# Software-Defined Inter-Domain Routing Revisited

Chen Chen*, Bo Li*, Dong Lin*, and Baochun Li+

*Department of Computer Science and Engineering, Hong Kong University of Science and Technology
+Department of Electrical and Computer Engineering, University of Toronto

***Abstract*—The decoupling of control and data plane in software-defined networking (SDN) has been shown to be promising to improve routing performance in the context of intra-domain routing. The applicability of SDN in inter-domain routing, especially with respect to route convergence, has not been properly explored. In this work, we propose a mathematical model to quantify the BGP convergence time for inter-domain routing by capturing only the essential components in BGP convergence process. Based on the model and some practical observations, we study how SDN may help to facilitate the inter-domain routing. We further present a greedy algorithm that selects Autonomous Systems (ASes) for incremental SDN deployment with the objectives of minimizing the BGP convergence time. The simulation result based on the real world Internet topology confirms the effectiveness of our proposed algorithm.**

## I. INTRODUCTION

Border Gateway Protocol (BGP) is the *de facto* inter-domain routing protocol of the Internet, and with it, newly generated routing information is spread in a peer-to-peer manner. One problem BGP has encountered, due to the decentralized nature, is its slow converging speed. According to measurement studies in [1], after a topology change, it takes an average of 3 minutes for the Internet to restore all the routing tables, with a worst case of 15 minutes. Such a delay may lead to excessive packet losses, and is thus unacceptable to many delay-sensitive applications. Therefore, speeding up BGP convergence is a critically important, yet challenging, problem.

Meanwhile, Software Defined Networking, or SDN, has been proposed as a potential direction for the Internet to evolve into. Its fundamental design principle is to decouple the control plane from the forwarding plane, so that networks can be managed in a programmable and centralized manner. Rather than relying on decentralized negotiations between routers, under SDN routing tables are instead computed by a centralized controller. The benefits of SDN, especially for intra-domain routing, have been extensively explored in previous works like RCP [2]. Particularly, Fu *et al.* demonstrated that intra-domain routing convergence can be accelerated by replacing the decentralized link-state routing protocols with a centralized control scheme [3].

Beyond the intra-domain scenario, there have been several attempts to apply SDN in inter-domain routing. In [4], Gupta *et al.* proposed SDX, a software defined exchange, in which the Internet Exchange Point (IXP) was considered as the starting point of adopting SDN principles to revolutionize the wide-area traffic delivery. More generally, in [5], Kotronis *et al.* examined, from both technical and financial perspectives, the incentives to adopt SDN principles in routing control logic of multiple ASes, in which BGP session messages were redirected to a central routing control platform to improve traffic engineering and inter-domain routing. More recently, [6] developed an inter-domain emulation framework, which incorporated a hybrid strategy involving both SDN and BGP. According to its experimental results, SDN was able to accelerate the convergence of inter-domain routing.

In this paper, we revisit the idea of applying software-defined networking principles to inter-domain routing, and investigate two important challenges that are still largely uncharted territories. First, how to quantitatively evaluate the performance benefit that software-defined networking may introduce to inter-domain routing, especially with respect to the convergence of routing tables? Second, as illustrated in [7], it is not feasible to globally integrate new technologies like SDN into current infrastructures; therefore, how shall we strategically select locations for such incremental deployment to take place?

To answer these questions, we first present a mathematical model for quantifying the routing improvement introduced by applying SDN, as well as for examining how and where SDN can be incrementally deployed in the Internet to achieve fast convergence. This is quite challenging because of the complexity of BGP; building a simple model to capture many fine details is not realistic, but those details do contribute to the overall convergence time. By carefully analyzing the BGP routing behavior, we build a model to capture the essential aspects of BGP, and to quantitatively estimate the convergence times with and without SDN. We then present a metric for identifying the most critical locations for deploying SDN with the objective of minimizing the global convergence time. Finally, based on our metric, we propose an algorithm to guide the incremental deployment of SDN. Simulations based on a real-world Internet topology have been conducted to evaluate the effectiveness of the algorithm.

