

An Approach to Accelerate Convergence for Path Vector Protocol

Jiazeng Luo⁺, Junqing Xie^{*}, Ruibing Hao^{*} and Xing Li⁺

jiazeng@ns.6test.edu.cn, jxie1@lucent.com, rhao@research.bell-labs.com, and xing@cernet.edu.cn

⁺Department of Electric Engineering, Tsinghua University, 100084, Beijing, China

^{*}Bell Labs Research China, Lucent Technologies, 100080, Beijing, China

Abstract- BGP, a path vector protocol, is the de facto inter-domain routing protocol. However, slow convergence problem, the bigotry of path vector protocol, has demonstrated a significant impact on the performance of BGP. In this paper, we propose an enhancement to path vector protocol to alleviate the impact of slow convergence process. We compare the convergence time of the legacy path vector protocol and the improved one for complete AS graph, meanwhile the upper and lower bound of convergence time for any AS graph are given for the improved protocol. Simulation results reveal that the improved path vector protocol has a much better performance than the original one.

I INTRODUCTION

BGP, a path vector protocol, is used as the inter-domain routing protocol to exchange routing information among ASes[1]. Maintaining a correct routing table is critical for a BGP system, otherwise the wrong routing table will result in packet loss and degraded quality for applications. However, BGP does not always perform well under all circumstances, especially during the convergence period [2-6]. Convergence is the process of changing from one correct and stable routing table state to another correct and stable routing table state in the event of topology or policy changes. As Labovitz described in [4], BGP demonstrates longer convergence time for T_{down} (a previously available route withdrawn) and T_{long} (an active route with a shorter AS Path implicitly replaced by a new route with a longer AS Path) events compared with the time for T_{up} (a previously unavailable route announced as available) event. And the slow convergence of BGP system is caused by some BGP speakers incorrectly selecting the invalid route before reaching ultimate stable state, and the delay may last for hundreds of seconds or even longer [5]. This phenomenon will threaten the deployment of time critical Internet applications such as VoIP. Therefore, it is desirable to speed up the convergence process of BGP and alleviate the unkind impacts brought by the accidental unreachable of some destinations.

Figure 1 is an example of the slow convergence problem of BGP. Here only one prefix X (an internal network residing in AS_0) is considered. Policy and commercial relationship are not taken into consideration, only the shortest path first rule is applied. All the possible routes learned through BGP are listed aside each AS, and the first line is the best one the AS prefers. Now, suppose prefix X is down for some reason (such as software or link failure, etc.), but it does not affect the normal operation of BGP speaker of AS_0 . This BGP speaker will soon detect X unreachable, and will send withdrawal messages to AS_1 and AS_2 .

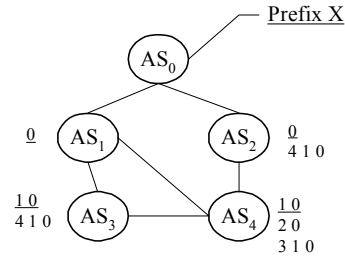


Figure 1. Simple Topology

As AS_1 receives the withdrawal from AS_0 , it will remove route (0) and send withdrawal to both AS_3 and AS_4 . While when AS_2 receives the withdrawal, it will remove route (0), incorrectly select route (4 1 0) as its new best route, and send the new route to AS_4 . However, as for AS_4 , it will receive two messages from AS_1 and AS_2 . If the message from AS_2 comes first, AS_4 will only remove route (2 0) and still take the invalid best route (1 0); and if the message from AS_1 comes first, AS_4 will remove route (1 0) and choose (2 0) as its new best route until it receives the message from AS_2 . This phenomenon (a BGP speaker may select invalid backup route) will continue to happen until reaching the final convergence. It is clear that during the process of convergence, each AS may choose a “best” route to X based on all the information it has, while this “best” route may in fact be an invalid one, which may mislead the BGP speaker and cause packet loss.

One approach to alleviate the problem is to adjust timer setting, especially the Minimum Route Advertise Interval [9]. However, the adjustment of timer is strongly related to the topology of the network. Another approach utilizes some consistency assertions to find out the invalid routes [10], which may bring more burdens to the BGP routers. While in this paper, we would seek another approach to alleviate the problem.

