



دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)

به روز رسانی یکپارچه افزایشی در شبکه های نرم افزار بنیان

روش تحقیق و گزارش نویسی
دکتر شیری

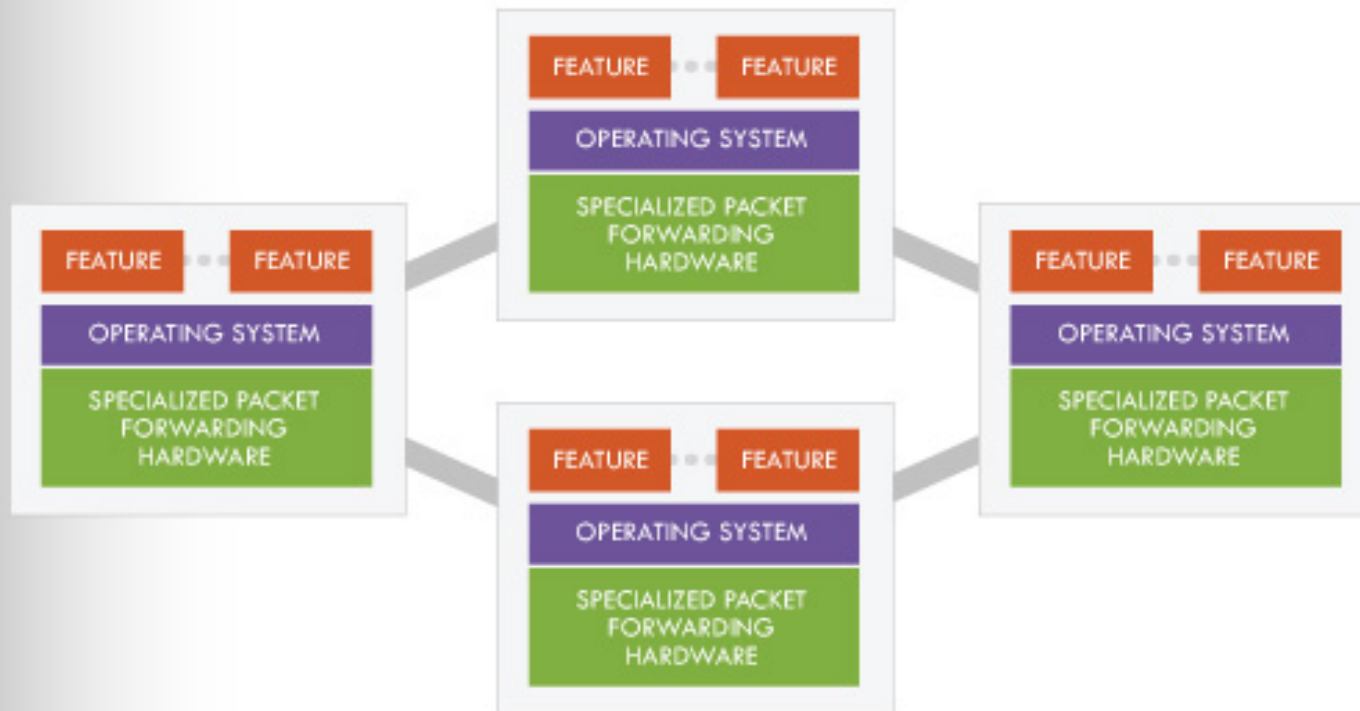
الهه جلال پور

فهرست

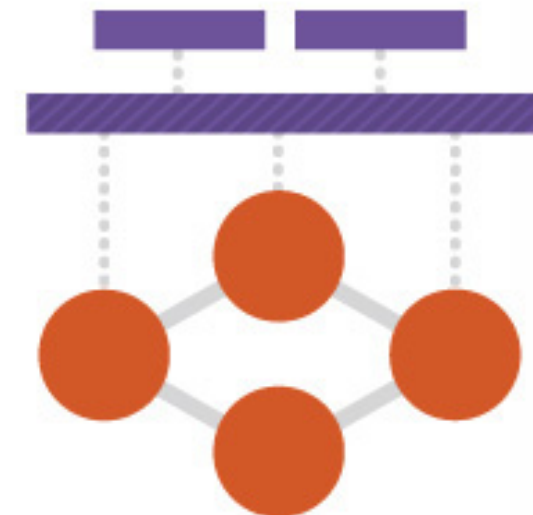
- مقدمه
- شبکه های نرم افزار بنیان (SDN)
- سیاست در شبکه
- به روز رسانی یکپارچه
- تعریف مسئله
- مشکل فضای قانون
- کار های قبلی
- راه حل McGeer
- تئوری کار
- الگوریتم کلی کار
- آزمایش ها
- نتیجه گیری
- منابع