The remainder of this paper is organized as follows. In Sec. II, we examine the BGP routing behavior and create a mathematical model that can be used to quantitatively compare the convergence times with and without SDN involved. In Sec. III, we discuss how and where SDN should be deployed over the Internet, based on our model and subsequent analysis. Our simulation results are presented in Sec. IV, and finally Sec. V concludes the paper.

978-1-4799-6664-6/16/$31.00 ©2016 IEEE

## II. MODEL

In this section, we discuss some simplifying assumptions for modeling the essential features in BGP, and then examine different types of route updates to quantitatively obtain the route convergence time. Finally we investigate the effect that SDN brings to the BGP routing convergence process.

### A. Assumptions

BGP is a path-vector routing protocol, in which the best paths to different destinations are acquired and disseminated through BGP's routing update messages. Once there is a topology change and some best paths are no longer valid, routers will select new best paths and announce them to others. Finally, after enough iterations of hop-by-hop negotiations between routers running BGP, all the involved routing tables would be updated to comply with the new topology.

Modeling BGP is a challenging task. First, now that there are more than $50,000$ ASes in the Internet [8], each containing a large number of network devices and endpoints, it is not realistic to capture the topological properties in a fine granularity. Additionally, in order to satisfy the scalability, security and flexibility requirements, BGP is designed to be quite complex, making it difficult to be modeled precisely. Fortunately, instead of obtaining the accurate BGP convergence time, our objective is, to merely examine how BGP convergence could be accelerated with the assistance of SDN. Hence, rather than building an exact model that is hardly achievable, we focus only on capturing the essential BGP components that contribute to the convergence time. To create such a model, we make a few simplifications as follows:

- *Topology*: We model an AS as a node, with the physical connection between two ASes as a link. Similar simplifying approaches have been adopted in [1] [9]. Admittedly, an AS may consist of many border and internal routers, within which there might be various protocols, such as IBGP and interior routing protocols like OSPF. Nevertheless, based on the results in [1], these protocols do not actually contribute much to the overall route convergence time. Therefore, we do not include such details within an AS in our model.
- *Policy*: Although BGP allows the administrator of an AS to arbitrarily specify the policies, the default routing policy used by most ISPs, according to [1], is simply to select the path with the shortest length, which will be considered in our model. Additionally, when two paths from different neighbor ASes have the same path length, the path from the neighbor AS with a smaller ID is preferred. Notice, however, that our proposed model can be easily extended to include other policies.
- *One-hop delay*: Based on the above topological simplification, route update messages are disseminated from an AS to its neighboring ASes hop by hop. In practice, the one-hop delay can vary a great deal, due to the fluctuating propagation delay between ASes, plus the unfixed processing time at BGP routers. Here, for simplification, we assume an identical one-hop delay across the network. Based on this assumption, the time needed for propagating route update messages from one AS to another would be proportional to the number of hops between them.

With the simplifications above, we next describe the model for analyzing the BGP convergence process.

### B. BGP Convergence Model

We model the Internet as an undirected graph $G = (V, E)$, where $V$ is the set of nodes representing ASes, and $E$ is the set of links. A path from the source AS to the destination AS consists of a sequence of links from $E$, its length determined by the number of links. According to the taxonomies in [1], there are four types of routing events:

- *Tup*: A previously unavailable destination is announced as available.
- *Tshort*: An active path is replaced by a preferable one.
- *Tdown*: A previously available destination is announced as unavailable.
- *Tlong*: An active path is replaced by a less preferred one.

In our model, only the *Tup* event is considered; how the other types of events can be incorporated will be discussed later. For a *Tup* event, without loss of generality, we assume the destination announced available is an IP prefix located within certain AS. Triggered by this event, the routing items from other ASes to that destination would be subsequently updated. We next define the terms used in the model.

**Definition 1.** *Level.* Suppose $\alpha$ is an AS where a *Tup* event occurs, referred as a *Tup origin*, the *level* of another AS $\beta$ is determined by the length of the shortest path from $\alpha$ to $\beta$.

