

Using Looking Glasses to Understand the Internet's Structure

Jagjit Singh

Betreuer: Quirin Scheitle

Seminar Innovative Internettechnologien und Mobilkommunikation SS2016

Lehrstuhl Netzarchitekturen und Netzdienste

Fakultät für Informatik, Technische Universität München

Email: jagjit.singh@tum.de

KURZFASSUNG

Looking Glasses (LGs) spielen eine sehr wichtige Rolle, um die Struktur des Internets besser darstellen zu können. Sie bieten sowohl Kontrollschicht- als auch Datenschichtmessungen von verschiedenen Blickwinkeln aus dem Internet. Da es jedoch keinen Standard für die Ein- und Ausgabe der Daten gibt, wird die Zusammenarbeit verschiedener LG Server erschwert, die wichtig ist, um das Potenzial auszunutzen. Es gibt mehrere Frameworks, die die Zusammenarbeit verschiedener LGs erlauben. Eines dieser Frameworks, Periscope, wird in diesem Paper genauer untersucht.

Schlüsselworte

Looking Glass (LG) Server, Border Gateway Protocol (BGP), Inter-Domain Routing, Autonomous Systems (ASes)

1. EINLEITUNG

Messungs- und Monitoring-Tools sind sehr wichtig und notwendig, denn sie bieten Lösungen für mehrere Problemstellungen: Von wissenschaftlicher Forschung bis hin zur Ermittlung von Topologien, Anomalien und Sicherheitsmängeln in den Netzwerken.

Viele Internetdienstleister verwenden eigene Diagnose- und Monitorings-Tools, unter anderem auch Looking Glass Server, um Störungen im eigenen Netz schnell finden zu können und die Leistung des Netzes zu verbessern. LG Server bieten eine Reihe von Befehlen wie *Ping*, *Traceroute* und *Border Gateway Protocol (BGP)*-Befehle als Webschnittstelle an. Diese Befehle können verwendet werden, um Informationen über Übertragungswege zu erhalten, die sehr hilfreich sind, um z.B die Ursache einer Störung, wie falsch konfigurierte BGP Route Advertisement, ausfindig zu machen [12]. Traceroute- und BGP-Anfragen liefern Informationen über die logischen Verbindungen der Border-Router und wie die Datenpakete tatsächlich weitergeleitet werden. Das Problem allerdings ist, dass verschiedene LG Server unterschiedlich implementiert sind. Jeder LG Server wird unabhängig von anderen LG Servern betrieben und es gibt keine zentrale Datenbank, die alle LG Server indiziert. Es gibt auch keinen Standard für das Format der Anfragen und Ausgabe der angefragten Daten. LG Server sind für die manuelle Nutzung gedacht. Die Anzahl der Anfragen ist auch limitiert, um DoS-Attacken zu vermeiden.

Ein Schwerpunkt dieses Papers ist das Framework Periscope. Dieses Framework vereinigt die unterschiedlichen, von-

einander unabhängig betriebenen LG Server und bietet eine standardisierte öffentlich verfügbare API, die on-demand Messungen erlaubt [7].

Der weitere Aufbau dieser Ausarbeitung ist wie folgt: Abschnitt zwei beschreibt kurz eine abstrakte Sicht für den Aufbau des Internets, im Abschnitt drei werden verschiedene Möglichkeiten der Ermittlung von AS-Topologien beschrieben und Studien vorgestellt, die auch mit LGs arbeiten. Im Abschnitt vier werden die Periscope Architektur und dessen Komponenten beschrieben. Abschnitt fünf fasst die behandelten Themen noch einmal zusammen und gibt weitere Verweise zur Nutzung von Periscope.

2. GRUNDLAGEN

Das Internet besteht aus Autonomen Systemen (eng. Autonomous Systems ASes), die Inter-Domain Routing Informationen mit Hilfe des Border Gateway Protocols (BGP) untereinander austauschen [23]. Das Internet kann als ein Topologie-Graph auf AS-Ebene betrachtet werden, wobei jedes Autonome System einen Knoten darstellt und eine logische Verbindung zwischen zwei Autonomen Systemen eine Kante. Diese logische Verbindung zwischen zwei ASes wird auch Link genannt. LG Server können sowohl als Webschnittstelle, als auch via Telnet zur Verfügung gestellt werden und sind öffentlich zugänglich. Einige LG Server erlauben das Auslesen von Routingtabellen.

