



ارائه شده توسط:

سایت ترجمه فا

مرجع جدیدترین مقالات ترجمه شده

از نشریات معتبر

متعادلسازی بار دینامیک با استفاده از شبکه های مبتنی بر نرم افزار

چکیده:

در عصر مدرن امروزی، استفاده از اینترنت به طور وحشتناکی افزایش یافته است. با این حساب، ترافیک شبکه بالای بوجود می آید که مستلزم خدمات بسیاری مانند DNC به منظور کنترل می باشد. برای حل این امر متعادلسازی بار استفاده شده است. ولیکن متعادل سازهای بار اختصاص یافته گران قیمت بوده و سریعا به یک نقطه ضعف منفرد و ازدحام تبدیل می شوند. ولیکن این امر می تواند برای کار بهتر بهبود یابد. در رویکرد من شبکه مبتنی بر نرم افزار که از پروتکل OpenFlow استفاده می کند اجرا گردید تا کارایی را بهبود بدهد. در Future Internet، شبکه مبتنی بر نرم افزار (SDN) به شکل یکی از نویدبخش ترین پارادایم ها دیده شده است. با استفاده از این تکنیک شبکه مستقیما قابل برنامه ریزی و سریع الانتقال می شود. در اینجا درخواست های http از سوی کلاینت های مختلف به سرورهای http از قبل تعیین شده مختلف براساس الگوریتم زمانبندی نوبت گردشی هدایت خواهد شد. الگوریتم زمانبندی نوبت گردشی راحت اجرا شده و به بهترین نحوی در سرورهای وب با توزیع جغرافیایی استفاده می شود.

کلیدواژه ها: متعادلسازی بار، SDN، الگوریتم OpenFlow، الگوریتم نوبت گردشی.

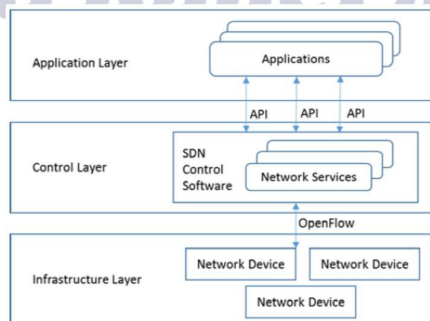
1-مقدمه

متعادلسازی بار یک فناوری برجسته است و می تواند به صرفه جویی در برق و بهبود استفاده از منابع در شبکه کمک کند. فراساختار شبکه کنونی متکی بر یک معماری عمودی مبتنی بر انواع نرم افزار و سخت افزار می باشد، با این حساب ایجاد یک سناریو با اندک انعطاف می نماید. چنین عدم انعطاف پذیری اثرات نامطلوبی را اخیرا تحت عنوان جزمیت اینترنت بوجود آورده است. SDN یک پارادایم جدید است که انسجام عمودی میان صفحه کنترل شبکه و صفحه داده هایش را نقض می کند. کاربران SDN می توانند برنامه های اپلیکیشن را به طور ترکیبی در بالای کنترل کننده اجرا کنند تا بر کل شبکه به حالت متمرکز و در زمان واقعی نظارت و مدیریت داشته باشند.

کنترل کننده های SDN تصمیم می گیرند که بسته چگونه مدیریت شود یا اینکه کار باید به نحوی اجرا بشود که بسته حذف شده یا بتواند به یک ورود جریان جدید برای فوروارده سازی بسته های مشابه در آینده افزوده بشود. یک تکنیک متعادل سازی بار معمولی قرار است از یک متعادل ساز کننده بار اختصاصی استفاده کند تا درخواست های کلاینت را به سرورهای مختلف فوروارد سازد، این تکنیک مستلزم پشتیبانی سخت افزاری است که گران قیمت است، فاقد انعطاف پذیری است و به سهولت به یک نقطه شکست منفرد تبدیل می شود. در اینجا ما درباره اجرای یک الگوریتم متعادل سازی کننده بار پویا برای توزیع انواع جریانات ترافیک منتقله در یک شبکه از طریق مسیره های موازی مختلف میان منبع و مقصد توضیحاتی می دهیم. الگوریتم OpenFlow متداولترین پروتکل استفاده شده در شبکه های SDN است که برای تبادول کنترل کننده با کلیه عناصر شبکه استفاده می شود.