شبکه های نرم افزار بنیان (SDN)

OpenFlow/SDN Difference



Network of vertically integrated,
closed, proprietary switches



OpenFlow/SDN:

- Separation of control and data plane
- Open interface between control and data plane
- Open interface to the control plane
- Network control and management features in software

مقدمه

سیاست در شبکه

- یک سیاست مجموعه ای از قوانین R است که هر کدام یک ۲ تایی مرتب به شکل (P, a, z) هستند.

- P یک گزاره، a یک کنش و z یک اولویت را نشان می دهد.

- P دسته ای از بسته ها را بر اساس سرآیند آنها و مکان آنها (سوئیچ و درگاه ای که به آن رسیده اند) مشخص می کند. برای مثال گزاره ی $(srcIP=10.0.0.1 \& switch=3 \& inport=7)$ بسته هایی را که از آدرس IP مبدا 10.0.0.1، از طریق درگاه 7، به سوئیچ 3 آمده اند را توصیف می کند.

به روز رسانی یکپارچه

یکپارچگی به روز رسانی

- یک مکانیزم قدرتمند برای SDN است.
- کار آن ایجاد اطمینان برای حفظ ویژگی های کلیدی در هنگام عوض شدن حالات سیستم است.
- اگر سیاست شبکه در حال تعویض از وضعیت A به وضعیت B باشد، به روز رسانی یکپارچه تضمین می کند که تمام بسته ها برای عبور از شبکه تنها از سیاست A و یا از سیاست B پیروی کنند.

تعریف مسئله

- شبکه‌ها پیوسته در حال تغییر هستند. اپراتورها به صورت پیوسته در حال تغییر دادن جدول‌های مسیر یابی، تنظیم کردن وزن لینک‌ها، تغییر دادن لیست‌های کنترل دسترسی برای مهندسی ترافیک، اصلاح آسیب‌پذیری‌های امنیتی و غیره هستند.
- با اینکه به روز رسانی‌ها در ابتدا به درستی برنامه‌ریزی می‌شوند پیاده‌سازی آنها مشکل‌است و می‌تواند منجر به شکستگی‌هایی مانند خطاهای گذرا، از دست دادن ارتباط با سرور، آسیب‌پذیری‌های امنیتی پیش‌بینی نشده، وقفه‌هایی در تماس‌های VoIP شوند.

* راه حل:

- در شبکه‌های سابق :
پژوهشگران تعداد زیادی تعمیم برای پروتکل‌های موجود به قصد جلوگیری از این رفتارهای خلاف قاعده ارائه داده‌اند.
- در SDN :
فرصت بزرگی برای تولید تجردی کلی برای مدیریت به روز رسانی شبکه فراهم می‌شود.

تعریف مسئله

مشکل فضای قانون

- راه کلی برای پیاده سازی به روز رسانی یکپارچه از یک عمل دو مرحله ای استفاده می کند. برای آماده سازی:

* بسته ها بر اساس شماره نسخه ۱ یا ۲ برچسب زده میشوند.