3. MESSMÖGLICHKEITEN

Es gibt unterschiedliche Möglichkeiten Topologien auf AS-Ebene zu erstellen. Dafür können Datenquellen verwendet werden, die unterschiedliche Methoden verwenden, um die Topologien zu erstellen: BGP basiert (IRL [24], RouteView [20], RIPE-RIS [19]), Traceroute-basiert (Ark [3], iPlane [13]) und Internet Routing Registry (IRR) [9]. Jedoch können die unterschiedlichen Datenbanken veraltet sein. Ein Paper von Khan *et al.*[12] beschreibt eine AS-Topologie mit Hilfe der LG Server. Dafür kommen BGP-Befehle wie *show ip bgp summary* zum Einsatz. Diese liefern die Routingtabellen von ASes vom jeweiligen Server, in dem sich der LG Server befindet. Eine typische Ausgabe zum Beispiel von einem Cisco Router sieht wie folgt aus:

BGP router identifier 10.0.0.1, local AS number 123

```
...
Neighbor    V  AS  others
192.168.0.1  4  123  ...
192.168.1.1  4  456  ...
```

Das Ergebnis liefert die Information, dass der angefragte Router in AS123 liegt und zwei Nachbarrouter besitzt, wobei einer im gleichen AS ist und der andere sich in AS456 befindet. Die AS-Topologie enthält dann zwei Knoten (123, 456) und eine Kante (123-456).

3.1 Verschiedene AS-Topologien

In diesem Abschnitt werden einige AS Topologien kurz beschrieben, die in den Studien von Khan *et al.* [12], Augustin *et al.* [1] und Zhang *et al.* [26] verwendet wurden.

BGP Trace Collectors: BGP Trace Collectors [26] sind mit kommerziellen ISP Netzwerken über BGP Sitzungen verbunden. Ein Collector empfängt BGP Nachrichten von anderen Routern, die damit verbunden sind, schickt jedoch keine Prefix-Advertisements zu anderen Routern. So sammelt ein Collector die Routingtabellen und Updates, die er von anderen Routern empfangen hat.

Routingtabellen enthalten Informationen über bevorzugte Pfade, um ein Zielpräfix von IP Adressen zu erreichen. Die Routingupdates liefern Informationen über alternative Pfade. RouteViews [20] und RIPE RIS [19] sind die Hauptprojekte, die BGP Trace Collectors aufstellen und die gesammelten Daten veröffentlichen.

Route Servers: Route Servers sind Router, die von einigen Internetdiensteanbietern öffentlich gemacht werden. Benutzer können sich über Telnet mit diesen Routern verbinden und verschiedene Router-Befehle wie *show ip bgp* ausführen, um die ganze Routingtabelle zu sehen. Allerdings speichern die Route Server keine Routing-Updates, wie dies bei BGP-Collectors der Fall ist.

IXP Datenbanken: Packet Clearing House [15] und PeeringDB[16] sind zwei Hauptquellen, die Daten über IXPs liefern. Sie enthalten Informationen wie IXP Namen, deren Standorte, IXP Präfixe, Liste der Mitglieder und Hyperlinks zu IXP Webseiten. Beide Datenbanken basieren auf freiwilligen Beiträgen. Packet Clearing House löscht nie IXPs, die einmal in der Datenbank sind. Sie werden als "stillgelegt" markiert, sobald die nötigen Beweise dafür verfügbar sind.

IRL: UCLA IRL [24] veröffentlicht regelmäßig die AS Topologie, welche von unterschiedlichen Datenquellen wie RouteViews[20], RIPE-RIS [19], Packet Clearing House (PCH) [15] und Internet2 [10] extrahiert wird [12].

Ark: CAIDA Archipelago (Ark) [3] bietet eine AS Topologie an, die auf Traceroute-Messungen basiert. Im März 2013 waren 71 Ark-Monitors über das Internet verteilt. Es wurden 116K direkte AS-Verbindungen von dem IPv4 Route /24 AS Datensatz verwendet [12].

iPlane: Madhyasta *et al.* [13] haben iPlane Dienst vorgeschlagen, welches Traceroute-Befehle von über 300 Planet-Lab Standorten schickt, um die Internet Topologie zu mappen. Es wurden 81 K AS-Links von iPlane AS Topologie im März 2013 gefunden [12].

Internet Routing Registry (IRR): IRR [9] ist eine Datenbank, in der Routing- und Adresseninformationen registriert sein können. Selbst wenn einige Informationen nicht mehr aktuell sind [11], enthält die Datenbank Informationen über AS-Links, die nicht in BGP und Traceroute-basierten Topologien enthalten sind[8] [12].

3.2 AS Topologie Ermittlung mithilfe von LG Servern

Dieser Abschnitt beschreibt, wie Khan *et al.*[12] eine AS-Topologie mit Hilfe der Daten von verschiedene LG Servern erstellen.

Ein automatisiertes Tool verarbeitet Anfragen an 388 verschiedene LG Servern. Das Tool schickt gleichzeitig 30 Anfragen an die LG Server und legt 15 Sekunden Pause ein, falls mehrere aufeinanderfolgende Anfragen an den gleichen LG Server gerichtet sind. Es sind fünf Schritte notwendig, um die Daten von den LG Servern zu sammeln:

- Das Tool parst die Website vom LG Server, um die unterstützten LG Befehle und LG Router zu finden, damit später Abfragen an sie geschickt werden können.
- An jeden LG Router wird ein *show ip bgp summary* Befehl über den LG Server gesendet.
- Aus dem Ergebnis extrahiert das Tool die IP-Adressen von den Nachbarroutern des LG Servers.
- An die gefundenen IP-Adressen der Nachbarrouter wird ein *BGP neighbor ip advertised routes*-Befehl gesendet, um die vorgeschlagenen Routen von dem ursprünglichen LG Server an die Nachbarn zu finden.
- Zum Schluss werden die Ergebnisse der beiden Befehle *show ip bgp summary* und *BGP neighbor ip advertised routes* in einer Textdatei gespeichert, um daraus die AS-Topologie konstruieren zu können.