Based on our assumptions, BGP route update messages are basically disseminated in a process of Breadth First Search (BFS), as illustrated in Fig. 1. In this example, each node represents an AS; there is a *Tup* event and AS $\alpha$ is the *Tup origin*. Since ASes $\beta$, $\gamma$ and $\delta$ are all one hop away from $\alpha$, they are all in *level* 1 of $\alpha$. Once there is a route update message originating from $\alpha$, subsequently $\beta$, $\gamma$ and $\delta$ would update their routing tables and propagate the route update messages to ASes in the next *level*. Likewise, after the second iteration, 5 ASes in *level* 2 would receive the route update message. Such a process repeats until that message finally reaches all the ASes.

**Definition 2.** *Converged state.* For a *Tup* event, an AS reaches a *converged state* if its selected path to the *Tup origin* has been updated by this event and will stay stable.

**Theorem.** *For a Tup event, when all the nodes in level $k$ have been visited by BFS, any node in level $i$ ($i = 0, 1, 2, \ldots, k$) is already in the converged state.*

*Proof.* We prove the theorem by mathematical induction.

$i)$ When $k = 0$, the theorem natually holds.

$ii)$ Suppose when $k = j$ the theorem holds. Fig. 2 depicts the $(j + 1)^{th}$ iteration in BFS, and the *Tup origin* is $\zeta$. For a node $\alpha$ in *level* $j + 1$, suppose among $\alpha$'s neighbors at *level*

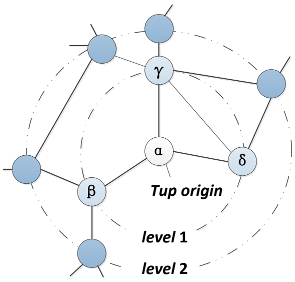

Fig. 1: BGP model

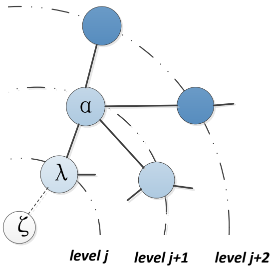

Fig. 2: One iteration in BFS

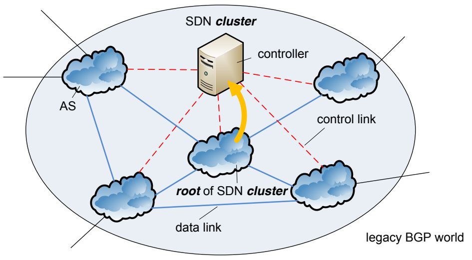

Fig. 3: Inter-domain software-defined routing.

$j$, $\lambda$ is the AS with the smallest ID. Then, since $\lambda$ is in a *converged state* and there are $j$ hops between $\lambda$ and $\zeta$, the updated path recorded in $\alpha$ has $j+1$ hops. Therefore, after this iteration, $\alpha$ has already updated its shortest path to $\zeta$ and is thus in a *converged state*. Similarly, the other ASes in *level* $j+1$ are also in the *converged state*. ☐

With this theorem, the entire network converges when all the ASes have been visited by BFS. To quantify the global convergence time, we give the following definition:

**Definition 3.** *Global convergence delay.* For a *Tup* event, the number of iterations required for visiting all the nodes in BFS from the *Tup origin* is the *global convergence delay*.

Since we assume that the time for propagating BGP route update messages by one hop is fixed, the metric *global convergence delay* can represent the time needed for all the nodes to reach the *converged state*.

Then let us turn to the other types of routing events. Note that the definition of *converged stage* is also applicable to those events. Regarding the *Tshort* event, its behavior is essentially similar to that of a *Tup* event. Based on measurement results in existing work [1], the convergence times of those two kinds of events are almost identical. For *Tdown* and *Tshort* events, [9] analyzed their convergence properties under the utilization of *root cause notification* (RCN). That work implies that the convergence time in these two types of events is proportional to the *global convergence delay* defined above. In sum, our analysis is applicable to all four types of events.

### C. Applying SDN to Inter-Domain Routing

In [5], Kotronis *et al.* proposed an architectural implementation to apply SDN principles in inter-domain routing: ASes outsource their routing control logic to an AS providing routing service, called a *contractor*; they form a SDN *cluster*.