* پیاده شدن قانونی که با آن بسته تطابق داشته و به سیاستی که با شماره نسخه آن بسته هماهنگی داشته باشد روی بسته ها در مرکز شبکه

* حذف شماره نسخه از بسته ها در سوئیچ خروجی

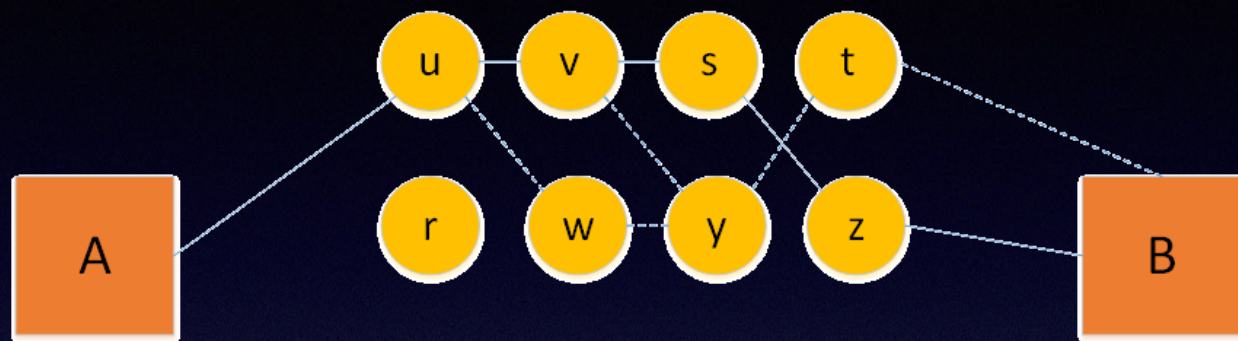
- اشکال عملی قابل توجه:

* نیاز به فضای قانون دو برابر در سوئیچ ها

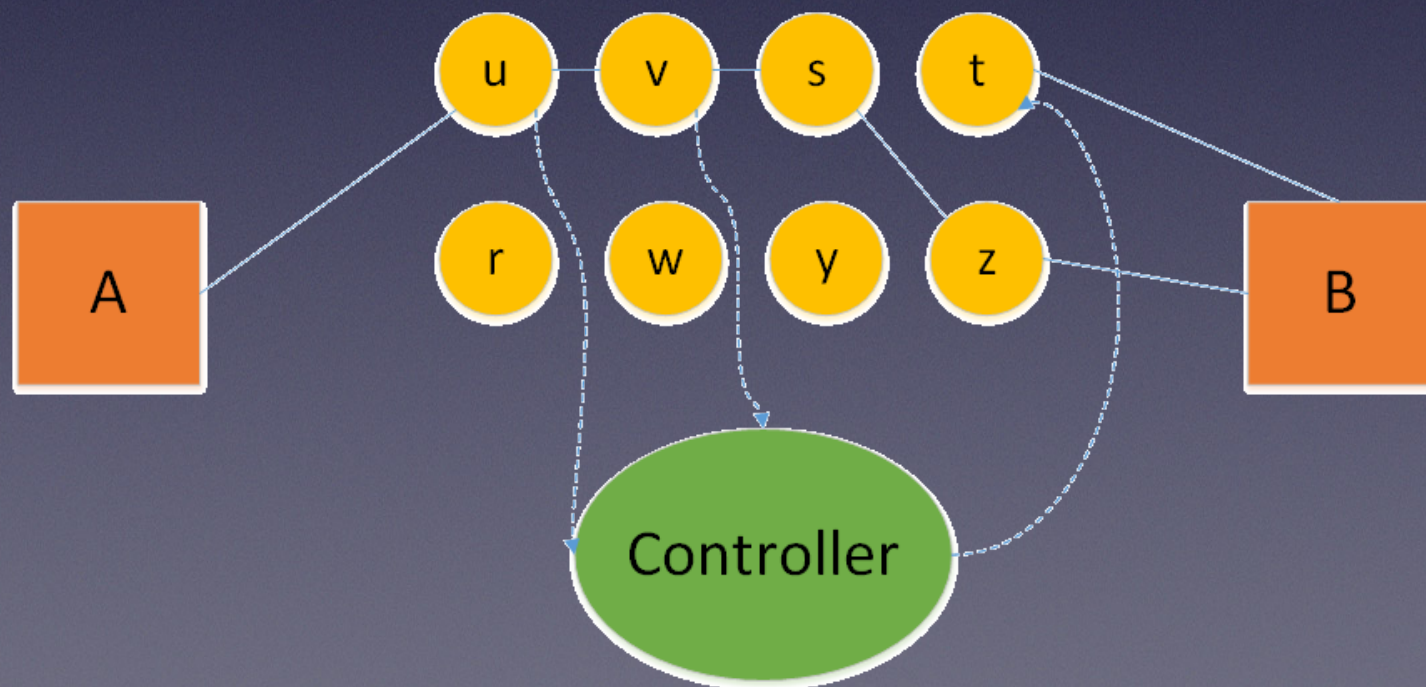
* گران و پر مصرف بودن این فضای قانون (TCAM)

