische Testbeds, die zwar noch näher an der Realität liegen, aber entsprechend aufwändiger zu konfigurieren und ungeeignet für E-Learning sind, insgesamt -8 Punkte. Solche Testbeds werden im NetLab zwar ebenfalls aktiv eingesetzt, um komplexe Szenarien abzubilden, die besonders hohe Praxisnähe erfordern, jedoch bieten andere Implementierungsmöglichkeiten eine höhere Flexibilität und Effizienz. Besonderes Augenmerk wird auf den Einsatz von Emulatoren gelegt, die gewissermaßen einen Mittelweg bieten, um die Vorteile aus Theorie und Praxis zu vereinen. Sie stellen jedoch neben den genannten Vorteilen hohe Anforderungen an die benötigten Ressourcen, da typischerweise Betriebssysteme bzw. Images von realer Netzwerk-Hardware in Form von virtuellen Maschinen mit hohem Arbeitsspeicherbedarf ausgeführt werden. Die mögliche Größe der emulierten Netze verhält sich somit - im Gegensatz zu Simulatoren - linear zu den verfügbaren Rechen- und Speicherressourcen.

Zur Bewertung der jeweiligen Lösungen wurden nachfolgend beschriebenen Kriterien herangezogen (vgl. Abbildung 3). Einen wichtigen Auswahlfaktor für den Betrieb im NetLab stellen die Lizenzkosten dar, die mit einer entsprechend hohen Gewichtung in die Bewertung eingehen. Kriterien wie ein zentrales Management der Umgebung, hohe Kompatibilität (z.B. bei der Verwendung verschiedener Betriebssysteme als Hosts für die Emulation) und die Zugangsmöglichkeiten im Hinblick auf E-Learning wurden mit etwas geringerem Gewicht in die Bewertung einbezogen. Mit einer etwas höheren Wertung wurde die Möglichkeit der Integration von Images realer Netzwerkgeräte einbezogen, um eine hohe Praxisnähe von experimentellen Labor-Umgebungen zu ermöglichen. Dies ist im NetLab aufgrund der praktischen Verwendung in Cisco-basierten Kurse (z.B. CCNA, CCNP), aber auch um eine hohe Realitätsnähe durch Verwendung echter Switch- und Router-Betriebssysteme verschiedener Hersteller zu erreichen, ein wichtiges Kriterium. Weitere Kriterien stellen die Ressourcenanforderungen und Skalierbarkeit der Lösung dar, die bei typischerweise 20 Teilnehmern in den Kursen die Vor- und Nachbereitungszeit von Laborveranstaltungen sowie die Performance innerhalb der Testumgebung maßgeblich bestimmen. Bei einem zentralen Betrieb der Emulationslösung auf Servern des NetLab, ist eine automatisierte Lastverteilung der von Studierenden gestarteten virtuellen Ressourcen und Testbeds (z.B. über einen verteilten Cluster) wünschenswert. Eine Reihe technischer Auswahlkriterien mit direktem Einfluss auf die didaktische Flexibilität der Lösungen wurde ebenfalls in die Bewertung einbezogen. Dazu gehört zunächst die Anbindung der Umgebung an ein reales Netz wie das physische Labornetz im NetLab und das Internet, z.B. um die Kursteilnehmer in die Lage zu versetzen, reale Werkzeuge (z.B. arp, ping, traceroute, Wireshark) in ihren Experimenten wie gewohnt und mit realen Zielen und realistischem Datenverkehr verwenden zu können. Gleichzeitig spielt die Anbindung an das Internet eine wichtige Rolle um den Zugang zu den emulierten Netzwerken über VPN zu ermöglichen und damit die Anforderungen im Bezug auf E-Learning zu erfüllen.