Fig. 3 illustrates the scenario where SDN is applied to inter-domain routing. Here we refer the AS providing the routing service, i.e., the AS where the SDN controller is located, as the *root* of the SDN *cluster*. To avoid unnecessary overhead at the SDN controller, we assume that outsourced to controller is only the control logic of BGP. Besides, to ensure backwards compatibility, a boundary router within the SDN would relay the routing information between the controller and external BGP routers.

The slow convergence of BGP is caused mainly by two factors. Firstly, between two neighboring ASes there exists a complex BGP negotiation process, in the form of a multi-step finite-state machine, which takes a considerable amount of time. Moreover, to avoid oscillation, Minimum Route Advertisement Interval (MRAI) [10] is introduced at each BGP router to trade converging speed for stability.

With SDN, however, since all the distributed forwarding tables are now collected and computed in a centralized manner, the tedious negotiation processes of BGP are no longer needed. In addition, given the existence of a centralized SDN controller, only one MRAI is needed for each SDN *cluster*, instead of one for each BGP router. Thus, we ignore the effect of MRAI within SDN *clusters* in our model. Therefore, the amount of time cost by one hop can be exempted with the assistance of a SDN controller, leading to a smaller *global convergence delay*, which will be analyzed in the next section.

## III. ALGORITHMIC ANALYSIS

Intuitively, if all ASes in the Internet are managed by only one central SDN controller, the BGP convergence time can be minimized in theory. However, this is far from being practical, given the scale of the Internet and the wide adoption of BGP; the only natural and feasible way is to deploy SDN incrementally. Noticing that our main task is thus to pinpoint the best location for incrementally deploying SDN, here, we consider a simple deployment principle, in which for given a SDN *root*, only ASes in *level* 1 join the SDN to form a *cluster*; in other words, ASes in *level* 2 or higher *levels* are not included. This has two implications: practically, it may not be feasible to demand an intermediate AS to route SDN control messages on behalf of another AS, given the concerns such as security; quantitatively, this results in the minimal improvement by using SDN, as the larger is the SDN *cluster*, the better gain can be expected [5], thus providing a lower bound on the performance improvement. Therefore, to describe the deployment of a SDN *cluster*, we now only need to specify an AS to play as the *root* of that *cluster*.

We now present the mathematical analysis for picking up proper ASes to deploy SDN. To start with, given an AS $\alpha$, let $N(\alpha)$ denote the set of its neighbors with $\alpha$ itself included.

After deploying a SDN *cluster* whose *root* is $\alpha$, the set of links covered by that SDN *cluster* can be represented by $\Gamma(\alpha) = \{l \in E \text{ and both endpoints of } l \text{ are in } N(\alpha)\}$.

Given a *Tup* event, with shortest-path policy and tie broken by AS ID, there is only one deterministic best path from the *Tup origin* $\beta$ to any other AS $\gamma$, here denoted by $p_{\beta\gamma}$.

**Definition 4.** *Decisive path.* Given a *Tup origin* $\beta$, $\exists\gamma \in V$, s.t. $|p_{\beta\gamma}| \geq |p_{\beta\lambda}|$, $\forall\lambda \in V$. Then $p_{\beta\gamma}$ is a *decisive path* of $\beta$.

Given a *Tup origin* $\beta$, since many ASes may be in the same *level*, there might be multiple *decisive path*s, but for clarity here we assume only one, simply denoted by $p_\beta$.

**Definition 5.** *Traversing duration*. The *traversing duration* of a *decisive path* is the number of iterations needed in BFS to visit all the nodes in that path.