The above example shows the existence of invalid backup routes in BGP is due to the insufficiency of route change information. If a BGP speaker is aware of the exact reason of the route change, it can avoid the selection of invalid backup routes, and thus accelerate the convergence process. Hence, we propose an enhancement to the legacy path vector protocol, which will exchange one additional attribute called Route Change Origin (RCO) to indicate the source/origin of the route change event. For instance, in Figure 1, the withdrawal messages will carry a RCO to specify that the source of prefix X has withdrawn it. When a BGP speaker receives a RCO, it can eliminate all the invalid backup routes at once.

Our research work shows RCO can effectively accelerate the convergence process of path vector protocol, especially for the T_{down} event due to source withdrawal. According to the simulation, it will converge about 2 magnitudes faster than the legacy path vector protocol.

This paper is organized as follows. In section 2, we define two simple path vector protocols, SPVP1 and SPVP2, for the original path vector protocol and enhanced one respectively. Based on the models, section 3 gives the upper and lower bounds of convergence time for T_{down} event due to source withdrawal. Section 4 reports the performance simulation results of SPVP1 and SPVP2. Section 5 concludes the paper.

II SIMPLE PATH VECTOR PROTOCOL MODELS

In order to analyze the convergence properties of the original path vector protocol and the enhanced one, we define two simple path vector protocol models. Model SPVP1 is for the original one, while SPVP2 is for the enhanced one. The basic behaviors of these two models are consistent with those defined in BGP standard [1]. But in order to focus on functions that are directly related to convergence and simplify the analysis, we've made the following assumptions and simplifications: (1) Each AS only has one border router running the path vector protocol. (2) IBGP, policy and multi-BGP sessions between any two ASes are not considered. (3) The propagation and processing time of update message for each router is same. Therefore, these two models are simplified path vector protocols based on shortest AS path first rule and can be analyzed at the granularity of AS instead of routers. Moreover, in the following discussion, only one destination prefix X , belonging to AS_0 , will be considered.

The models are similar to the one defined by Griffin in [7][8]. However, since our models are focused on the convergence properties, the definitions of state and transition are totally different.

A. Original Simple Path Vector Protocol – SPVP1

In the model of SPVP1, the Internet is modeled as an undirected graph $G = \{V, E\}$, where a node $v \in V$ represents an AS while an edge $e \in E$ represents a connection between two ASes. $neighbors(i)$ is the set of neighbors of node i , and $n_i = |neighbors(i)|$ (≥ 1 for connected graphs). At the initial state, node i may have candidate loop-free routes to X , $r_{ij} = (i_j, a_1, a_2 \dots, X)$, from neighbor $i_j \in neighbors(i)$ where $a_1, a_2, \dots \in V$. (Note: for convenience we use X to take place of node number '0'), or no route to X from its neighbor i_j , which is represented as $r_{ij} = (-)$.

Each node i will make route decision and select unique best route r_i with the highest rank value from the n_i candidate routes. The ranking function of r_{ik} is defined as: $rank(r_{ik}) = (1/r_{ik}.length, 1/r_{ik}.first_node)$, which means the shorter the r_{ik} 's length, the higher its rank. If the length of two candidate routes is equal, the smaller the numbering of the r_{ik} 's first node in its node path, the higher its rank. However, if node i does not have candidate route, $r_i = (-)$.

We define node i 's state s_i as its best route r_i , $s_i = r_i$. When its state changes (i.e., $s_i \rightarrow s_i'$), node i may send update messages to all of its neighbor(s). If $s_i \rightarrow s_i' \neq (-)$ and $s_i \neq s_i'$, node i sends an Announcement message to its neighbors j : $m_{ji} = i \rightarrow j : (i \cdot s_i')$. Here, the “ \cdot ” operation is the combination of the path. If $s_i \neq (-) \rightarrow s_i' = (-)$, it sends a Withdrawal message, $m_{ji} = i \rightarrow j : w$.

The global state S is the collection of the states of all the AS nodes in graph G together with all unprocessed messages at each node, i.e. $S = (s_1, s_2, \dots, s_{|V|}, M_1, M_2, \dots, M_{|V|})$, where $M_i = \bigcup_j m_{ij}, j \in neighbors(i)$.

The transition of the global state is defined as follows: $S \xrightarrow{\sigma} S'$, where $\sigma = (\sigma_1, \sigma_2, \dots, \sigma_k, \dots)$, $\begin{cases} \sigma_i \in M_i, \text{ if } M_i \neq \phi \\ \sigma_i = \text{null}, \text{ if } M_i = \phi \end{cases}$.