Kriterium	Gewicht 1-7	mininet		VIRL		CML		GNS3 (V2.1)		
		Bemerkung	#	Bemerkung	#	Bemerkung	#	Bemerkung	#	
Administrativer Aufwand										
- Lizenz (Kosten)	7	Frei (Open Source)	10	199€ für 20 Cisco Nodes p.a. (nicht skalierbar, Einzelplatz-Lizenz)	4	Kostenintensiv (> \$2.500 p.a. (für 15+10 Nodes)), skalierbar (zus. Kosten)	0	Frei (Open Source, GPLv3)	10	
- Zentrales Management	3	Isoliert (Einzelplatz), aber per Skript automatisierbar	7	Cluster mit mehreren Compute Nodes (OpenStack)	10	Cluster mit mehreren Compute Nodes (OpenStack)	10	Mehrere aber nicht zentral verwaltete Server möglich	3	
- Kompatibilität und Zugänglichkeit	2	verfügbar für Linux (Win, Mac als VM)	7	VMMaestro Client und Server VMs für Linux, Win, Mac, vSphere	8	VMMaestro Client und Server VMs für Linux, Win, Mac, vSphere	8	Verfügbar für Linux, Win, Mac, auch als vorkonfigurierte VM	7	
- Custom Images für Netzhardware	4	Kein Support für Images realer Netzhardware	0	Erweiterbar (nur für Cisco Nodes beschränkt), Third-Party Images verfügbar	5	Erweiterbar (nur für Cisco Nodes beschränkt), Third-Party Images verfügbar	5	Images für Vielzahl Router, Switches (Cisco Images nicht enthalten)	7	
- Load Balancing	2	Nur manuell	5	Ja	10	Ja	10	Nur manuell	5	
- Erforderliche Ressourcen	1	Gering (LXC Container)	10	Device-abhängig	6	Device-abhängig	6	Device-abhängig	8	
Didaktische Anforderungen										
- Anbindung an reales Netz	4	Ja	8	Ja	8	Ja	8	Ja (NAT über GUI, Bridge manuell)	6	
- Mehrere Benutzer und Sitzungen	5	Nur manuell	5	Nein (offiziell nur single user)	8	Ja	10	Ein Projekt pro User, aber Zugriff mehrerer Clients möglich	6	
- Konsolenverbindung	2	Beschränkte LXC Konsole	6	Jeder Benutzer hat eigene Konsole	6	Jeder Benutzer hat eigene Konsole	6	Gemeinsame Konsole (User)	10	
- QoS Emulation auf Links	1	Ja	10	Ja (nur Delay, Jitter, Loss)	8	Ja (nur Delay, Jitter, Loss)	8	Ja (nur Delay, Jitter, Loss)	8	
- VPN Zugang zu virtuellen Netzen	1	Manuell einrichtbar	5	Ja	10	Ja	10	über VPN Server in virtuellem Netz	5	
Gewichtete Bewertung			209			222			204	226

Abb. 3: Evaluierung im NetLab eingesetzter Netz-Emulatoren basierend auf Erfahrungswerten.

Studierende können sich so mit ihrem Rechner direkt in die von ihnen gestarteten virtuellen Testnetze verbinden. Um die Praxisnähe von virtuellen Netzen zusätzlich zu steigern, kann die Emulation von QoS-Metriken (z.B. Packet Loss, Delay, Jitter) auf Verbindungen in den virtuellen Netzen sinnvoll sein. Die Eignung der jeweiligen Lösung für diese Forderung spiegelt sich im Bewertungskriterium *QoS Emulation auf Links* wieder. Schließlich wurden auch Möglichkeiten der Kollaboration in Bezug auf die didaktischen Anforderungen

bewertet. Diese sind in unserem Setup essentiell, da sie die Arbeit in Teams aus mehreren Studierenden ermöglichen, was etwa die gleichzeitige Verbindung mehrerer Benutzer zur Konsole emulierter Netzwerk-Komponenten erfordert.

4 Evaluation geeigneter Virtual Network Testbeds

Als geeignete Emulatoren haben sich im Umfeld des NetLab vor allem Cisco VIRL, GNS3 und Mininet erwiesen. Die im vorherigen Abschnitt genannten Auswahlkriterien wurden von eins für ein niedriges bis sieben für ein hohes Ranking gewichtet. Die Merkmale für die Werkzeuge wurden basierend auf Erfahrungswerten aus dem Einsatz im NetLab in Werten zwischen null (am schlechtesten) und zehn (am besten) evaluiert. Gewicht und Wert werden für jedes Kriterium multipliziert und aufsummiert, um die gewichtete Gesamtpunktzahl für jedes Werkzeug zu erhalten. Das in Abbildung 3 aufgeführte Werkzeug Cisco Modeling Labs (CML) ist lediglich der Vollständigkeit halber enthalten, da CML für den Betrieb von Cisco VIRL als Mehrplatzvariante erforderlich ist. CML und VIRL nutzen die gleiche technische Basis, jedoch ist CML wegen seines Verkaufsmodells sehr kostenintensiv und wird deshalb im NetLab nicht eingesetzt. Auch andere Emulatoren wie EVE-NG [18b] bzw. UNetLabv2 [18h] wurden noch nicht einbezogen, da sie sich zum Zeitpunkt unserer Tests noch in einer frühen Entwicklungsphase befanden. Ebenfalls wurde eNSP [18c] bislang nur testweise im NetLab eingesetzt, da es zwar eine Vielzahl der gestellten Anforderungen erfüllt, jedoch auf die Bereitstellung von Huawei-Komponenten beschränkt ist.