245 von 388 LG Servern unterstützen den *show ip bgp summary*-Befehl von 1.900 Standorten, die über 110 Länder verteilt sind. Mit den 245 LG Servern wurden im März 2013 zweimal in der Woche Messungen durchgeführt. Das Ergebnis der insgesamt 8 Schnappschüsse stellt kombiniert einen AS-Link Datensatz dar. Die Intra-AS Links, deren Ziel- und Quelladressen sich im selben AS befinden, werden dabei ausgefiltert und es bleiben 16.000 Inter-AS Links übrig.

Von den 245 LG Servern unterstützen 59 LG Server den *BGP neighbor ip advertised routes*-Befehl, welcher einmal wöchentlich im März 2013 ausgeführt wurde. Deren LG Router sind an 250 Standorten in über 40 Ländern verteilt und schicken an 5.000 Routern in den Nachbar-ASes die Route-Advertisements. Die Studie hat 686 Autonome Systeme und 11.000 AS-Links entdeckt, die nicht in anderen Datenbanken, wie IRL, IRR, Ark und iPlane vorkommen.

3.3 IXPs mithilfe von AS-Topologie mappen

„IXPs: Mapped?“ ist eine Studie von Augustin *et al.* [1] die einen Ansatz vorstellt, um IXPs zu mappen, damit mehr Informationen über Peering-Matrizen (z.B. wer verbindet sich mit wem über welchen IXP) und den dadurch entstehenden Verkehr im Netz (z. B wie viele Daten tauschen die verschiedenen ASes über einen IXP aus) verfügbar sind.

Mehrere Datenquellen spielen eine entscheidende Rolle. Als erste Datenquelle kommt eine Liste von 278 IXPs zum Einsatz, die insgesamt 393 Präfixe, deren geografischen Standorte und Mitglieder enthält. Eine entscheidende Rolle spielt eine weitere Datenquelle, die 2300 LG Server auflistet. Diese erlauben das Ausführen von Traceroute-Befehlen und sind in

66 Ländern verteilt. Auch die AS Nummern (ASN) der zugehörigen Netzwerken sind bekannt. Diese Liste wurde mithilfe von *traceroute.org* Datenbank [22] erstellt und 486 weitere LGs wurden mithilfe von PeeringDB[16] hinzugefügt und 20 LGs wurde mithilfe von Suchmaschinen gefunden. Die dritte wichtige Komponente ist die gefolgerte AS-Karte, die beim Zusammenfügen von BGP Routingtabellen von RIPE RIS und RouteViews Projekte erstellt wurde und es wurde die aktuellste Version des Cyclops Projekts [6] heruntergeladen. Diese AS-Karte wird um von der Studie neu entdeckte Peerings ergänzt. Eine typische Cyclops-basierte AS-Karte enthält ungefähr 110.000 AS Links. Die erweiterte Karte von Augustin *et al.* [1] enthält ungefähr 10-20% mehr AS Links. Es wurden gezielte Traceroute-Experimente im Juli 2008, Dezember 2008 und April 2009 durchgeführt. Zusätzlich dazu wurden von der PlanetLab [17] Datenbank 254 aktive Knoten gewählt, um die Ergebnisse verschiedener Plattformen zu vergleichen. Anhand dieser Listen wurde von jedem aktiven Knoten und für jedes AS, ein Traceroute-Befehl zu einer IP Adresse durchgeführt, die auf den Ping-Befehl reagierte. Die Abbildung 2 listet für jede Studie die Anzahl der verschiedenen ASes, Länder und Regionen, in der sich die Traceroute-Quellen befinden. Die Abbildung 1 zeigt, wie die 2,3K LGs auf der Welt verteilt sind. Die Abbildung 3 zeigt, wie viele LGs pro AS verfügbar sind. Einige ASes bieten eine große Anzahl an LGs an, die weltweit verteilt sind. Viele ASes haben jedoch nur ein LG. Die meisten LGs befinden sich in Europa und Nordamerika.

Region	# LGs	Region	# LGs
Europe	1,361	South America	84
North America	718	Australia & New Z.	58
Asia	104	Africa	4

Abbildung 1: Anzahl LGs nach Standorten [1]

	CAIDA	PlanetLab	DIMES	LG
Sources	26	254	18K	2.3K
AS	26	223	n.a.	406
Countries	18	31	113	66

Abbildung 2: Reichweite verschiedener Datensätze [1]

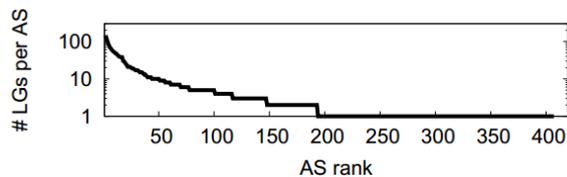


Abbildung 3: Verteilung der Anzahl der LGs pro AS[1]

4. PERISCOPE ARCHITEKTUR

Es gibt vier Hauptprobleme, die Periscope berücksichtigen muss [7]:

- Da es keine Standards für die Eingabe der Anfragen und Ausgabe der Ergebnisse gibt, muss Periscope die

Anfragen für jeden LG entsprechend anpassen. Ebenso müssen die Ergebnisse in einem benutzerfreundlichen Format dargestellt werden [7].