کارهای قبلی

McGeer حل



در راه حل او ترافیک به سمت کنترلر منحرف شده که با سربار قابل توجه و محدودیت کارایی پردازش بسته همراه است.



تئوری کار

در این مطالعه، ما:

- الگوریتمی را برای به روز رسانی یکپارچه معرفی میکنیم که با افزایش زمان اجرای به روز رسانی سربار فضای قانون را کاهش میدهد.
- هر به روز رسانی را به k مرحله تقسیم میکنیم که هر کدام قسمتی از ترافیک را به پیکربندی منتقل میکند. هرچقدر تعداد مراحل بیشتر شود، زمان به روز رسانی بیشتر شده و فضای قانون کوچکتر میشود.
- برای حفظ یکپارچگی، الگوریتم وابستگی بین سیاست های قدیم و جدید را آنالیز میکند تا مشخص شود که چه قوانینی نصب و چه قوانینی حذف شوند.

تئوری کار

به طور شهودی هر مرحله به صورت زیر کار میکند:

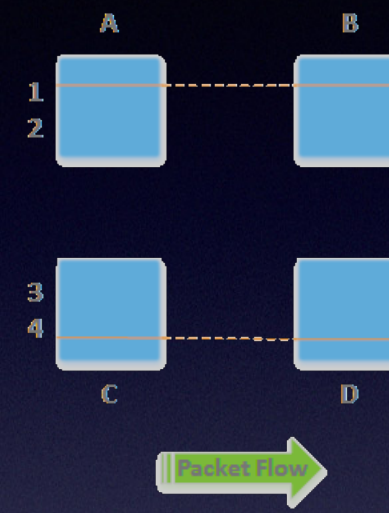
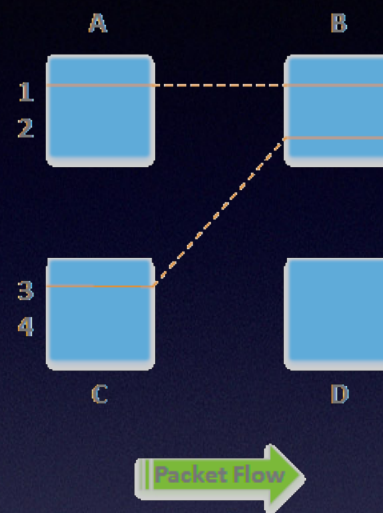
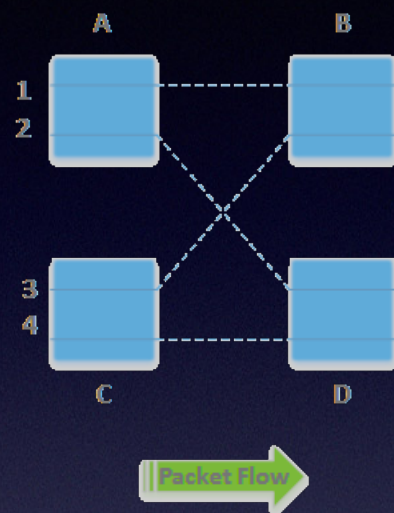
۱. انتخاب یک زیر مجموعه از جریان های بسته ها که سیاست جدید روی آنها پیاده شود.
۲. تعیین قوانینی که برای حفظ یکپارچگی با توجه به جریان در حال ورود باید به شبکه اعمال شوند.
۳. اعمال کردن این قوانین در شبکه با توجه به الگوریتم دو مرحله ای
۴. تعیین کردن قوانینی که بدون برهم زدن یکپارچگی می توانند به صورت امن حذف شوند.
۵. صبر تا زمانی که تمام بسته هایی که از سیاست قبلی پیروی میکنند از شبکه خارج شوند.
۶. حذف قوانینی که اجازه حذف آنها داده می شود.