In the transition, node i will change its state according to σ_i , i.e., $s_i' = process(s_i, \sigma_i)$. And the $process(s_i, \sigma_i)$ will take the following three steps just as what happens in BGP: Step 1. First, node i should refresh the candidate:

- 1) If $\sigma_i = \text{null}$, $r_{ij}' = r_{ij}$ for all j ;
- 2) If $\sigma_i = m_{ik} = k \rightarrow i : w$ ($k \in neighbors(i)$), $r_{ik}' = (-)$, and $r_{ij}' = r_{ij}$, for any $j \neq k$;
- 3) If $\sigma_i = m_{ik} = k \rightarrow i : ka_1 a_2 \dots X$ and i does not appear in $(ka_1 a_2 \dots X)$, $r_{ik}' = (ka_1 a_2 \dots X)$, otherwise $r_{ik}' = (-)$; and for any $j \neq k$, $r_{ij}' = r_{ij}$.

Step 2. Node i takes the route selection process defined above: $s_i' = Best(\{r_{ij}'\})$, $j \in neighbors(i)$

Step 3. Node i sends output message om to neighbors:

- 1) If $s_i' = s_i$, $om_{ji} = \text{null}, \forall j \in neighbors(i)$;
- 2) If $s_i = (ka_1 \dots X)$ but $s_i' = (-)$, $om_{ji} = i \rightarrow j : w$, for $j \neq k$, and $om_{ji} = \text{null}$, for $j = k$, where $j, k \in neighbors(i)$;
- 3) If $s_i' \neq (-)$ and $s_i' = (ka_1 \dots X) \neq s_i$, $om_{ji} = i \rightarrow j : (i \cdot s_i')$ $j \neq k$, and $om_{ji} = \text{null}$, for $j = k$, where $j, k \in neighbors(i)$.

Thereafter, we have the updated $\{M_i'\}, 1 \leq i \leq |V|$:

$$M_i' = M_i - \sigma_i + om_{ji}, j \in neighbors(i).$$

We call a transition σ a *round*. And a finite round sequence $\sigma^1 \sigma^2 \dots \sigma^h$, we call it an *h-round* transition sequence. A global state S is a *final stable state* if it has empty M_i for all $1 \leq i \leq |V|$, while a global state S is a *stable*

state if s_i for all $1 \leq i \leq |V|$ will not change under any transition sequence from S to a *final stable state*. And the length of the transition sequence from an initial global state S_0 to a stable state is the key metric used in our measurement.

B. Enhanced Simple Path Vector Protocol – SPVP2

Now we will enhance the SPVP1 to SPVP2 with the additional topology change information - RCO.

T_{down} event is a frequent event for route change in Internet and has the longest convergence time, thus we focus on reducing the convergence delay under T_{down} event in SPVP2. T_{down} event can be further divided into two types: T_{down} due to source withdrawal and T_{down} due to link failure. And this paper will only discuss the convergence properties for T_{down} event due to source withdrawal.

RCO is a description of the origin of a route change event. In the case of T_{down} event due to source withdrawal, RCO is used to indicate the AS first originating the withdrawal message, denoted as $origin_s(X)$.

When source withdrawal happens, internal prefix X is in fact unreachable, thus any routes in other AS nodes to X become invalid. Therefore, in SPVP2, Withdrawal message will be $m'_{ji} = i \rightarrow j : (w, origin_s(X))$, while Announcement message will keep unchanged. Any message incurred by a message containing a RCO should relay the RCO in its message to others, i.e., RCO is transitive. As a result, $process(s_i, \sigma_i)$ will be modified to $process'(s_i, \sigma_i)$. The only difference is when $\sigma_i = m'_{ik} \neq null$ and m'_{ik} is a Withdrawal message, node i will first invalidate all r_{ij} to prefix X , and take the route decision process as in $process(s_i, \sigma_i)$. The output message will be $om_{ji} = i \rightarrow j : (w, origin_s(X))$, for $j \neq k$ and $om_{ji} = null$ for $j = k$, here $j, k \in neighbors(i)$. Namely, once a node receives a Withdrawal for the first time when the source withdrawal event occurs, this node will mark all the candidate routes to X as invalid. Therefore, each AS will not select incorrect route any more.

III ANALYSIS OF CONVERGENCE PROPERTIES

Based on the models above, we will discuss the convergence properties under source withdrawal event for SPVP1 and SPVP2 respectively. Here only the Propositions are listed, the detailed explanation can be found in [13].

First we can easily get the following proposition.