For a *Tup* event whose *Tup origin* is $\beta$, the *global convergence delay* is the *traversing duration* of $\beta$'s *decisive path*. With one SDN *cluster* deployed and $\alpha$ as the *root*, since links within the SDN *cluster* would cause no delay, the *traversing duration* of a *decisive path* $p_\beta$ can be represented by:

$$T_\alpha(p_\beta) = \sum_{l \in p_\beta} \delta_\alpha(l), \quad \delta_\alpha(l) = \begin{cases} 0, & \text{if } l \in \Gamma(\alpha) \\ 1, & \text{if } l \notin \Gamma(\alpha) \end{cases}$$

The value $T_\alpha(p_\beta)$ is the *global convergence delay* for a *Tup* event whose *Tup origin* is $\beta$. For different *Tup origin*s, that value may be different. We seek to find an AS $\alpha$ to play as the *root* of a SDN *cluster*, so that the average *global convergence delay* with all the *Tup origins* considered can be minimized:

$$\min_{\alpha \in V} \overline{T}_\alpha, \quad \overline{T}_\alpha = \frac{1}{|V|} \sum_{\beta \in V} T_\alpha(p_\beta) = \frac{1}{|V|} \sum_{\beta \in V} \sum_{l \in p_\beta} \delta(l)$$

We then propose the metric *neighborhood weight*, for selecting the AS that can minimize $\overline{T}_\alpha$.

**Definition 6.** *Weight*. The *weight* of a link is the number of different *decisive path*s passing through it:

$$w(l) = \sum_{\beta \in V} \sigma(l, p_\beta), \quad \sigma(l, p_\beta) = \begin{cases} 0, & \text{if } l \notin p_\beta \\ 1, & \text{if } l \in p_\beta \end{cases}$$

**Definition 7.** *Neighborhood weight*. The *neighborhood weight* of an AS $\alpha$ is the total *weight* of all the links in $\Gamma(\alpha)$:

$$W(\alpha) = \sum_{l \in \Gamma(\alpha)} w(l) = \sum_{l \in \Gamma(\alpha)} \sum_{\beta \in V} \sigma(l, p_\beta)$$

Notice that

$$\sum_{l \in p_\beta} (1 - \delta_\alpha(l)) = \sum_{l \in \Gamma(\alpha)} \sigma(l, p_\beta)$$

We have

$$\begin{aligned} \overline{T}_\alpha &= \frac{1}{|V|} \sum_{\beta \in V} \left[ \sum_{l \in p_\beta} 1 \; - \sum_{l \in \Gamma(\alpha)} \sigma(l, p_\beta) \right] \\ &= \frac{1}{|V|} \left[ \sum_{\beta \in V} \sum_{l \in p_\beta} 1 \; - \sum_{\beta \in V} \sum_{l \in \Gamma(\alpha)} \sigma(l, p_\beta) \right] \\ &= \frac{1}{|V|} \left[ \Delta - W(\alpha) \right] \end{aligned}$$

Here $\Delta$ is a constant for a given topology. Thus, to get a minimum $\overline{T}_\alpha$, a maximum $W(\alpha)$ is required. This metric can be used as a guideline to select the optimal AS for deploying SDN; in another word, this is the criterion for optimally selecting a single AS as the *root* of a SDN *cluster*.

If there are multiple SDN *clusters*, when calculating the *neighborhood weight* for an AS outside any existing *cluster*, the *weights* of those links already covered by SDN *clusters* need to be excluded. Let $S$ denote the set of ASes which are the *roots* of current SDN *clusters*, $W(\alpha)$ can be updated by:

$$\tilde{W}(\alpha) = \sum_{l \in \left[ \Gamma(\alpha) \setminus \bigcup_{\epsilon \in S} \Gamma(\epsilon) \right]} w(l)$$

We can now describe a greedy algorithm for selecting $K$ candidates as *roots* of SDN *clusters*.

**Algorithm**: Iterative process for deploying $K$ SDN *clusters*.

```
$i \leftarrow 0$
$S \leftarrow \emptyset$
while $i < K$ do
    calculate $\tilde{W}(\alpha)$ for $\alpha \in V \setminus S$
    select $\hat{\alpha} \in V \setminus S$, s.t. $\tilde{W}(\alpha) \leq \tilde{W}(\hat{\alpha})$, $\forall\alpha \in V \setminus S$
    Deploy new SDN cluster with $\hat{\alpha}$ as the root
    $i \leftarrow i + 1$
    $S \leftarrow S \cup \{\hat{\alpha}\}$
end while
```

Within each iteration, the *neighborhood weights* of all the ASes not included in any existing SDN *cluster* are recalculated, and the AS with the highest *neighborhood weight* value is selected as the *root* of the next SDN *cluster* to be deployed. The performance of this algorithm will be evaluated in the next section.