Zum Einsatz kommen im NetLab derzeit Cisco VIRL [18i] und GNS3, die in einer VMware vSphere 6.5 Infrastruktur betrieben werden. Eine VIRL Lizenz kostet derzeit 199 € pro Jahr und erlaubt die Virtualisierung von bis zu 20 virtuellen Cisco Geräten in emulierten Topologien. Virtuelle Geräte anderer Hersteller (z.B. Arista vEOS, Juniper JunOS, GNU/Linux) können ohne Limitierung hinzugefügt werden. Bei einer Clusterbasierten Installation können die Topologien auf mehreren Compute Nodes implementiert werden, wobei bei VIRL OpenStack für eine gleichmäßige Lastverteilung gestarteter Topologien über mehrere verfügbare VIRL Hosts sorgt. Einen wesentlichen Nachteil von VIRL stellt jedoch die geänderte Strategie von Cisco in Bezug auf den Wegfall der Academic License dar, wodurch die Kosten für die Lizenzierung erheblich steigen (vgl. [18j]). Obwohl die Cisco-eigenen Tools wie Packet Tracer, VIRL oder CML am besten geeignet sind, unsere Cisco-basierten Kurse und Laborübungen zu unterstützen, haben diese Tools derzeit große Probleme in Bezug auf den Funktionsrealismus (bei Packet Tracer), bzw. den Wartungsaufwand oder die hohen Lizenzkosten (bei VIRL und CML).

Als Alternative zu VIRL erfüllt GNS3 [18d] seit der Version 2.1.0 derzeit die meisten Anforderungen für die Erstellung virtueller Testnetzwerke im Bereich des NetLab. Zudem bietet es die größte Unterstützung für virtuelle Maschinen, Images und Network Operating Systems (NOS) unterschiedlichster Netz-Komponenten und Hersteller. Verwendbare Cisco-Images für GNS3 können aus lizenztechnischen Gründen allerdings nur von VIRL in

GNS3 importiert oder über ein Cisco CCO-Konto, z.B. durch Cisco Networking Academy-Mitglieder, in GNS3 bereitgestellt werden. Bis auf wenige Einschränkungen können mit GNS3 die in Abschnitt 3 definierten Anforderungen des NetLab umgesetzt werden. Der Ressourcen-Verbrauch ist deutlich geringer als bei VIRL, was sich in größeren realisierbaren virtuellen Netztopologien sowie einer geringeren Rüstzeit bemerkbar macht. Hierfür wurden Messungen im Vergleich zur in [Ri17] beschriebenen VIRL-Umgebung nach dem gleichen dort vorgestellten Verfahren durchgeführt. Für die Messung wurden fünf Instanzen der in [Ri17] beschriebenen Netztopologie (bestehend aus vier Arista vEOS Nodes) parallel jeweils in GNS3 und VIRL gestartet. Die Bereitstellungszeit für einen einzelnen VIRL Host lag in [Ri17] bei 315,2 Sekunden. Mit GNS3 konnte bei identischer Konfiguration auf einem einzelnen Host für die gleichen Topologien eine Bereitstellungszeit von von 173,5 Sekunden (55% der von VIRL benötigten Zeit) erzielt werden. Die aktuell in GNS3 noch fehlende Lastverteilung (Cluster-Lösung) im Vergleich zu VIRL könnte zukünftig durch die Entwicklung geeigneter Erweiterungen umgesetzt werden. Im Umfeld des NetLab sind hierzu Projekte und Abschlussarbeiten als Erweiterungen für GNS3 (Web-Interface basierend auf der GNS3 RESTful API) geplant.