Proposition 1. For T_{down} event due to source withdrawal, node of SPVP2 system will converge as soon as it receives the Withdrawal, and it will not announce any invalid route except Withdrawals.

A. Convergence properties for complete AS graph

In this part, the convergence time of a complete AS graph will be discussed. Although complete AS graph is a very special case, it has been used frequently as a reference to analyze the convergence properties [4][9].

Here we consider a complete graph G of size n with node AS_1, AS_2, \dots to AS_n . And all of the nodes are directly connected to a node AS_0 (also denoted as node X), which initially possesses prefix X and withdraws the prefix later to produce a T_{down} event due to source withdrawal. In fact, the above complete graph is of size $n+1$ (including node X), but the state of X is not changed after sending out the withdrawal and has no impact on the final results. Hence we still call it an n -node complete graph in the Propositions.

For such T_{down} event due to source withdrawal, the value of $origin_s(X)$ is $[X]$, the initial global state S_0 is $((X)_1, (X)_2, \dots, (X)_n, \{X \rightarrow 1:w\}_1, \{X \rightarrow 2:w\}_2, \dots, \{X \rightarrow n:w\}_n)$ and the final stable state is $((-)_1, (-)_2, \dots, (-)_n, \phi_1, \dots, \phi_n)$. And the convergence time in the following results is measured using the number of rounds from S_0 to a stable state S_s in which each s_i equals $(-)$.

Proposition 2: For a complete graph G of size n , the convergence time for T_{down} event due to source withdrawal is 1 for SPVP2.

Proposition 3: For a complete graph G of size n , the lower bound of convergence time for T_{down} event due to source withdrawal for SPVP1 is $O(n^2)$.

Proposition 4: For a complete graph G of size n , the upper bound of convergence time for T_{down} due to source withdrawal for SPVP1 is $O((n-1)!)$.

This is consistent with the observation Labovitz got for a complete graph with n AS nodes [4].

B. Convergence properties of SPVP2 for any AS graph

For non-complete AS graph, the convergence properties are more difficult to get and still unclear for SPVP1. But the following deterministic results can be derived for SPVP2.

Proposition 5: For any AS graph running SPVP2, the convergence time of node i for T_{down} due to source withdrawal is $|p|$, where p is the shortest path from node i to the withdrawal origin $origin_s(X)$, and the system convergence time is proportional to $|p_l|$, where p_l is the longest path and namely the depth of AS graph with $origin_s(X)$ as its root.

The above analysis give the result based on the concept of ‘round’ defined in SPVP, it is quite different from the real BGP operation, where different ASes may have different update message propagating time and processing time. However, Proposition 6 will give the upper and lower bounds of convergence property for any AS graph based on the total

consumed messages, which is more intuitive and can be measured in simulation or in practical operation as well.

Proposition 6: For any AS graph, let $|V|$ and $|E|$ denote the number of nodes and edges in the graph, the total messages consumed for the SPVP2 system to converge for T_{down} due to source withdrawal is $2|E| - |V| + 1$.

IV PERFORMANCE EVALUATION OF SPVP1 AND SPVP2

In this section, we report the performance evaluation results of SPVP1 and SPVP2 based on simulation.

A. Methodology

The simulation tool we used is the Scalable Simulation Framework (SSFnet) simulator [11]. Some tiny modifications have been made to the original BGP implementation in SSFnet (which conforms to SPVP1) to implement SPVP2: the RCO is coded as a transitive optional attribute in the modified BGP implementation. Besides, sender side loop detect mechanism is enabled in our simulation.

Two kinds of AS topology graphs are used in our simulation. One is a simple full mesh AS graph with 10 nodes. The other is generated by the BRITE topology generator [12] with 70,100,150 and 200 nodes. For those generated graphs, each node has an average of two links and there is no stub node. Besides, most of the depth of spanning trees rooted at every node is controlled to about 4 or 5, and we expect this topology more like a subset of nowadays Internet.

In practice, time spent by a message is related to message propagating, queuing and processing. Considering the practical Internet, the time parameters in SSFnet is set as follows. The propagating time for each link is a constant value, i.e., 300ms in the full mesh case and 50ms in the BRITE generated cases. And the other two times are randomly distributed between 0.1 and 1.0 seconds for each node. Besides, we expect that the AS with more links takes a little longer time, so that we let those ASes spend additional more time on message queuing and processing.