1-1- شبکه های مبتنی بر نرم افزار

سیستم شبکه سنتی و معمولی دارای صفحه کنترل و صفحه داده ها با همدیگر است. در صورتیکه SDN برای ساخت یک شبکه کامپیوتری رهیافت زده است که شبکه را به صفحه کنترل و صفحه داده ها مجزاسازی و انتزاعی می کند. صفحه داده یک عملیات انتقال بسته ها از طریق شبکه را انجام می دهد. برخلاف شبکه های سنتی و قدیمی، کلیدهای زیربنایی اجرای صفحه کنترل را انجام نمی دهند. صفحه کنترل به همراه هوشمندی اش قادر به ساخت صفحه های داده ها روی شبکه است.



شکل 1- معماری SDN

صفحه کنترل یک نرم افزار یا هویت منطقی است که کلیه تصمیمات مسیره های را که توسط صفحه داده ها گرفته شده است پردازش می نماید. باینحساب، شبکه مستقیماً قابل برنامه ریزی و سریع الانتقال است. شکل 1 نشان دهنده معماری پایه SDN می باشد. جداسازی منطق اپلیکیشن از بستر فوروارده سازی تا حد زیادی باعث تسهیل

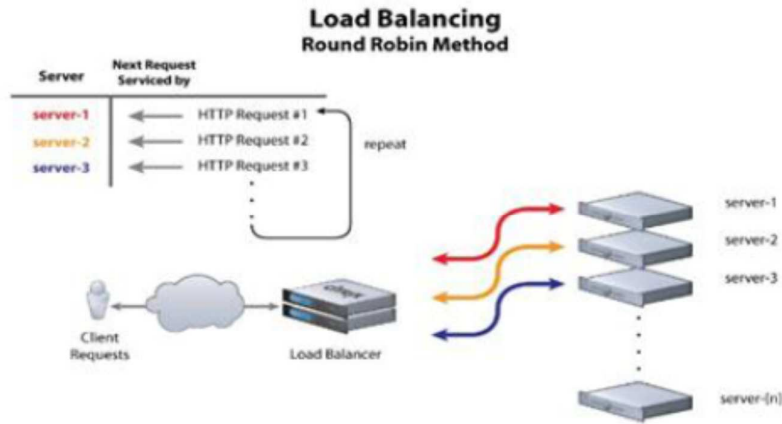
استقرار و گرداندن خدمات جدید می شود و باعث می شود که SDN به طور برجسته ای به تقاضاهای شبکه در حال تغییر و اصلاحات توپولوژی سوبسترا واکنش نشان بدهد. OpenFlow متداولترین پروتکل استفاده شده در شبکه های SDN است که برای تبادل کنترل کننده با کلیه عناصر شبکه استفاده می شود.

2-1- مسيردهی قدیمی در مقابل مسيردهی SDN

یک شبکه OpenFlow توسط یک یا چند کلید OpenFlow تشکیل شده است که حاوی یک جدول جریان با ورود جریانهای تعریف شده و اقدامات اجرا شده می باشد. تفاوت اصلی بین طراحی شبکه توزیع شده قدیمی و یک شبکه OpenFlow در ساختار شبکه صفحه کنترل نهفته است. برای معماری توزیع شده قدیمی مانند پروتکل دروازه مرزی (Border Gateway Protocol) یا BGP، اولین مسیر کوتاه باز (Open Short Path First) یا OSPF و سیستم حدواسط برای سیستم حدواسط (Intermediate System to Intermediate System) یا IS-IS، توپولوژی کنترل به شکل همتا به همتا می باشد. این امر بدان معناست که هر وسیله فرورارسازی به رویدادهایی از همتایانش گوش می دهد و تصمیمات مستقلی را براساس دیدگاه محلی اتخاذ می کند.