## IV. SIMULATION RESULTS

In this section, simulation study for corroborating our previous analysis is presented. We first examine some properties of the Internet topology, and then provide the *global convergence delay* distribution for *Tup* events originating from different ASes. Next discussed is the distribution of the proposed metric, *neighborhood weight* for different ASes. Finally the effectiveness of the proposed metric and algorithm are verified through comparisons.

Our simulation is based on the latest data from Internet AS-level Topology Archive [8]. That data, generated by Cyclops

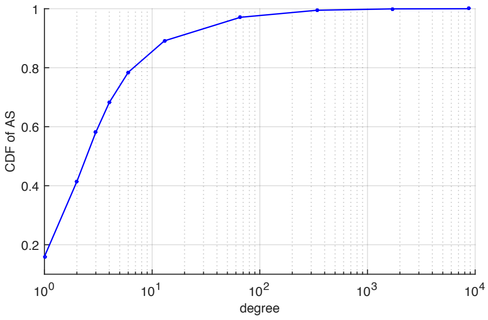

Fig. 4: degree distribution

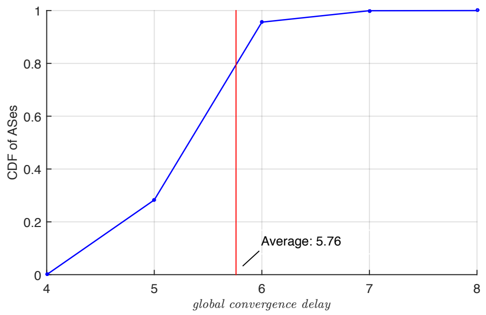

Fig. 5: *global convergence delay* distribution

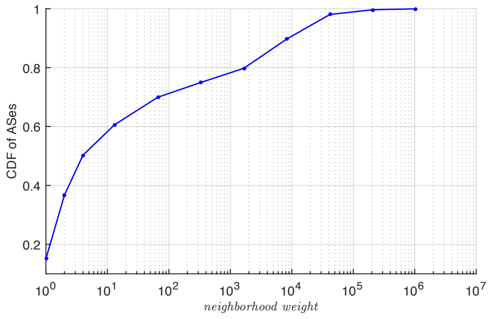

Fig. 6: *neighorhood weight* distribution

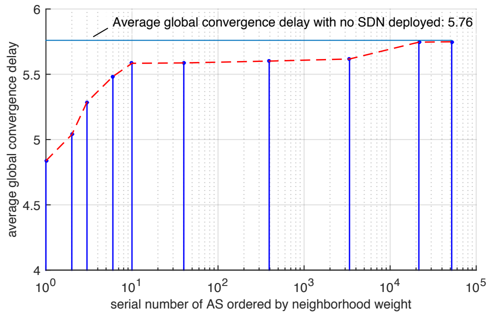

Fig. 7: *global convergence delay* when selecting different ASes as the *root* of a SDN *cluster*

[11], is provided in the form of AS adjacencies and contains $51,984$ ASes. For a better recognition of the Internet topology, we give the degree distribution of all the ASes.

Fig. 4 shows the cumulative distribution function of AS degree, where the degree of an AS is the number of its neighbors. From the figure, more than $95$ percent of all ASes have a degree smaller than $100$, implying that most ASes have a relatively small degree. In contrast, there exist some ASes with an extremely large degree: the highest degree is $8741$, supposed to be from one of the backbone ISPs like AT&T. However, those high-degree ASes are very rare: according to our data, there are only $6$ ASes with a degree larger than $5000$.

The *global convergence delay* distribution with no SDN deployed is then illustrated in fig. 5. Since the *global convergence delays* for different *Tup* origins can be different, for each AS, we first select it as the *Tup* origin, and then simulate the corresponded *global convergence delay*. From the simulation results, although there are a large number of ASes in the Internet, only an average of $5.76$ iterations are required for all ASes to reach the *converged state*. Meanwhile, the maximum value is $8$ and the minimum is $4$, a lower bound reached when the *Tup origin* is the AS with the largest degree ($8741$).

