Feasibility study on distributed simulations of BGP

David Coudert, Luc Hogie, Aurélien Lancin, Dimitri Papadimitriou, Stéphane Pérennes, Issam Tahiri

Feasibility study on distributed simulations of BGP

David Coudert*, Luc Hogie*, Aurélien Lancin*, Dimitri Papadimitriou†, Stéphane Perennes*, Issam Tahiri*

*INRIA, I3S(CNRS, UNS), 2004 route des Lucioles, 06902 Sophia Antipolis, France
Email: (david.coudert,luc.hogie,aurelien.lancin,stephane.perennes,issam.tahiri)@inria.fr
†Alcatel-Lucent Bell, Copernicuslaan 50, 2018 Antwerpen, Belgium
Email: dimitri.papadimitriou@alcatel-lucent.com

Abstract—The Autonomous System (AS) topology of the Internet (up to 61k ASs) is growing at a rate of about 10% per year. The Border Gateway Protocol (BGP) starts to show its limits in terms of the number of routing table entries it can dynamically process and control. Due to the increasing routing information processing and storage, the same trend is observed for routing model simulators such as DRMSim specialized in large-scale simulations of routing models. Therefore, DRMSim needs enhancements to support the current size of the Internet topology and its evolution (up to 100k ASs). To this end, this paper proposes a feasibility study of the extension of DRMSim so as to support the Distributed Parallel Discrete Event paradigm. We first detail the possible distribution models and their associated communication overhead. Then, we analyze this overhead by executing BGP on a partitioned topology according to different scenarios. Finally, we conclude on the feasibility of such a simulator by computing the expected additional time required by a distributed simulation of BGP compared to its sequential simulation.

Index Terms—distributed simulation; network; BGP; Internet;

I. INTRODUCTION

The Internet evolution pushes its routing system to its limits in terms of i) memory cost due to the size of the routing tables (RT); ii) communication cost or complexity; iii) Border Gateway Protocol (BGP) complexity. Routing research has investigated new routing paradigms to address these issues. However, simulation of stateful distributed routing protocols operating asynchronously becomes a real issue at large-scale (order of $10k$ nodes) [9]. No simulator provides the means to measure and to characterize the performance and behavior of such routing protocols on large networks and to compare them with BGP on the same simulation environment. For this purpose, we study the extension of DRMSim [3] in order to support the distribution of the routing model by partitioning the topology with respect to its properties and by extending the communication model.

This paper is organized as follows. After describing in Section II the state-of-the-art in routing model simulation and DRMSim, we detail two distributed models together with their associated communication overhead. In Section IV, we describe our simulation scenarios and execution environment followed by the simulation results as well as the impact of topology partition on the communication and their analysis. Finally, we conclude on the feasibility of such a simulator in Section V.

II. STATE OF THE ART

We can distinguish three classes of routing simulators: i) protocol simulators dedicated to the performance measurement and analysis of the routing protocol (procedures and format) at the microscopic level such as NS [5] and SSFNet [7]; ii) configuration simulators dedicated to the simulation of BGP protocol specifics like SimBGP [8] and C-BGP [6]; iii) routing model simulators like DRMSim [3] that do not execute the protocol low level procedures but their abstraction. Designed for the investigation of the performance of dynamic routing models on large-scale networks, these simulators allow execution of different routing models and enable comparison of their resulting performance.

Routing model simulators require specification of an abstract procedural model, data model, and state model sufficiently simple to be effective on large-scale networks but still representative of the actual protocol execution. However, incorporating (and maintaining up to date) routing state information is technically challenging because of the amount of memory required to store the data associated to each state.

DRMSim implements the Discrete-Event Simulation (DES) approach. It addresses the previous issue by means of efficient graph-based data structures. It allows us to defined simulation scenarios where topology dynamics like link/router failures are considered. Several models can be easly developed and integrated with their own metric model for performance measurement. The provided set of models includes BGP, the Routing Information Protocol (RIP), and compact routing schemes such as NSR [4] and AGMNT [1].