Die in Abbildung 3 genannte Open-Source-Lösung Mininet [18e] findet im NetLab insbesondere für Abschlussarbeiten im Bereich SDN und NFV sowie für Master-Lehrveranstaltungen Einsatz. Durch die Verwendung von LXC Containern und OpenVSwitch als Basis für virtuelle Hosts und Netze erlaubt Mininet das Starten von großen Netztopologien auch auf Einzelplatzrechnern. Eine Skalierung von Mininet über mehrere Compute Nodes (vgl. Load Balancing) wird nicht unterstützt. Die Python-API von Mininet ermöglicht zusätzlich Experimente zur Reproduzierbarkeit von aktuellen Forschungsergebnissen im Netzwerkbereich [YM17]. Dies umfasst auch die Emulation von QoS-Eigenschaften von Links (basierend auf NetEm im Linux Kernel). Für realitätsnahe Experimente steht die komplette Vielfalt von Linux-Werkzeugen auf der Command Line zur Verfügung. Auch die Anbindung an reale Netze bzw. das Internet ist möglich. Mininet unterstützt allerdings keine Bereitstellung von Images realer Netz-Hardware im Vergleich zu den anderen genannten Lösungen. Zusätzlich ist aufgrund der Realisierung von Mininet als Kommandozeilenwerkzeug und der daher fehlenden GUI die Kollaboration von Studierenden sowie die Einarbeitungszeit in Laborveranstaltungen schlechter als bei den betrachteten Alternativen GNS3 und VIRL. Aufgrund seines leichtgewichtigen Ansatzes für große Topologien und die hervorragende Eignung als SDN-Umgebung bleibt Mininet insb. für forschungsnahe Experimente im Master-Bereich trotz geringerer Bewertung im Vergleich zu GNS3 und VIRL aktuell die beste Wahl.

5 Fazit

Der Einsatz von virtuellen Netzwerk-Testbeds ermöglicht für das NetLab in verschiedenen Veranstaltungen eine gute Balance zwischen Praxis und Theorie. Durch den Einsatz von Emulatoren arbeiten und lernen Studierende mit realitätsnahen Tools, können Laborsitzungen vorbereiten, ihre Arbeit speichern und orts- sowie zeitunabhängig auch außerhalb der

Hochschule fortsetzen. Im Laufe der vergangenen Semester wurde hauptsächlich Cisco VIRL zur Realisierung von Virtual Network Testbeds im NetLab eingesetzt. Während diese Lösung unsere Anforderungen weitgehend erfüllte, führt die Umstellung des Lizenzmodells auf Einzelplatzversionen dazu, dass sich VIRL zukünftig nur schwer im NetLab verteilen und warten lässt. Ein Umstieg auf die von Cisco beworbene Mehrplatz-Alternative CML ist für uns mit zu hohen Kosten verbunden und somit nicht realisierbar. Daher wird ein Umstieg auf GNS3 evaluiert, welches seit der aktuellen Version einen Großteil unserer Anforderungen erfüllt. Zwar bietet GNS3 derzeit keine automatisierte Lastverteilung, jedoch benötigt es in Bezug auf die Skalierbarkeit im Vergleich zu VIRL weniger Ressourcen. Für skalierbare und forschungsnaher Testbeds (z.B. in Master-Lehrveranstaltungen) bleibt Mininet, insbesondere im SDN-Umfeld, das Mittel der Wahl auch wenn es insgesamt weniger unserer Anforderungen erfüllt als GNS3 und VIRL. Für die Zukunft planen wir neben GNS3 auch weitere Alternativen wie EVE-NG zu betrachten, da diese vielversprechende Ziele verfolgen. Zudem planen wir eine detaillierte Analyse der Performance und Skalierbarkeit von GNS3 im Vergleich zu Cisco VIRL durchzuführen.