- Es gibt keine zuverlässige Liste, die vorhandene, gültige LG Server beinhaltet. Periscope muss in der Lage sein, automatisch LG Spezifikationen zu ermitteln und validieren, die von verschiedenen Quellen stammen [7].
- Die Verfügbarkeit und Spezifikationen der LG Server ändert sich häufig. Periscope muss automatisch die Änderungen ermitteln und Spezifikationen anpassen [7].
- LGs sind für kleine Anfragen gedacht und blockieren somit die Benutzer, die eine bestimmte Anzahl an Anfragen überschreiten. Periscope muss in der Lage sein, gleichzeitig Anfragen von verschiedenen Benutzer auszuführen ohne die Grenzen von LGs zu verletzen [7].

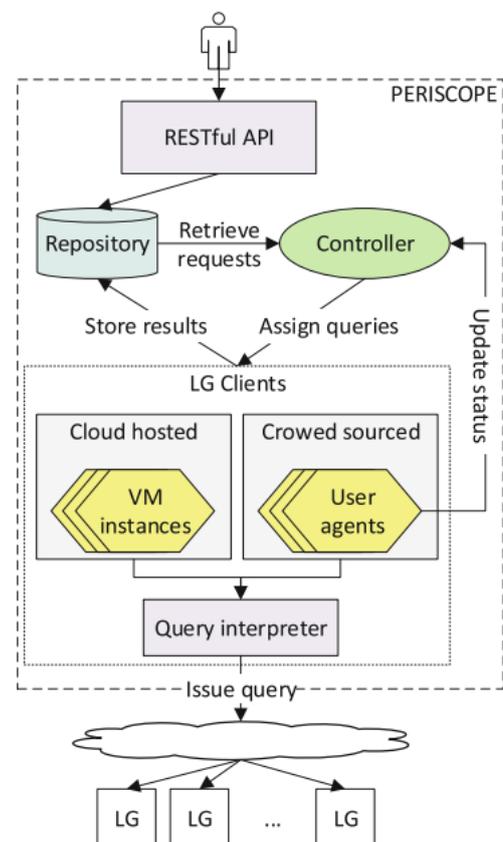


Abbildung 4: Periscope Architektur [7]

4.1 Arbeitsablauf vom Periscope

Als erstes holt Periscope die Listen der verfügbaren LG Server, die noch aktiv sind. Dafür werden öffentlich verfügbare Ressourcen, wie PeeringDB[16] und *traceroute.org*, die LG URLs beinhalten, verwendet. Diese Liste kann jedoch veraltet sein und auch nicht mehr aktive LGs auflisten. Deshalb besucht ein Web Crawler die URLs der LG Server und filtert dabei nicht erreichbare LGs heraus.

Um festzustellen, ob die URLs überhaupt auf LGs verweisen, wird im HTML Quelltext überprüft, ob dieser ein Element der Form eines LG beinhaltet. Die meisten LGs basieren auf Open-Source-Projekten. [2] Es wurde für sieben bekannte Open-Source-Projekte jeweils eine Vorlage der Implementierung erstellt, welche die HTTP-Elemente und HTML-Parameter beschreibt.

Ein Web-Scraper extrahiert die *<Form>*-Elemente aus dem HTML-Quelltext und vergleicht die Eingabefelder mit den vorhandenen Vorlagen. Wenn mindestens eine Vorlage alle Eingabefelder aus dem HTML-Quelltext extrahierten Felder enthält, gilt diese als passende Vorlage. Dabei können die extrahierten Elemente eine Teilmenge der Elemente der Vorlage sein.

Sobald eine passende Vorlage gefunden ist, erstellt Periscope eine JSON-Konfigurationsdatei, die das LG-Interface beschreibt. Des Weiteren enthält die Konfigurationsdatei die HTTP-Anfrage und die Eingabeparameter mit zulässigen Eingabewerten. Diese JSON-Konfigurationsdatei wird vom *Query Parser* verwendet, um die Messanfragen in ein Format zu übersetzen, das vom jeweiligen LG unterstützt wird. Findet der *Web Scraper* jedoch keine passende Vorlage, so sucht er nach LG-spezifischen Schlüsselwörtern wie z.B. Netzwerkbefehle. Wird ein Schlüsselwort gefunden, so muss das Formelement manuell geparkt werden. Im Anschluss wird eine aktualisierte Vorlage erstellt, um LGs ähnlicher Form in Zukunft automatisch erkennen zu können. Abbildung 5 zeigt eine Beispielvorgabe für die Eingabeparameter des Version 6 [25] LGs. Der letzte Schritt ist es, die Richtigkeit der automatisch generierten Konfigurationsdateien sicherzustellen. Ein *Health Checker* verwendet den *Query Parser*, um eine Messung anzufordern und verarbeitet das Ergebnis, das der Parser zurückliefert. Wird ein HTTP-Fehler zurückgeliefert oder handelt es sich um ein leeres Ergebnis, so markiert der *Health Checker* den verwendeten LG für eine manuelle Untersuchung. Diese Tests werden vom *Health Checker* regelmäßig durchgeführt, um die Änderungen in LG-Vorlagen, Eingabeparametern oder in HTTP-Rückmeldungen festzustellen [7]. Abbildung 6 zeigt den Arbeitsablauf, um LGs zu entdecken und deren Spezifikationen von Webquellen zu extrahieren.