III. DISTRIBUTION MODELS

Implementing DRMSim as a distributed parallel discrete event simulator implies to distribute routers with their data structures but also the events execution among the different computational resources. The simulation feasibility depends mainly on the number of events to be transmitted between each logical processes (LP) and the available bandwidth between them.

The main problems observed with the non-distributed version of DRMSim are:

- Routing tables (RT): let n be the number of routers in the network and k the size of a routing entry. The memory needed for storing all the routing tables is $O(k.n^2)$.
- Entries updates: event with the highest number of occurrences. We always start from empty routing tables.

Routers and BGP peering sessions are distributed among the different logical processes. We model the system by a graph $G(V, E)$, where the set of vertices V is partitioned and every partition is managed by a logical process. The edges internal to a partition are known only by the corresponding LP. For edges with end-points in distinct partitions, there are two possible solutions.

- *Solution A*: update events including the corresponding entries are transmitted between LPs.
- *Solution B*: boundary edge end-points are duplicated, implying that the original vertices (as an example v and w in Fig.1 must synchronize their copies (v' and w').

The advantage of *Solution B* compared to *Solution A*, is that only modified RT entries are sent, thus reducing the communication cost. However, duplication of vertices (routers) may be very harmful if the cuts resulting from the partitioning algorithm have many edges, reducing in turn the memory gain.

A. Partitioning algorithms and complexity

We assume that BGP routing updates constitute the main problem to solve, other type of events lead to negligible effects. For convenience, we use the following notations:

- $G(V, E) \triangleq$ the graph representing the network topology;
- The vertices of the graph are partitioned: $V = \cup_{i=1}^{K} V_i$;
- $N_{V'}(v) \triangleq$ the neighbors set of a vertex v on the subgraph induced by V';
- $ME(v) \triangleq$ the set of modified RT entries at the router modeled by v;
- $e(v, V') = 1$ if v is not in V' and has a neighbor in V', and 0 otherwise;
- $Esize \triangleq$ the average size of a RT entry.

Solution A implies no internal communication overhead, but leads to $\sum_{i=1}^{k} \sum_{v \in V_i} (|N_{V \setminus V_i}(v)|.|ME(v)|)$ communication between LPs. In *Solution B*, as boundary nodes are duplicated, there are $\sum_{i=1}^{k} \sum_{v \in V_i} [(\sum_{j=1}^{k} e(v, V_j)).|ME(v)|]$ internal updates, but the communication between LPs is negligible since we only need to send an update event containing the identifiers of both routers. There is no memory overhead in *Solution A* compared to the $|V|.\sum_{i=1}^{k} \sum_{v \in V_i} [(\sum_{i=1}^{k} e(v, V_j))]$ bits that are needed in *Solution B* to store the duplicated boundary nodes.

We use mixed integer programs (MIP) to compute optimal bipartitions of the topology by minimizing the communication
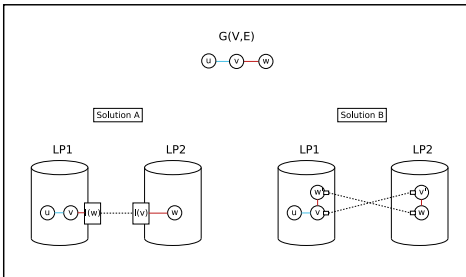
overhead. In *Solution A*, we assign to every edge $uv \in E(G)$ a weight W_{uv} that approximates the number of entries exchanged on this edge. The sum of weights over all edges having end points in different subsets is minimized. In *Solution B*, we assign to every vertex $v \in V(G)$ a weight W_v that approximates $|ME(v)|$. In this case, the sum of weights over all vertices having at least one neighbor in the foreign subset is minimized.

IV. EXECUTION

Our objective is to determine the communication overhead of BGP caused by a distributed parallel implementation of DRMSim. We count the number of BGP update messages exchanged at the boundary edges of the topology partitions. The communication time overhead is derived by estimating the transmission and the propagation time for a single BGP update message between two LPs.