3-1- متعادلسازی بار

متعادلسازی بار می تواند نه تنها توسط نرم افزار یا پروتکل تخصصی بلکه با نصب دروازه تخصصی یا متعادلسازی کننده های بار حاصل بشود. اولین فناوری متعادلسازی کننده بار همان DNS می باشد. متعادلسازی کننده بار DNS یک روش ساده و موثر می باشد ولیکن امکان اینکه بین سایر سرورها را تشخیص بدهد وجود ندارد. وقتی یک سرور خراب می شود به زمان بیشتری برای رفرش شدن جهت کار مجدد و طبیعی نیاز دارد. کارهای نوبت گردشی با پاسخ دهی به درخواست های DNS نه تنها با یک نشانی IP منفرد بلکه با یک فهرست از نشانی های IP سایر سرورهاست که خدمات یکسانی را میزبانی می کند. نشانی های IP از فهرست براساس حالت نوبت گردشی باز می گردد. سرورهای وب با توزیع جغرافیایی به بهترین نحوی با بکارگیری توزیع محتوایی سرور نوبت گردشی متعادلسازی کننده بار DNS خدمات رسانی می شوند.



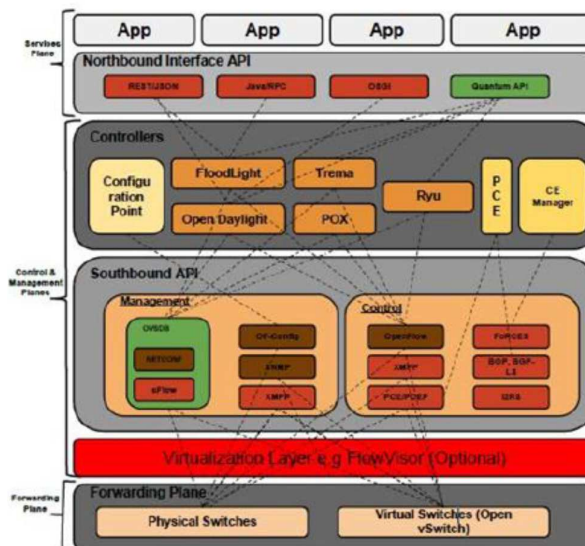
شکل 2- متعادلسازی کننده بار

در متعادلسازی کننده بار دینامیک و پویا، سنجش‌ها براساس نظارت مداوم سرورها می‌باشد و از اینرو به طور مداوم تغییر می‌کند. این روش ارتباطات را براساس جنبه‌های مختلف تحلیل عملکرد سرور زمان واقعی مانند تعداد کنونی اتصالات و ارتباطات به ازای هر گره یا سریعترین زمان پاسخ دهی گره توزیع می‌کند.

2-زمینه فنی

2-1- کنترل کننده های SDN

پنج تا از کنترل کننده های منبع باز ارشد از لحاظ کاربرد آنها در اینجا آنالیز شده اند: یعنی POX و Ryu و Trema و FloodLight و OpenDaylight. برای درک مفهوم SDN باید یک کنترل کننده مناسبی را انتخاب کنیم.



شکل 3- سطح مشترک کنترل کننده های SDN

در اینجا ما کنترل کننده POX را برای اپلیکیشن خود انتخاب کرده ایم. این اپلیکیشن دارای عملکرد موثری برای توسعه سریع و پیش نمونه سازی نرم افزار کنترل شبکه است. POX یک سطح مشترک Pythonic OpenFlow دارد. این سطح مشترک برای تعداد زیادی تحقیقات در دانشگاه ها استفاده شده است چرا که برای برنامه ریزی خیلی آسان است. این سطح مشترک از OpenFlow v1.0 پشتیبانی می کند. این مورد می تواند هر جایی اجرا بشود و می تواند با زمان اجرای خارج از نصب PyPy برای سهولت استقرار دسته بندی بشود. همچنین برای بررسی توزیع پیش نمونه، از بین بردن عیوب برنامه SDN، مجازی سازی شبکه، طراحی کنترل کننده و مدل های برنامه ریزی استفاده می شود.