The simulation is taken as follows: for each specified topology, randomly select one node containing destination X as Source AS. At the beginning, the system operates as normal to reach a stable state. After a period of time, the Source AS withdraws the prefix X it once announced, during which we record the evaluation metrics for the convergence process. For each topology graph, we repeat the simulation for 1,000 times and calculate the average metrics.

B. Evaluation metrics

In order to get a more comprehensive view on the performance of SPVP1 and SPVP2, we use two metrics:

- 1) Total number of updates messages (including Withdrawal and Announcement messages) to converge; and,

- 2) The system convergence time (the time taken from the initial time at which the withdrawal event occurs to the time the last node in the system converges).

C. Simulation results

C.1 10-node complete graph simulation result

Figure 2 (a) and (b) show the total number of update messages and convergence time distribution of 1000 tests for SPVP1 and SPVP2 with a 10-node mesh topology. The

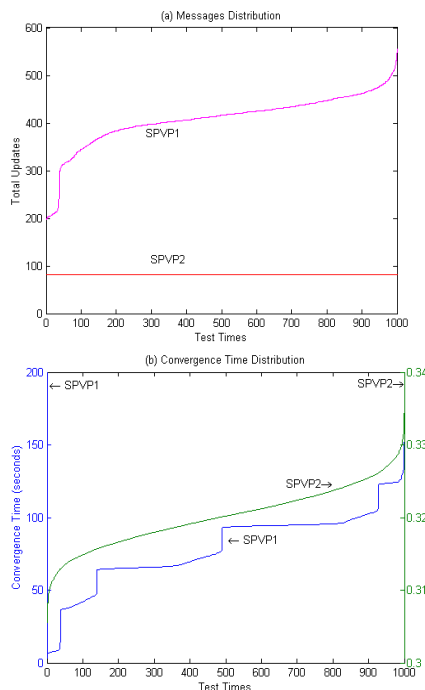


Figure 2 Convergence properties due to source withdrawal (10-node complete graph, SPVP1 and SPVP2)

horizontal axis is the test number, and the vertical axis is the total number of updates in 2(a) and the convergence time in seconds in 2(b). For each test, the total updates or convergence time are all different, so we sort the data of each test according to their value.

The simulation result shows that in most of the tests for SPVP1, the system would take 200 to over 500 messages to converge, and convergence times are mostly distributed

at some steps of 30 seconds, such as about 60, 90 or even more seconds (Figure 2(b)). But as to SPVP2, the system will only take 81 updates after the T_{down} due to source withdrawal event happens. Moreover, the convergence time of SPVP2 is also less than 1 second. During the convergence, no invalid route is selected.

C.2 BRITE-generated topology simulation result

Figure 3 (a) and (b) exhibit the total number of updates and the convergence time distribution of SPVP1 with four curves, corresponding to the four different-size topologies. The horizontal and vertical axis has the same meaning as in Figure 2. Figure3 illustrates that, for SPVP1, routing system would usually take 10^3 to 10^4 updates and 10^2 to 10^3 seconds to converge.

Figure 4 shows the source withdrawal convergence properties of SPVP2. We can see that the total number of updates messages consumed to converge is proportional to the size of the system. And the convergence time for SPVP2 also only takes less than 10 seconds. These two properties are both improved with almost 2 orders compared to SPVP1.

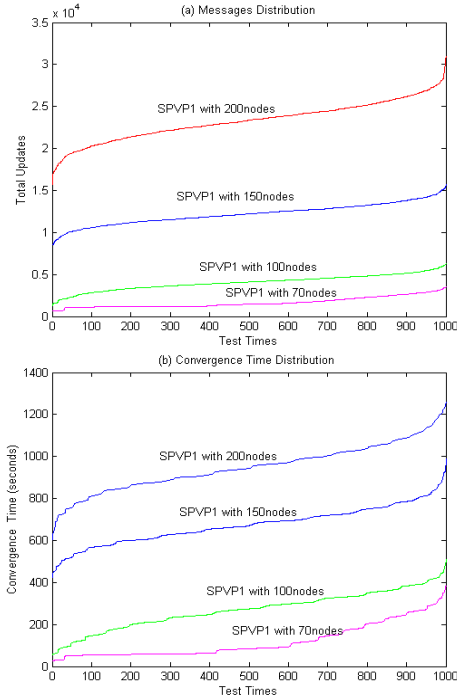


Figure 3 Convergence properties due to source withdrawal (BRITE generated topologies, SPVP1)

Table 3: Theoretical & Experimental number of Update messages due to source withdrawal (BRITE generated topologies, SPVP2)

System Size	70	100	150	200
Total Edges	137	197	297	397
Theo. Updates	205	295	445	595
Exp. Updates	205	295	445	595

In Table 3, we give the theoretical estimation of the number of update messages of SPVP2 for the four systems (according to Proposition 6 of Section 3), together with the observed values in simulation. From which we can see that the observed values are all consistent with the theoretical ones.