الگوریتم کلی کار

```
procedure update(old:Policy, new:Policy, T:Topo) {  
  old_remains = old;  
  new_placed =  $\emptyset$ ;  
  packets = ingress;  
  while packets  $\neq \emptyset$  do  
    choose ingress predicate P;  
    // compute new rules to be added to switches  
    to_add = reachable_rules(P, new, T) - new_placed;  
    new_placed = new_placed  $\cup$  to_add;  
  
    // compute the old rules to be removed  
    packets = packets - P;  
    required = reachable_rules(packets, old, T);  
    to_remove = old_remains - required;  
    old_remains = required;  
  
    // perform the slice update consistently  
    two_phase_commit(to_add, to_remove);  
  return;  
}
```


بهینه سازی

انتخاب زیر مجموعه ها به صورت بهینه



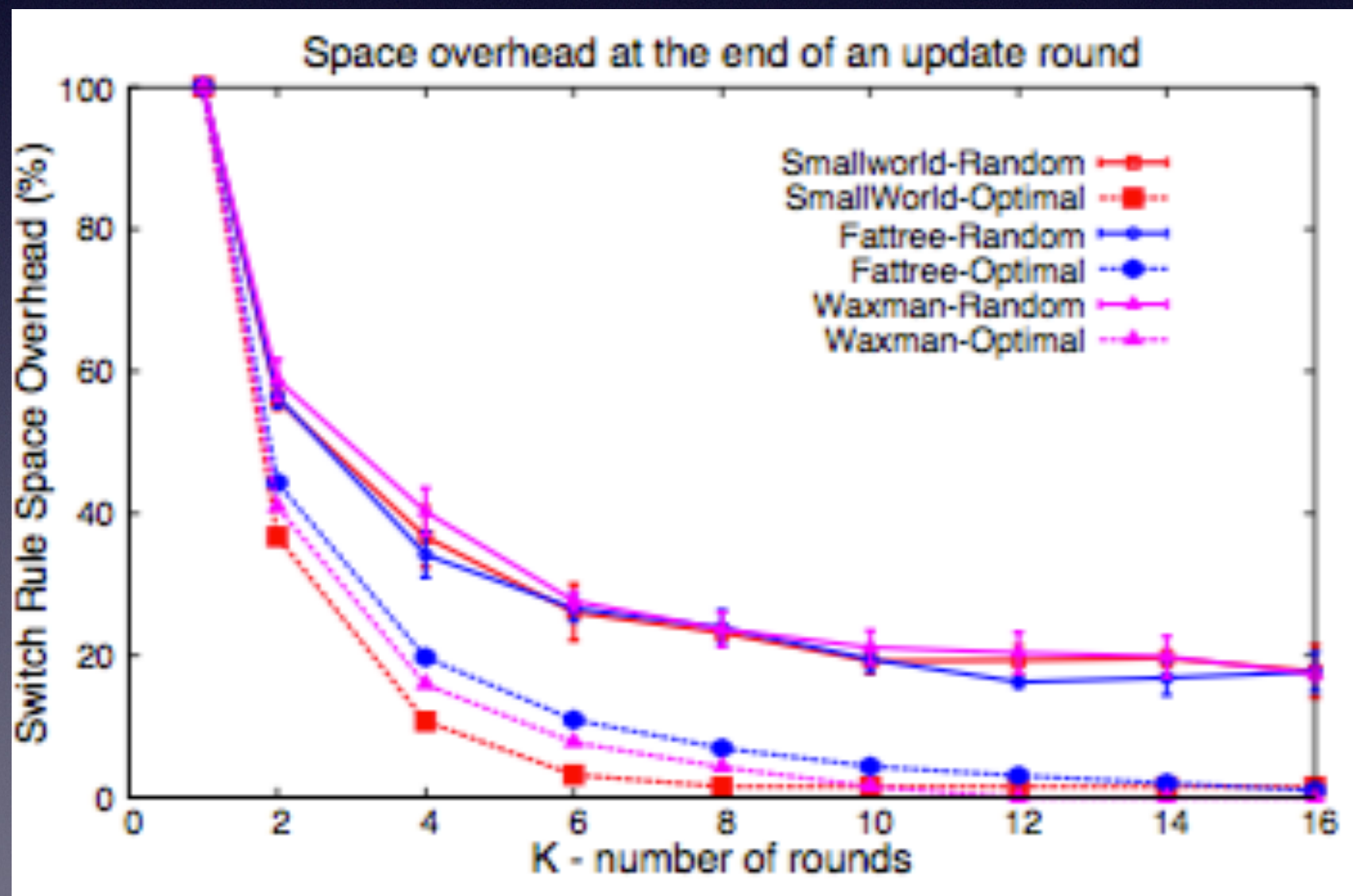
در شکل سمت چپ ۴ سوئیچ با نام های A، B، C، D نشان داده شده اند که در نهایت در هر کدام از آنها ۲ قانون نشان داده شده باید نصب شوند.