Literatur

- [18a] Cisco Modeling Labs, <http://www.cisco.com/c/en/us/products/cloud-systems-management/modeling-labs>, abgerufen am: 2018-04-16, 2018.
- [18b] Emulated Virtual Environment Next Generation (EVE-NG), <http://www.eve-ng.com/>, abgerufen am: 2018-04-16, 2018.
- [18c] eNSP - Enterprise Network Simulator, <http://support.huawei.com/enterprise/en/network-management/ensp-pid-9017384>, abgerufen am: 2018-04-16, 2018.
- [18d] GNS3 - The software that empowers network professionals, <https://www.gns3.com>, abgerufen am: 2018-04-16, 2018.
- [18e] Mininet - An Instant Virtual Network on your Laptop (or other PC), <http://mininet.org>, abgerufen am: 2018-04-16, 2018.
- [18f] ns-3, <https://www.nsnam.org>, abgerufen am: 2018-04-16, 2018.
- [18g] Packet Tracer - A free network simulation and visualization tool for the IoT era. <https://www.netacad.com/about-networking-academy/packet-tracer>, abgerufen am: 2018-04-16, 2018.
- [18h] Unified Networking Lab v2 (UNetLabv2), <http://www.routereflector.com/unetlab/>, abgerufen am: 2018-04-16, 2018.
- [18i] VIRL - Virtual Internet Routing Lab, <http://virl.cisco.com>, abgerufen am: 2018-04-16, 2018.
- [18j] VIRL Academic license will no longer be available. <https://learningnetwork.cisco.com/thread/103534>, abgerufen am: 2018-04-16, 2018.

- [Ba03] Baumgartner, F.; Braun, T.; Kurt, E.; Weyland, A.: Virtual Routers: A Tool for Networking Research and Education. *SIGCOMM Comput. Commun. Rev.* 33/3, S. 127–135, Juli 2003, ISSN: 0146-4833.
- [Ba17] Bajpai, V.; Kühlewind, M.; Ott, J.; Schönwälder, J.; Sperotto, A.; Trammell, B.: Challenges with Reproducibility. In: *Proceedings of the Reproducibility Workshop, Reproducibility@SIGCOMM 2017*, Los Angeles, CA, USA, August 25, 2017. S. 1–4, 2017.
- [Fl17] Flittner, M.; Bauer, R.; Rizk, A.; Geißler, S.; Zinner, T.; Zitterbart, M.: Taming the Complexity of Artifact Reproducibility. In: *Proceedings of the Reproducibility Workshop, Reproducibility@SIGCOMM 2017*, Los Angeles, CA, USA, August 25, 2017. S. 14–16, 2017.
- [Ha12] Handigol, N.; Heller, B.; Jeyakumar, V.; Lantz, B.; McKeown, N.: Reproducible network experiments using container-based emulation. In: *Proceedings of the 8th international conference on Emerging networking experiments and technologies*. ACM, S. 253–264, 2012.
- [KE04] Kayssi, A.; El-Haj-Mahmoud, A.: EmuNET: A Real-time Network Emulator. In: *Proceedings of the 2004 ACM Symposium on Applied Computing*. SAC '04, ACM, New York, NY, USA, 2004.
- [MK16] Momeni, B.; Kharrazi, M.: Partov - a network simulation and emulation tool. *J. Simulation* 10/4, S. 237–250, 2016.
- [Ob14] Obstfeld, J.; Knight, S.; Kern, E.; Wang, Q. S.; Bryan, T.; Bourque, D.: VIRL: The Virtual Internet Routing Lab. *SIGCOMM Comput. Commun. Rev.* 44/4, S. 577–578, Aug. 2014.
- [PR16] Pizzonia, M.; Rimondini, M.: Netkit: network emulation for education. *Software: Practice and Experience* 46/2, S. 133–165, Feb. 2016.
- [Ri17] Rieger, S.: Skalierbare virtuelle Netz-Testbeds für Lehr- und Forschungsumgebungen am VIRL. In: *10. DFN-Forum - Kommunikationstechnologien*, 30.-31 Mai 2017, Berlin, Germany. S. 125–134, 2017.
- [SRP17] Seifert, C.; Rieger, S.; Pape, C.: Realization Possibilities for Virtual Networking Labs in Higher Education Courses. In: *13 th Annual International Conference on Computer Science and Education in Computer Science (CSECS)*. 2017.
- [Ta16] Tagliacane, S. V.; Prasad, P. W. C.; Zajko, G.; Elchouemi, A.; Singh, A. K.: Network simulations and future technologies in teaching networking courses: Development of a laboratory model with Cisco Virtual Internet Routing Lab (Virl). In: *2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*. IEEE, S. 644–649, 2016.
- [YM17] Yan, L.; McKeown, N.: Learning Networking by Reproducing Research Results. *ACM SIGCOMM Computer Communication Review* 47/2, S. 19–26, 2017.