In our analysis, the *neighborhood weight* is used for selecting an AS as the *root* of a SDN *cluster*. Next, we calculate the *neighborhood weights* of all the ASes and the distribution is shown in Fig. 6. What has been revealed is that a majority of ASes have a small *neighborhood weight*, whereas there are also a few ASes with an extraordinary large value, which is consistent with the degree distribution.

To confirm that our proposed metric can help select those ASes contributing most to accelerating global convergence, we conduct a simulation corresponded to Fig. 7. Firstly all the ASes are sorted in a decreasing order according to their *neighborhood weights*. Afterwards, $10$ ASes with different *neighborhood weights* are sampled for independent trials. For each AS, we create a SDN by using that AS as the *root* of the SDN *cluster*, and then compute the average *global convergence delay* by simulating for $1000$ times, where different *Tup origins* are set in different trials. As illustrated in Fig. 7, when the optimal AS is chosen, the *global convergence delay* can be reduced by $16\%$; in contrast, little improvement is brought in when ASes with smaller *neighborhood weights* are chosen. In general, the larger is the *neighborhood weight* of an AS, the shorter is the *global convergence delay* after SDN is deployed based on it.

We next examine the performance of our incremental SDN deployment algorithm. In the experiment, our algorithm introduced in Sec. III is compared with two other algorithms: a greedy algorithm based on the degree, plus a randomized deployment algorithm. Fig. 8 shows, under the three different algorithms, the changing tendency of the *global convergence delay* with stepwise deployment of SDN. Similar with the former simulations, each data point is the average value from $1000$ trials, and for each trail the *Tup origin* is different. By vertical comparison, from the curve of our greedy algorithm,

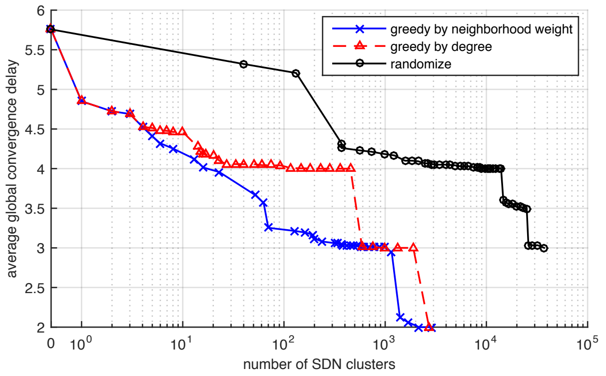

Fig. 8: *global convergence delay* under incremental SDN deployment using different algorithms

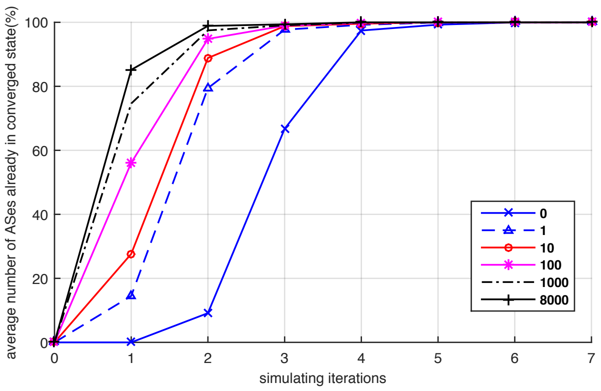

Fig. 9: converging processes with different numbers of SDN *clusters* deployed

deployment of the first 10 SDN *clusters* can decrease the *global convergence delay* by more than 20%. Such effect is less evident when the number of SDN *clusters* increases. Besides, by horizontal comparison, our algorithm, which is based on our proposed *neighborhood weight*, is more efficient than the greedy algorithm based on degree, especially when the quantity of SDN *clusters* is approximately between 10 and 1000. Moreover, in the randomized deployment algorithm, the reduction of *global convergence delay* is not significant until hundreds of SDN *clusters* are deployed, a performance much worse than that of our algorithm.