Input name	Input type	Expected values	Meaning
query	radio	[bgp, trace, ping]	[sh ip bgp, traceroute, ping]
addr	text	*	Query target
router	select	*	Router identifier
protocol	select	[IPv4, IPv6]	IP version

Abbildung 5: Beispielvorgabe der Eingabeparameter von Version 6 LG [25]

4.2 Komponenten der Periscope-Architektur

Periscope stellt *RESTful* API bereit, die verwendet wird, um Anfragen an vorhandenen LGs zu schicken, neue Messungen durchzuführen und Ergebnisse abzurufen. Abbildung 4 zeigt die verschiedenen Komponenten von der Periscope-Architektur. Alle Anfragen werden in einem *Repository* gespeichert. Das dient der Vermittlung zwischen Periscope-API und den restlichen Komponenten von Periscope.

Ein *LG Client* fragt die in der *Repository* gespeicherten Anfragen an und übersetzt diese in LG-Abfragen. LG Clients

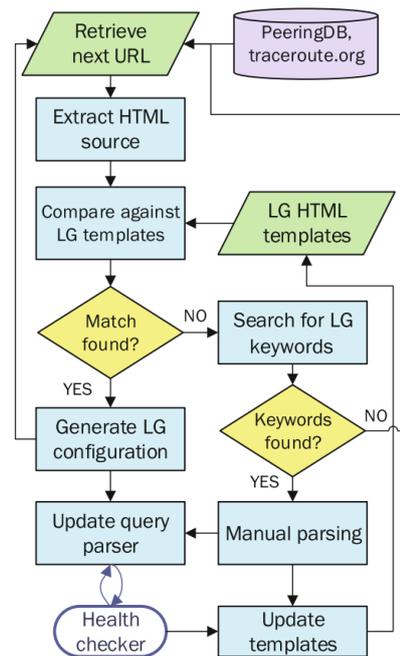


Abbildung 6: Arbeitsablauf um LGs zu entdecken und Spezifikationen zu extrahieren [7]

verwenden Selenium [21], um die Anfragen auszuführen. Selenium ist eine Web-Browser-Automatisierungssuite, die mit LGs über JSON-Konfigurationsdateien interagiert. LGs sind nicht für eine große Anzahl an Anfragen gedacht. Überschreitet ein Client eine bestimmte Anzahl an Anfragen in einem bestimmten Zeitintervall, so wird er vorläufig blockiert. Es wird zum Beispiel die IP-Adresse vom Client und die Zeit der Anfrage gespeichert, damit der LG-Server entscheidet, ob er die nächste Anfrage vom selben Client zulässt oder nicht. Aus diesem Grund kann Periscope nicht gleichzeitig Anfragen von unterschiedlichen Benutzern, die an den gleichen LG-Server gerichtet sind, von einer öffentlichen IP-Adresse verarbeiten. Dieses Problem wird gelöst, indem jedem LG-Client eine andere öffentliche IP-Adresse zugeordnet wird und somit die entsprechenden Anfragen von unterschiedlichen IP-Adressen an den LG-Server gesendet werden.

Der erste Ansatz, um unterschiedliche öffentliche IP-Adressen für die LG-Clients zu verwenden, ist *Crowd-Sourced User Agents* zu verwenden. Der zweite Ansatz ist *Cloud-basierte Virtuelle Maschinen (VMs)* zu verwenden. Jedem LG-Client wird eine Instanz der virtuellen Maschine zugewiesen. Periscope schaltet die VMs bei Bedarf ein und aus. Es sind mindestens so viele LG-Clients notwendig wie die Anzahl der unterschiedlichen Periscope-Benutzer, die gleichzeitig Anfragen an einen LG-Server schicken. Periscope versucht erst mit dem ersten Ansatz die Nachfrage zu bedienen und schaltet

bei Bedarf virtuelle Maschinen ein.