شکل میانی نشان می دهد که اگر ما بسته هایی که به قانون های ۱ و ۳ مربوط می شوند را در ابتدا اعمال کنیم سربار روی سوئیچ B زیاد خواهد شد.

اگر به جای آن بسته هایی که مربوط به قانون های ۱ و ۴ می شوند را اعمال کرده بودیم از سرریز بار در B جلوگیری کرده بودیم.

آزمایش ها

الگوریتم مطرح شده به زبان python بر روی کنترلر NOX اجرا شده است و با استفاده از نرم افزار mininet بر روی سه توپولوژی استاندارد Fattree, Smallworld و Wax-man با ۲۴ سوئیچ و ۵۷۶ host مورد آزمایش قرار گرفته است.



در به روز رسانی افزایشی تصادفی به هر یک از K مرحله زیر مجموعه ای را که N/K از کل سیاست است اختصاص می دهد و سپس یکپارچه بودن را آنالیز می کند.

نتیجه گیری

- در این مطالعه الگوریتمی مطرح شد که تعادلی را بین زمان انجام شدن به روز رسانی و سربار فضای قانون برقرار می کند.

- این الگوریتم با تقسیم کردن سیاست کلی به زیر مجموعه های کوچک و حرکت از سیاست قبلی به سیاست جدید، با اجرا کردن هر کدام از این زیر مجموعه ها و حذف زیر مجموعه های بی استفاده از سیاست قبلی به صورتی کاملاً بهینه این به روز رسانی را انجام می دهد.

- با زیاد شدن تعداد زیر مجموعه ها فضای قانون به طرز قابل توجهی کاهش میابد.

منابع

- [1] N. P. Katta, J. Rexford, and D. Walker. Incremental Consistent Updates. In *HotSDN*, 2013.
- [2] M. Reitblatt, N. Foster, J. Rexford, C. Schlesinger, and D. Walker. Abstractions for network update. In *ACM SIGCOMM*, Aug. 2012.
- [3] R. Mahajan, and R. Wattenhofer. On Consistent Updates in Software Defined Networks. In *HotNets*, 2013.
- [4] M. Reitblatt, N. Foster, J. Rexford, and D. Walker. Consistent updates for software-defined networks: Change you can believe in!. In *HotNets*, Nov 2011.
- [5] P. Peres, M. Kuz, M. Canini, D. Kostic. ESPRES: Easy Scheduling and Prioritization for SDN. 2014
- [6] R. McGeer, A safe. Efficient update protocol for OpenFlow networks. In *HotSDN*, Aug. 2012.
- [8] S. Raza, Y. Zhu, and C.-N. Chuah. Graceful network state migrations. In *IEEE/ACM Trans. on Networking*, vol. 19, Aug 2011.
- [9] J. P. John, E. Katz-Bassett, A. Krishnamurthy, T. Anderson, and A. Venkataramani. Consensus routing: The Internet as a distributed system. In *NSDI*, Apr 2008.