2-2- الگوریتم OpenFlow

الگوریتم OpenFlow یک مکانیسمی را برای SDN فراهم می سازد. ضمن اینکه OpenFlow اولین بار به عنوان شیوه ای برای توانمندسازی محققان برای اجرای آزمایشات در شبکه های دانشگاهی مطرح گردید، مزیت های آن باعث استفاده آنها فراتر از آن مورد شده است. مدل کنترل مرکزی OpenFlow می تواند از نیاز به ساخت سیاستگزاری های عام از پیکربندی های کلید به کلید اجتناب ورزد، و می تواند همچنین از مدیریت ترافیک نزدیک به بهینه هم پشتیبانی کند. رفتار یک وسیله فورواردهای می تواند در دو مرحله خلاصه سازی گردد: (1) وقتی که یک بسته ای را دریافت می کند که با یک ورود خاصی در جدول مسیرهدهی اش تطابق ندارد، با کنترل کننده تماس می گیرد تا تعریف کند که چگونه بسته باید فوروارده شود یا دور انداخته شود (2) وقتی بسته دریافتی با قانون جدول مسیرهدهی آن تطابق داشته باشد، عملکرد مربوطه در جدول فورواردهای اجرا می شود.

3-2- Mininet

Mininet یک شبکه مجازی واقع گرایانه را ایجاد می کند و هسته واقعی، کلید و کد اپلیکیشن را روی یک ماشین منفرد (VM، ابری یا منفی) در عرض چند ثانیه با یک فرمان منفرد اجرا می کند. Mininet همچنین یک راه عالی برای ایجاد، اشتراک گذاری، و آزمایش با سیستم های شبکه سازی مبتنی بر نرم افزار و OpenFlow می باشد. شکل 4 نشان دهنده ایجاد توپولوژی معمول می باشد.

```

mininet@mininet-vm:~$ sudo mn --top single,6 --nec --arp --controller=remote
*** Creating network
*** Adding controller
Unable to contact the remote controller at 127.0.0.1:6633
*** Adding hosts:
h1 h2 h3 h4 h5 h6
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1) (h3, s1) (h4, s1) (h5, s1) (h6, s1)
*** Configuring hosts
h1 h2 h3 h4 h5 h6
*** Starting controller
*** Starting 1 switches
s1
*** Starting CL:
mininet> []

```

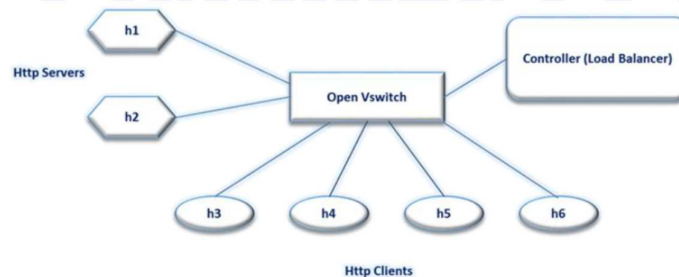
شکل 4- خلق توپولوژی Mininet

3- اجرا و ارزیابی

```

1: // N is the size of circular ACK re-sequencing buffer
2: wSeqNum = (cumulative ACK sequence number) mod N
3: if wSeqNum > left then
4:   left = wSeqNum // in-sequence, new ACK
5:   put ACK at the end of the re-sequencing buffer
6:   release held ACK(s) up to left point with a peak rate
7: end if
8: if wSeqNum == left then
9:   switch RPC state // duplicate ACK
10:    case NORMAL // re-sequencing
11:      hold ACK with an identified number, break
12:    case LOSS // no re-sequencing
13:      release ACK to a sender, break
14:   end switch
15: end if

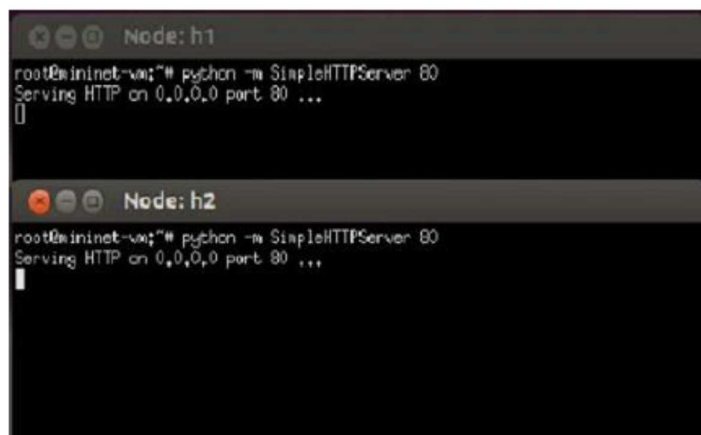
```