Ein zentraler *Controller* kommt zum Einsatz, um jeder Messungsanfrage einen LG Client zuzuweisen. Da der Controller einen globalen Überblick vom System hat, werden die Anfragen so bearbeitet, dass die auf einzelnen LG Server eingestellte Abfragegrenzen nicht überschritten werden. Der Controller verwaltet dabei sowohl Cloud-basierten VM-Instanzen als auch die Crowd-Sourced-Instanzen. Letztere schicken alle fünf Minuten eine Keep-Alive-Nachricht, um den Controller mitzuteilen, dass sie bereit sind, weitere Abfragen zu verarbeiten. Sobald eine neue Messanfrage eingeht, entscheidet der Controller wann und welche Instanz die Anfrage verarbeiten soll. Die Logik des Controllers basiert auf zwei LG-spezifischen Variablen, die die maximale Anzahl gleichzeitiger Abfragen an einem LG beschränken¹:

1. *Timeout* ist die minimale Zeit zwischen zwei aufeinanderfolgenden LG Abfragen von dem selben Benutzer.
2. *Query slots* ist die maximale Anzahl der Abfragen, die Periscope zu einem bestimmten Zeitpunkt für einen LG akzeptieren kann.

Damit ein Benutzer überhaupt eine Anfrage an einem LG schicken kann, muss mindestens ein Query Slot übrig sein. Anderenfalls wird die Anfrage zu einer Schlange hinzugefügt, damit sie später verarbeitet werden kann. Wann die Anfrage zur Verarbeitung bereit ist, entscheidet der Controller. Für jede Anfrage berechnet der Controller den Zeitpunkt, zu dem er die Anfrage durchführen kann. Dieser Zeitpunkt basiert auf dem Zeitpunkt der letzten Anfrage des gleichen Benutzers zum gleichen LG und auf der Timeout-Variablen des LGs. Die Anfrage wird so lange in der Schlange gespeichert, bis mindestens ein Query Slot frei wird und die Timeout-Zeit abgelaufen ist. Wenn alle Anforderungen zur Verarbeitung erfüllt sind, wird die Anfrage von der Schlange genommen und einer geeigneten Client-Instanz zur Ausführung zugeordnet. Eine Instanz ist für die Ausführung der Anfrage geeignet, wenn diese für die Dauer der Timeout-Zeit keine Anfragen an der entsprechenden LG geschickt hat. Steht keine geeignete Client-Instanz zur Ausführung bereit, beantragt der Controller eine neue Cloud-Instanz. Die Anzahl der aktiven Client-Instanzen hängt lediglich von der Anzahl der Anfragen ab, die von verschiedenen Benutzer gleichzeitig an einem LG geschickt werden und nicht von der Anzahl der aktiven Benutzer oder angefragten LGs.

4.3 Evaluierung

Periscope hat für 262 LGs automatisch die Konfiguration erstellt. 35 weitere wurden manuell geparkt, da es am Anfang keine passende Vorlagen für sie gab. Insgesamt wurden bis Dezember 2015 297 LGs von Autonomen Systemen extrahiert. Es ist nicht immer leicht LGs zu AS Nummern (ASN) zu mappen. Deshalb wird die IP des LG Hosts verwendet, um diese mit Hilfe der Longest Prefix Match Methode zu einer ASN zuzuordnen. Um die IP Adresse von dem Router, die die einzelnen LGs hostet, zu ermitteln, wird ein Traceroute-Befehl zum Router ausgeführt. Mit *tcpdump* wird im Anschluss die Quelladresse von den ankommenden Paketen extrahiert. Dieses Verfahren wird auch verwendet um die Traceroute Protokolle der verschiedenen LGs zu entdecken. 266 LGs verwenden UDP-basiertes Traceroute und 31 LGs ICMP-basiertes Traceroute.

¹Die Werte der timeout und query slot Variablen wurden dabei empirisch konservativ abgeleitet[7].

Viele LGs unterstützen das Ausführen der Befehle aus verschiedenen Blickwinkeln (eng. Vantage Points (VPs)) innerhalb eines Autonomen Systems. Wenn das LG-Interface keine Informationen über den Standort liefert, wird die gleiche Methode, wie bei Ermittlung von ASN vom jeweiligen LG, verwendet, um den jeweiligen LG zu einer Stadt zuzuordnen. Dabei wird die IP Adresse des LGs mit Hilfe der NetAcuity's geolocation Datenbank [14] zu einer Stadt zugeordnet. Periscope hat automatisch aus 262 LGs 1.691 VPs geparkt. Diese sind über 501 Städten in 76 Ländern verteilt. 40% der LGs haben mehr als einen Blickwinkel auf Stadtebene und 20% der LGs haben 10 oder mehr VPs. Abbildung 7 zeigt, wie viele VPs die jeweiligen Befehle unterstützen. Mehr als 75% der VPs unterstützen Kontrollschicht- als auch Datenschichtmessungen und 60% unterstützen IPv6-Befehle zusätzlich zu IPv4-Befehlen.

Um die Topologien von Periscope, Atlas und Ark VPs zu vergleichen, wurden in Oktober 2015 Traceroute-Befehle zu 2.000 Zielen von jeder Plattform ausgeführt. Zu diesem Zeitpunkt hatte Atlas 7.292 öffentliche Ressourcen in 2.779 verschiedenen ASes, verteilt in 160 Ländern. Ark hatte 107 Ressourcen in 71 ASes, die in 41 Ländern verteilt waren.