A. Simulation scenarios

We first consider MinRouteAdvertisementInterval (MRAI) = 0s in our experiments. It represents an upper bound on the amount of communication between BGP routers. Indeed, in the Internet, the default MRAI value is set to 30s in order to limit the message rate between BGP peers.

The BGP peering session establishment delay together with the update propagation delay between routers play a major role in the amount of transmitted updates. Three scenarios have been elaborated:

- *Scenario 1*: considers BGP peering sessions establishment before the start of updates exchanges. Once sessions are established, received updates are executed in their scheduled order.
- *Scenario 2*: considers *Scenario 1*. However, upon reception, updates are executed in a random order simulating highly random communication delay.
- *Scenario 3*: after one peering session establishment, the resulting updates are executed. Then, the next scheduled BGP peering session is established.

These scenarios are executed once on topologies of 2.5k, 3k, 3.5k, 4k, 4.5k and 5k nodes. These topologies are generated according to the Generalized Linear Preferential (GLP) model parameterized as specified in [2].

B. Execution environment

Sequential simulations of BGP to derive inter-partitions communication have been executed with DRMSim on an Intel Xeon 3.20Ghz with 64GB of RAM. Execution of *Scenario 1* took 52 minutes (min), *Scenario 2* 48min and *Scenario 3* 16min on topologies of 5k nodes.

C. Simulation results

We first measure to obtain reference values, the number of BGP updates with their respective number of entries on non-partitioned topologies according to the different scenarios.

Fig. 1. Two solutions to allocate edges with end-points on different LPs

TABLE I
NUMBER OF UPDATE ENTRIES

	2.5k	3k	3.5k	4k	4.5k	5k
Scenario 1 ($\times 10^6$)						
No partition	24.6	36.1	50.1	65.0	83.1	102.4
Sol A on bipartition	2.6	3.8	5.1	6.7	9.4	10.5
Sol B on bipartition	1.4	2.2	2.9	3.9	5	6.1
Scenario 2 ($\times 10^6$)						
No partition	58.7	87.0	121.7	158.8	204.3	252.8
Sol A on bipartition	6.1	9.0	12.3	16	20.4	25.2
Sol B on bipartition	2.8	4.4	6.2	8	10.2	12.6
Scenario 3 ($\times 10^6$)						
No partition	33.5	49.0	68.4	88.0	111.7	138.8
Sol A on bipartition	3.3	4.6	6.4	8.3	10.3	12.8
Sol B on bipartition	2.6	3.9	5	6.9	8.6	10.9

1) BGP reference results on non partitioned topologies:
From Table I on non partitioned topologies, the number of BGP update entries increases drastically. If the communication overhead between LPs behaves similarly, then the time to perform simulations becomes excessive. This observation shows that setting MRAI value to 0s leads to detrimental effects in terms of routing convergence time. We also observe that the increase of BGP update entries over the topology size is linear in their root square, allowing us to extrapolate this number for the topology size of interest (100k nodes).

2) BGP results on bipartitioned topologies: We compute optimal bipartitions of the topologies according to the methodology presented in Section III-A with Solution A and B. As optimal bipartitions, we know that the measured communication give us a lower bound on the overhead.

In *Solution A*, for the three scenarios and a topology of 5k routers, around 10% of the total number of updated entries transit between partitions. In *Solution B*, we observe for the three scenarios that the number of transmitted entries improves compared to *Solution A*. Only 5.6% of the reference updated entries have to be transmitted with *Scenario 1*, 5% for *Scenario 2* and 8% for *Scenario 3*.

3) Communication overhead: We measure the total size of updated entries transmitted for the different scenarios and topologies. The average size of each entry is 1 for *Scenario 1* and between 2.8 and 4.5 for *Scenarios 2* and *Scenario 3*. We also need to determine the time to transmit an updated entry between two LPs. Considering 1Gbps link using TCP/IP, we obtain an average of 0.26ms for each packet (over 1000 packets sent one after the other) to reach its destination.