شکل 5- توپولوژی معمول

شکل 5 نشان دهنده توپولوژی شبکه است که ما برای ارزیابی اتخاذ کرده ایم. کنترل کننده POX برای اجرای متعادل سازی بار استفاده می شود. Mininet برای ایجاد یک توپولوژی شبکه مجازی استفاده می شود. در اینجا ما یک توپولوژی را با کمک 6 گره های میزبان در نظر گرفته ایم. جایی که دو گره به شکل سرور http بکار برده می شوند 4 تای دیگر به شکل کلاینت های http می باشند. بسته به رسیدن ترافیک به سرور از گره های کلاینت،

سرور زمانبندی می شود. کلاینت ها از طریق یک نشانی IP عمومی منفرد که از طریق یک کلید دروازه قابل دسترسی است، به خدمات دسترسی دارند.

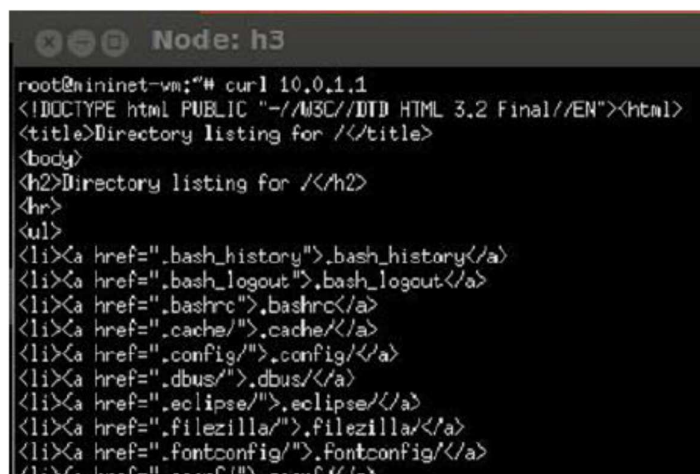


```
Node: h1
root@mininet-vm:~# python -m SimpleHTTPServer 80
Serving HTTP on 0.0.0.0 port 80 ...

Node: h2
root@mininet-vm:~# python -m SimpleHTTPServer 80
Serving HTTP on 0.0.0.0 port 80 ...
```

شکل 6-تنظیم سرور

کلید متعادلسازی کننده بار به بازنویسی نشانی IP مقصد برای هر بسته کلاینت آمده برای نشانی نسخه اختصاص داده شده می پردازد. سیاستگزاری نوبت گردشی از یک صف دایره ای برای گرفتن این تصمیم استفاده می کند که در کجا یک تقاضا ارسال بشود. سیاستگزاری مبتنی بر بار یک درخواستی را به سرور با پایین ترین بار ارسال می کند، که طی آن بار به صورت تعداد درخواست های معوقه تعریف می شود.