From Fig. 8 there are some penetrating points, i.e., sharp decrease of *global convergence delay* with little increase of deployed SDN *clusters*. In fact, under the Internet topology there might be multiple *decisive paths* for one *Tup origin*, and only when all those *decisive paths* are overlapped by SDN *clusters* can the *global convergence delay* be reduced. That is the basic cause of the penetrating points.

In the BGP convergence process, which is modeled by us as a BFS-like traverse, the number of ASes in *converged state* keeps increasing with the BFS iterations. Here we take a look at the convergence process with different numbers of SDN *clusters* ($K$ values) deployed under our greedy algorithm, and the number of ASes in *converged state* after each BFS iteration is examined. In Fig. 9, the convergence processes with 0, 1, 10, 100, 1000 and 8000 existing SDN *clusters* are depicted respectively. Each data point is still the average value from 1000 independent trials with different *Tup origins*. From the figure, by horizontal comparison, with very few SDN *clusters* deployed, the BGP convergence process can be sped up remarkably. Taking the second iteration as an example, just with one SDN deployed in our algorithm, the percentage of ASes in the *converged state* can be enhanced substantially from 10% to 80%.

## V. CONCLUSION

Applying software-defined networking or SDN in inter-domain routing is a promising developing direction of the Internet. In this paper, we study how SDN could be used in accelerating global convergence of BGP. By simplifying BGP negotiation process, we propose a mathematical model that can quantitatively obtain the BGP convergence time in an inter-domain routing environment. We examine how SDN can help speed up inter-domain routing, and present a greedy algorithm that selects ASes for incremental SDN deployment to minimize BGP convergence time. The simulation results based on a real-world Internet topology demonstrate the effectiveness of the proposed metric and algorithm.

## ACKNOWLEDGEMENT

The research was supported in part by grants from RGC under the contracts 615613 and 16211715, as well as a grant from NSFC under the contract U1301253.

## REFERENCES

[1] C. Labovitz, A. Ahuja, A. Bose, and F. Jahanian, "Delayed internet routing convergence," *ACM SIGCOMM CCR*, vol. 30, no. 4, pp. 175–187, 2000.
[2] M. Caesar, D. Caldwell, N. Feamster, J. Rexford, A. Shaikh, and J. van der Merwe, "Design and implementation of a routing control platform," in *Proc. USENIX NSDI*, 2005, pp. 15–28.
[3] J. Fu, P. Sjödin, and G. Karlsson, "Intra-domain routing convergence with centralized control," *Computer Networks*, vol. 53, no. 18, pp. 2985–2996, 2009.
[4] A. Gupta, L. Vanbever, M. Shahbaz, S. P. Donovan, B. Schlinker, N. Feamster, J. Rexford, S. Shenker, R. Clark, and E. Katz-Bassett, "Sdx: A software defined internet exchange," in *Proc. ACM SIGCOMM*, 2014, pp. 551–562.
[5] V. Kotronis, X. Dimitropoulos, and B. Ager, "Outsourcing the routing control logic: Better internet routing based on sdn principles," in *Proc. ACM HotNets-XI*, 2012, pp. 55–60.
[6] A. Gämperli, "Evaluating the effect of sdn centralization on internet routing convergence," *Master's thesis, ETH Zürich*, 2014.
[7] T. Wolf, J. Griffioen, K. L. Calvert, R. Dutta, G. N. Rouskas, I. Baldin, and A. Nagurney, "Choicenet: toward an economy plane for the internet," *ACM SIGCOMM CCR*, vol. 44, no. 3, pp. 58–65, 2014.
[8] "Internet AS-level Topology Archive," http://irl.cs.ucla.edu/topology/.
[9] D. Pei, M. Azuma, D. Massey, and L. Zhang, "Bgp-rcn: Improving bgp convergence through root cause notification," *Computer Networks*, vol. 48, no. 2, pp. 175–194, 2005.
[10] T. G. Griffin and B. J. Premore, "An experimental analysis of bgp convergence time," in *Proc. IEEE ICNP*, 2001, pp. 53–61.
[11] "Cyclops," http://cyclops.cs.ucla.edu/.