V SUMMARY AND FUTURE WORK

From our analysis and simulation, SPVP2 with RCO demonstrates a much better performance than SPVP1 during the convergence period, especially for T_{down} event due to source withdrawal. Of course, to apply to the current BGP, it requires to add a new transitive optional attribute, thus some modifications to current BGP implementation are necessary. But the improved performance should make it up.

However, we still need to do more work to understand the RCO mechanism, including the convergence properties on T_{down} due to link failure event, policy change and other situations. And work has already been in progress with some simulation observations, which we would discuss in future papers. Security should also be studied in case RCO becomes

another DoS attack source. But we believe our idea would provide another promising approach to alleviate the slow convergence problem of the path vector protocol.

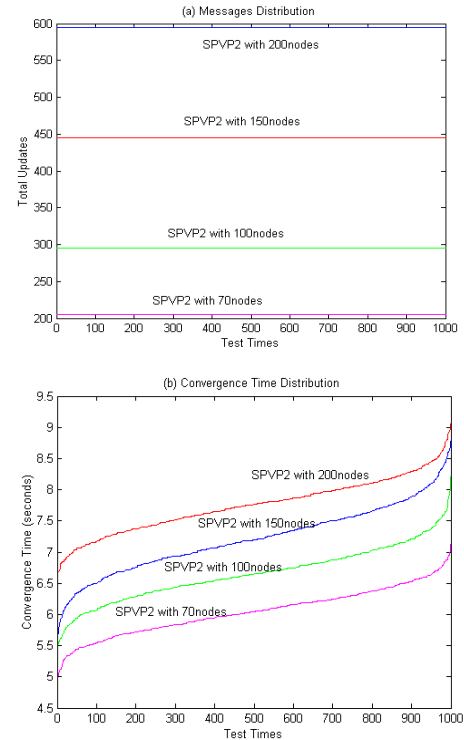


Figure 4 Convergence properties due to source withdrawal (BRITE generated topologies, SPVP2)

REFERENCES

- [1] Y.Rekhter and T.Li, A border gateway protocol, RFC 1771, March 1995.
- [2] C.Labovitz, G.Malan and F.Jahanian, Internet Routing Instability, in Proceedings of ACM Sigcomm 97, Sept.1997.
- [3] C.Labovitz, A.Ahuja and F.Jahanian, Experimental Study of Internet Stability and Wide-Area Network Failures, FTCS 99, June 1999.
- [4] C.Labovitz, A.Ahuja, A.Bose and F.Jahanian, Delayed Internet Routing Convergence, Sigcomm 00, Aug.2000.
- [5] C.Lavovitz, R.Wattenhofer, S.Venkatachary and A.Ahuja, The impact of Internet policy and topology on delayed routing convergence, IEEE INFOCOM 01, April 2001.
- [6] R.Govindon and A.Reddy, An Analysis of Internet Inter-Domain Topology and Route Stability, IEEE INFOCOM'97, Japan, April 1997.
- [7] T.Griffin and G.Wilfong, An Analysis of BGP Convergence Properties, in Proc. ACM SIGCOMM, Aug.1999, pp.277-288.
- [8] T.Griffin, F.Shepherd and G.Wilfong. Policy Disputes in Path-vector Protocols. In Proc. Inter. Conf. on Network Protocols, Nov. 1999.
- [9] T.Griffin and B.Premore, An Experimental Analysis of BGP Convergence Time, In Proc. Inter. Conf. on Network Protocols, Nov.2001.
- [10] D.Pei, X.Zhao, L.Wang, D.Massey, A.Mankin, S.Wu and L.Zhang. Improving BGP Convergence Through Consistency Assertion. IEEE INFOCOM 02, New York, June 2002.
- [11] Scalable Simulation Framework, <http://www.ssfnet.com/>.
- [12] Boston University Representative Internet Topology Generator, <http://www.cs.bu.edu/brite/>.
- [13] J.Luo, J.Xie, R.Hao and X.Li, An approach to accelerate convergence for path vector protocol, Lucent Technical Memo: ITD-02-43169Y, June, 2002.