Um eine wertneutrale Menge von Zielen zu erstellen, wurden zuerst die IP Adressen gesammelt, die in iPlane Datenbank [13] waren und im Anschluss wurde ein *ZMap* Scan ausgeführt, um nur IP Adressen zu behalten, die auf UDP und ICMP Befehlen geantwortet haben. Die IP Adressen wurden zu deren zugehörigen ASes gemappt und für jedes AS wurde zufällig eine IP Adresse ausgewählt, bis eine Zielmenge von 2.000 IP Adressen erreicht wurde. Eine kleine Zielmenge ist nötig, da die LGs und Atlas Infrastrukturen nur eine beschränkte Anzahl an Anfragen zulassen. Es wurden Messungen von allen Atlas Netzwerk Monitoren ausgeführt. Mehr als 6 Millionen Traceroute-Daten wurden dabei innerhalb von zwei Monaten gesammelt. Dies war nur mit einem Account mit höheren Rechten möglich. Mit einem Standardaccount hätte es ungefähr fünf Jahre gedauert [18].

Traces von LG VPs überquerten 3.109 ASes, 29.525 AS-Links und 167 IXPs. Die Traces von der Atlas Plattform überquerten 3.369 ASes, 55.936 AS-Links und 171 IXPs zu gleichen Zielen. Bei Ark waren es 1.608 ASes, 10.237 AS-Links und 136 IXPs. In der Tabelle 1 ist die Anzahl der ASes, AS-Links und IXPs von jeweiligen Datensätzen aufgelistet, auch die, die eindeutig sind. 47% (13.969 von 29.525) AS-Links waren nicht in den Tracedaten von Atlas oder Ark enthalten und 26% (809 von 3.109) ASes von LG Traces waren nicht in Atlas oder Ark Datensätzen enthalten. 16 IXPs die in LG Trace-Datensätzen gesichtet wurden, waren nicht in den Datensätzen von Atlas oder Ark Traces.

Die neu entdeckten ASes und AS-Links erweitern bisherige AS-Topologien. Mithilfe von Periscope können viele weitere, bisher unentdeckte, ASes und Links ermittelt werden. Das Konzept zeigt, wie man verschiedene LGs effizient verwenden kann, ohne dabei deren Grenzen zu verletzen.

4.4 Test-Messungen

Die API von Periscope ist auf der offiziellen Website von CAIDA [4] dokumentiert. Hier [5] ist ein Beispielbefehl für eine neue Messung aufgelistet. Um eine neue Messung durchzuführen, braucht man Zugangsdaten, die von CAIDA auf Nachfrage zur Verfügung gestellt werden. Da kein Kontakt mit CAIDA hergestellt werden konnte, war es auch nicht

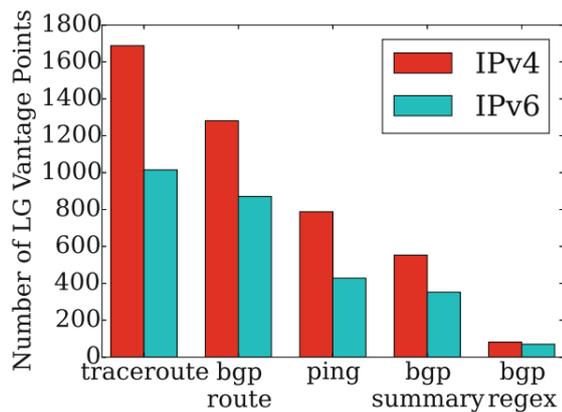


Abbildung 7: Anzahl der VPs verschiedener Befehle[7]

Tabelle 1: Anzahl ASes und Verbindungen dazwischen im LG, Atlas und Ark Datensätzen[7]

Dataset	ASes		AS Links	
	Gesamt	Eindeutig	Gesamt	Eindeutig
LG-Periscope	3.109	809	29.525	13.969
Atlas	3.369	1.464	55.936	40.620
Ark	1.608	59	10.237	1.625
LG Khan [12]	45.400	686	116.000	11.000
IXP: Mapped	3.300		36.000	29.600

möglich, Messungen durchzuführen. Aus dem selben Grund konnten auch keine neue LGs hinzugefügt werden.