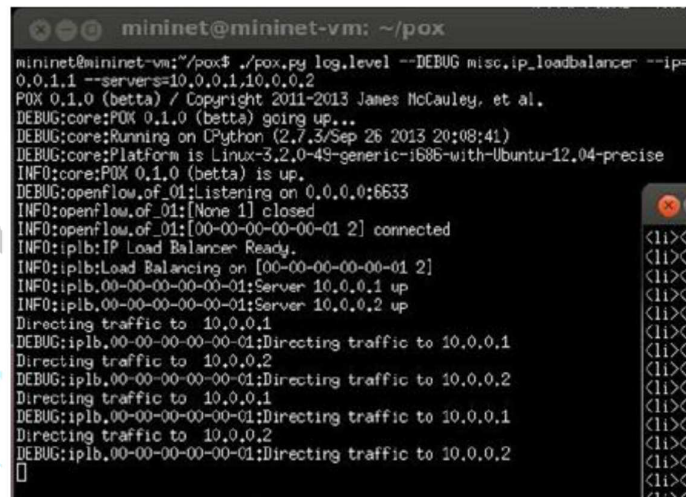
```
Node: h3
root@mininet-vm:~# curl 10.0.1.1
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 3.2 Final//EN"><html>
<title>Directory listing for /</title>
<body>
<h2>Directory listing for /</h2>
<hr>
<ul>
<li><a href=".bash_history">.bash_history</a>
<li><a href=".bash_logout">.bash_logout</a>
<li><a href=".bashrc">.bashrc</a>
<li><a href=".cache/">.cache/</a>
<li><a href=".config/">.config/</a>
<li><a href=".dbus/">.dbus/</a>
<li><a href=".eclipse/">.eclipse/</a>
<li><a href=".filezilla/">.filezilla/</a>
<li><a href=".fontconfig/">.fontconfig/</a>
<li><a href=".gnome/">.gnome/</a>
```

شکل 7-ترافیک از گره کلاینت

بعد از ایجاد توپولوژی معمول با استفاده از Mininet، سپس سرورها باید تنظیم بشوند. در گره 1 و 2، سرور HTTPServer با تعداد پورت 80 عدد ساخته می شود. سرورها باید با نشانی IP منفرد تنظیم بشوند. این امر در

شکل 6 نشان داده شده است. IPreg یک ابزار شبکه مشهور است که برای اندازه گیری عملکرد پهنای باند TCP و UDP وجود آمد. کاربر باید یک تعداد از مونهایی را اجرا کند که یک بینشی را به موجودیت پهنای باند شبکه، از دست رفتن داده ها، تاخیر و jitter فراهم می سازد.

بعد با استفاده از فرمان curl ترافیک به سرور ارسال می شود. فرمان curl از کلید چهار گره HTTPClient استفاده می کند. شکل 7 نشان دهنده کلاینت هایی است که ترافیک را به سرورها ارسال می کند. این امر صفحه وب را از IP سرور کسب می کند. با این حساب با استفاده از الگوریتم نوبت گردشی کلاینت سرور خود را به حالت گردشی کسب می کند. شکل 8 نشان دهنده هدایت کردن ترافیک از گره های مختلف به سرورها می باشد. همگی گره های کلاینت درخواستی را به سرور داده اند. سرور شروع به هدایت کردن ترافیک به صورت ترتیبی می نماید.



```
mininet@mininet-vm: ~/pox
mininet@mininet-vm:~/pox$ ./pox.py log.level --DEBUG misc.ip_loadbalancer --ip=1
0.0.1.1 --servers=10.0.0.1,10.0.0.2
POX 0.1.0 (beta) / Copyright 2011-2013 James McCauley, et al.
DEBUG:core:POX 0.1.0 (beta) going up...
DEBUG:core:Running on CPython (2.7.3/Sep 26 2013 20:08:41)
DEBUG:core:Platform is Linux-3.2.0-49-generic-i686-with-Ubuntu-12.04-precise
INFO:core:POX 0.1.0 (beta) is up.
DEBUG:openflow.of_01:listening on 0.0.0.0:6633
INFO:openflow.of_01:[None 1] closed
INFO:openflow.of_01:[00-00-00-00-00-01 2] connected
INFO:iplb:IP Load Balancer Ready.
INFO:iplb:Load Balancing on [00-00-00-00-00-01 2]
INFO:iplb.00-00-00-00-00-01:Server 10.0.0.1 up
INFO:iplb.00-00-00-00-00-01:Server 10.0.0.2 up
Directing traffic to 10.0.0.1
DEBUG:iplb.00-00-00-00-00-01:Directing traffic to 10.0.0.1
Directing traffic to 10.0.0.2
DEBUG:iplb.00-00-00-00-00-01:Directing traffic to 10.0.0.2
Directing traffic to 10.0.0.1
DEBUG:iplb.00-00-00-00-00-01:Directing traffic to 10.0.0.1
Directing traffic to 10.0.0.2
DEBUG:iplb.00-00-00-00-00-01:Directing traffic to 10.0.0.2
```