With *Solution A* applied to *Scenario 1* on a topology of 5k routers, we can expect 45min to transmit all updated entries between the LPs. By extrapolation, we can expect around 91min of overhead for a topology of 10k routers and 15hours for 100k routers. For the two other scenarios executed on a topology of 5k routers, *Scenario 2* cumulates an overhead of 110min and *Scenario 3* remains in the same order of time as *Scenario 1*. By extrapolating *Scenario 2* overhead, we can expect about 4h of overhead for a topology of 10k routers and about 36hours for 100k routers.

V. CONCLUSION

The expansion of the Internet and its dynamics require to handle the evolutions of its routing system for the next decade. Moving DRMSim to a Distributed Parallel Discrete Event simulator seems to provide a promising technique to make abstraction of the topology size although it induces communication overhead between logical processes. Therefore, this paper quantifies the expected additional time needed to distribute the simulation of BGP on topologies representative of this growth, i.e., 10 to 100k. For this purpose, we have identified BGP updates as the main cause of communication between nodes. We computed the number of updates and their respective number of entries for different BGP execution scenarios. Then, we have computed the best bipartition of each topology - by using the previous measured number of updates - to derive the amount of communication (the overhead) between partitions. It appears that for Scenario 1, distributing the simulation between partitions requires an additional time ranging from 91min (10k nodes) to 15hours (100k nodes). As shown by the Scenario 2 and Scenario 3, this additional time clearly depends on the execution scenario and can reach up to 36hours for topologies of 100k nodes. Decreasing the number of updates can be achieved by increasing the MRAI time to a value higher than 0. As the MRAI increases, the number of BGP updates decreases significantly (in the order of factor 10), until the convergence time reaches an average minimum. The critical issue becomes thus to determine the optimal MRAI value, for which the decrease in update rate would not increase the convergence time while limiting the amount of additional required memory. In order to further reduce the overhead, we have proposed to synchronize boundary nodes between partitions. By applying this method, we were able to save half of the overtime when considering Scenario 1 and 2. Consequently, such a distributed parallel simulator seems thus feasible, leading in turn to the next step of our work being the extension of our routing model simulator DRMSim according to the proposed distribution models.

REFERENCES

[1] I. Abraham, C. Gavoille, D. Malkhi, N. Nisan, and M. Thorup, *Compact name-independent routing with minimum stretch*, ACM Transactions on Algorithms, vol.4, no.3, Jun.2008.

[2] T. Bu, and T. Don, *On distinguishing between Internet power law topology generators*, Proc. 21th Annual IEEE International Conference on Computer Communications (INFOCOM), vol.2, 2002.

[3] *DRMSim simulator*. http://drmsim.gforge.inria.fr/.

[4] N. Nisse, K. Suchan, and I. Rapaport, *Distributed computing of efficient routing schemes in generalized chordal graphs*. Theoretical Computer Science, to appear.

[5] *The Network Simulator - ns-2*. http://www.isi.edu/nsnam/ns/.
B.J. Premore and T.G. Griffin, *An Experimental Analysis of BGP Convergence Time*, Proc. 9th IEEE International Conference on Network Protocol (ICNP'01), pp.53–61, Riverside (CA), USA, Nov.2001.

[6] B. Quoitin, S. Uhlig, *Modeling the routing of an autonomous system with C-BGP*, IEEE Networks, vol.19, no.6, pp.12–19, Nov.2005.

[7] *Scalable Simulation Framework (SSF)*. http://www.ssfnet.org/

[8] *SimBGP, a simple BGP simulator*. http://www.bgpvista.com/simbgp.php.

[9] G. Yaun, et al., *Large scale network simulation techniques: Examples of TCP and OSPF models*, ACM SIGCOMM Computer Communication Review (CCR), 2003.