5. ZUSAMMENFASSUNG

Messungen im Internet und dazugehörige Monitoring-Tools sind wichtig, um die aktuelle Netzwerkstruktur besser zu verstehen, Störungen in Netzwerken zu finden und um diese schneller beheben zu können und die Struktur des Internets in der Zukunft verbessern zu können. LGs spielen dabei eine sehr wichtige Rolle, denn diese erlauben sowohl Traceroute-Befehle, Ping-Befehle als auch BGP-Befehle und bieten somit Kontrollschicht- als auch Datenschichtmessungen an. Da die verschiedenen LGs unterschiedlich implementiert sein können, wird die Zusammenarbeit dieser verschiedenen LGs erschwert. Mehrere Studien stelle Konzepte vor, die die Zusammenarbeit unterschiedlich implementierter LGs ermöglichen. Periscope ist eines dieser Konzepte, das die LGs verschiedener Provider zusammenfasst. Es ermöglicht den Benutzern Anfragen an beliebigen LGs zu schicken, ohne direkt mit den entsprechenden LGs zu interagieren und sich Gedanken über Timeout-Limits und Anfragen-Limits zu machen oder selber Code zu entwickeln, um automatisiert Anfragen an LGs verschicken zu können. Periscope erweitert die AS-Topologie, die anhand der Daten von Atlas und Ark erstellt werden, um weitere wichtigen Daten. Periscope berücksichtigt die Begrenzung von LGs und erlaubt den Benutzern nicht, mehr Anfragen an LGs zu schicken, als sie in einem gegebenen Moment verarbeiten können. Es werden dabei von Periscope mehrere Cloud-basierte und Crowd-Sourced-Instanzen verwendet. Es ist auch wichtig, dass die von LGs festgelegten Grenzen nicht überschritten werden,

denn sonst würde die Verfügbarkeit der LGs für öffentlichen Zwecke reduziert werden[7]. CAIDA's Archipelago [3] Infrastruktur bietet laut dem Paper [7] bereits 132 VPs, die als LG Client-Instanzen verwendet werden können.

Die Ermittlung unentdeckter AS-Links aber ein unvollständig gelöstes Problem. Es wird weiterhin in diesem Bereich geforscht, um bisher nicht entdeckte ASes, AS-Links und IXPs ausfindig zu machen.

6. LITERATUR

- [1] B. Augustin, B. Krishnamurthy, and W. Willinger. IXPs: Mapped? In *Proceedings of the 9th ACM SIGCOMM Conference on Internet Measurement Conference, IMC '09*, pages 336–349, New York, NY, USA, 2009. ACM.
- [2] Bruno L., Graziano M., Balzarotti D., and Francillon A. Through the looking-glass and what eve found there. In *WOOT (2004)*.
- [3] CAIDA ARK. <http://www.caida.org/projects/ark>.
- [4] CAIDA Periscope API Dokumentation. <http://www.caida.org/tools/utilities/looking-glass-api/>.
- [5] CAIDA Periscope API neue Messung. <http://www.caida.org/tools/utilities/looking-glass-api/#create>.
- [6] Y.-J. Chi, R. Oliveira, and L. Zhang. Cyclops: The AS-level connectivity observatory. *SIGCOMM Comput. Commun. Rev.*, 38(5):5–16, Sept. 2008.
- [7] V. Giotsas, A. Dhamdhere, and k. claffy. Periscope: Unifying Looking Glass Querying. In *Passive and Active Network Measurement Conference (PAM)*, Mar 2016.
- [8] Y. He, G. Siganos, M. Faloutsos, and S. Krishnamurthy. Lord of the links: A framework for discovering missing links in the internet topology. *IEEE/ACM Trans. Netw.*, 17(2):391–404, Apr. 2009.
- [9] Internet Routing Registry. <http://www.irr.net>.
- [10] Internet2. <http://ndb7.net.internet2.edu/bgp>.
- [11] P. M. Kevin Butler, Toni Farley and J. Rexford. A survey of BGP security issues and solutions. In *Proceedings of the IEEE, vol. 98, no. 1*, January 2010.
- [12] A. Khan, T. Kwon, H.-c. Kim, and Y. Choi. AS-level topology collection through looking glass servers. In *Proceedings of the 2013 Conference on Internet Measurement Conference, IMC '13*, pages 235–242, New York, NY, USA, 2013. ACM.
- [13] H. V. Madhyastha, T. Isdal, M. Piatek, C. Dixon, T. Anderson, A. Krishnamurthy, and A. Venkataramani. iplane: An information plane for distributed services. In *Proceedings of the 7th Symposium on Operating Systems Design and Implementation, OSDI '06*, pages 367–380, Berkeley, CA, USA, 2006. USENIX Association.
- [14] Netacuity. <http://www.digitalelement.com/solutions/>.
- [15] Packet Clearing House. <http://www.pch.net/resources/data.php>.
- [16] PeeringDB. <http://www.peeringdb.com>.
- [17] PlanetLab. <http://www.planetlab.org>.
- [18] RIPE Atlas rate. <http://atlas.ripe.net/docs/udm/#rate-limits>.
- [19] RIPE-RIS. <http://www.ripe.net/ris>.

- [20] RouteViews. <http://www.routeviews.org>.
- [21] Selenium browser automation suite.
<http://www.seleniumhq.org/>.
- [22] T. Kernen, "traceroute.org" 2008.
www.traceroute.org.
- [23] Y. T.Li and S. Hares". A border gateway protocol.
RFC 4271, January 2006.
- [24] UCLA IRL. <http://irl.cs.ucla.edu/topology>.
- [25] Version6 Looking Glass.
<https://github.com/Cougar/lg>.
- [26] B. Zhang, R. Liu, D. Massey, and L. Zhang. Collecting the internet AS-level topology. *SIGCOMM Comput. Commun. Rev.*, 35(1):53–61, Jan. 2005.