شکل 8-هدایت ترافیک

4-نتیجه گیری و کار آتی

صرفنظر از پیشرفتهای انجام گرفته در صنعت IT ، شبکه های مبتنی بر نرم افزار یک پارادایم محاسباتی در حال تکامل است که همه هویت های دیگر را در کلید بخشهای خصوصی تحت تاثیر قرار داده است. الگوریتم متعادلسازی بار DNS نوبت گردشی بهترین کار را برای خدمات به دلیل تعداد زیادی ارتباطات توزیع شده یکنواخت به سرورهای دارای ظرفیت متعادلسازی شده انجام می دهد. این پروژه کار را با بررسی درباره الگوریتم های متعادلسازی کننده بار مرتبط تر و الگوریتم های چندمنظوره برای SDN بیشتر توسعه داده است. نیز در آینده الگوریتم متعادلسازی کننده بار می تواند با در نظر گیری پارامترهای بیشتر ارزیابی بشود.

5. REFERENCES

- [1] Yuanhao Zhou, Li Ruan, Limin Xiao, Rui Liu. 2014 A Method for Load Balancing based on Software Defined Network. School of Computer Science and Engineering Beihang University, Beijing, China.
- [2] Richard Wang, Dana Butnariu, and Jennifer Rexford. OpenFlow-Based Server Load Balancing Gone Wild. Princeton University; Princeton, NJ
- [3] Marti Boda Navarro, 2014. Dynamic Load Balancing in Software Defined Networks. Aalborg University.
- [4] Zdravko Bozakov and Amr Rizk, 2013 Taming SDN Controllers in Heterogeneous Hardware Environments. Leibniz Universit at Hannover, Germany.
- [5] Yannan Hu, Wendong Wang, Xiangyang Gong, Xirong Que, Shiduan Cheng, 2012. Balanceflow: Controller Load Balancing For Openflow Networks. Beijing University of Posts and Telecommunications, Beijing, 100876, China.
- [6] Jehn-Ruey Jiang, Widhi Yahya and Mahardeka Tri Ananta. 2011. Load Balancing and Multicasting Using the Extended Dijkstra's Algorithm in Software Defined Networking. National Central University Zhongli City, Taiwan.
- [7] Junjie Zhang; Kang Xi; Min Luo; Chao, H.J. Load balancing for multiple traffic matrices using SDN hybrid routing.
- [8] Rahamatullah Khondoker, Adel Zaalouk, Ronald Marx, Kpatcha Bayarou. Feature-based Comparison and Selection of Software Defined Networking (SDN) Controllers. Fraunhofer Institute for Secure Information Technology Rheinstr. 75, Darmstadt, Germany.
- [9] Felipe Alencar, Marcelo Santos, Matheus Santana, Stenio Fernandes, How Software Aging Affects SDN: A View on the Controllers. Universidade Federal de Pernambuco (UFPE) Recife, Brazil.
- [10] G. Araniti+, J. Cosmas_, A. Iera+, A. Molinaro+, R. Morabito+, A. Orsino+. OpenFlow over Wireless Networks: Performance Analysis. University Mediterranea of Reggio Calabria, Italy.
- [11] Zhihao Shang, Wenbo Chen, Qiang Ma, Bin WU. Design and implementation of server cluster dynamic load balancing based on OpenFlow. Lanzhou University Communication Network Center Lanzhou, China.



برای خرید فرمت ورد این ترجمه، بدون واتر مارک، اینجا کلیک نمایید.



این مقاله، از سری مقالات ترجمه شده رایگان سایت ترجمه فا میباشد که با فرمت PDF در اختیار شما عزیزان قرار گرفته است. در صورت تمایل میتوانید با کلیک بر روی دکمه های زیر از سایر مقالات نیز استفاده نمایید:

لیست مقالات ترجمه شده ✓

لیست مقالات ترجمه شده رایگان ✓

لیست جدیدترین مقالات انگلیسی ISI ✓

سایت ترجمه فا ؛ مرجع جدیدترین مقالات ترجمه شده از نشریات معتبر خارجی