A Digital Twin based Framework to Enable "What-If" Analysis in BGP Optimization

Marco Polverini, Ilaria Germini, Antonio Cianfrani, Francesco G. Lavacca, Marco Listanti

Department of Information engineering, Electronics and Telecommunications (DIET)
University of Roma "Sapienza" - Via Eudossiana 18, 00184 Roma, Italy
Telephone: +39 0644585371, e-mail: name.surname@uniroma1.it

***Abstract*—Nowadays, inter domain routing optimization is performed based on the so called "Tweak and Pray" approach, which consists in performing changes in the configuration of the BGP protocol without knowing in advance the consequences of such a modification. This is due to the lack of cooperation among Network Operators in the configuration of the BGP to optimize the inter domain routing. Inefficiency in the resource usage, network anomalies and outages are common consequences of wrong configuration changes performed by Network Operators in an attempt to improve the performance of their infrastructures. In this paper we propose a novel framework based on the Digital Twin technology to enable the execution of "what-if" analysis in the context of Traffic Engineering performed by tuning BGP parameters. Such a paradigm shift will allow Network Operator to be aware of the effects of BGP configuration changes before their actual execution. A proof of concept related to the balancing of inbound traffic in an Autonomous System network, based on the use of the AS Path Prepending technique, is realized to validate the feasibility of the proposed approach.**

***Index Terms*—Digital Twin Network, Border Gateway Protocol, What-If analysis**

I. Introduction

The Internet is an interconnection of network infrastructures owned and operated by different entities, the so called Autonomous Systems (ASs). To determine an end-to-end route over the Internet, two distinct routing problems must be solved. The first one aims at determining the route to transit through an AS, i.e., the intra-AS routing, while the latter consists in the assessment of the sequence of ASs that the traffic has to go through to reach a destination prefix, i.e., the inter-AS routing.

Concerning the inter-AS routing, the Border Gateway Protocol (BGP) [1] is the de-facto protocol used to exchange routing information between connected ASs. Along the years, several strategies have been defined to optimize the network performance by tuning the BGP configuration parameters, such as the AS Path Prepending or the use of BGP Community attribute [2]. For example, in an attempt to balance the incoming traffic among peering links, a Network Operator (NO) can perform the AS Path Prepending in order to force other ASs to prefer the use of a specific route for a prefix [3]. Nonetheless, since each AS only knows its own BGP configuration, the effect of a change cannot be predicted before its application, leading to the definition of the so called *"tweak and pray"* approach. In practice, a NO will be aware of the effectiveness of a performed action only after its application.

In today's network ecosystem, where applications ask for more reliability and where few seconds of service disruption can potentially turn into severe revenue losses, the actuation of such a paradigm is unacceptable. As a concrete example, in October 2021, Facebook infrastructure has experienced an outage due to a misconfiguration happened during a maintenance activity [4]. The performed changes have caused the withdrawal of the Facebook's datacenter prefixes that were advertised through BGP, leading to a disconnection of its datacenters from the rest of the Internet. Another example is the route leak [5] event happened in 2015 and involving Malaysia Telecom and Level 3 [6], a major backbone provider. Malaysia Telecom told one of Level 3's networks that it was capable of delivering traffic to anywhere on the Internet. Once Level 3 decided the route through Malaysia Telecom looked like the best option, it diverted a huge amount of traffic to Malaysia Telecom. In these situations, the availability of a tool to perform *"what if" analysis* would have been able to prevent the outages.

For all these reasons, a shift of paradigm is mandatory, passing from an incautious *"tweak and pray" approach* to a more reliable and consequence aware *"what if" analysis*. In the past, several attempts have been done in this direction. For instance, in [7] it is proposed a framework to compute the outcome of the BGP route selection process for each router in a single AS given as input a new BGP update. Despite these approaches aim to overcome the criticality of the *"tweak and pray"* approach, they present some weaknesses, such being too intrusive to the normal operation of the BGP protocol, or by providing inaccurate estimation of the final outcome of a protocol re-configuration.

In this paper, we aim to adopt the Digital Twin (DT) [8] philosophy in order to enable such a paradigm shift. A DT is a digital replica of a real system, which is continuously synchronized with its physical counterpart, thus reproducing an accurate description of its current status. In the considered scenario, the physical system is represented by an inter domain network realized as the interconnection of multiple ASs: it is then a system of systems. Thus, we propose to model a single AS with a DT, and the whole system with a network of DTs, i.e., a Digital Twin Network (DTN). In particular, the main

contributions of the paper are the following ones:

- the proposition of a DTN to enable the execution of *"what if" analysis* in the context of BGP configuration;
- the design of an architectural framework to handle the management of the DTN;
- the creation of a Proof of Concept (PoC) to validate the effectiveness of the proposed approach.

The rest of the paper is organized as follows: Sec. II provides an overview of the state of the art with respect to the DT technology and inter domain routing optimization, Sec. III introduces the proposed architectural framework, whereas the PoC is presented in Sec. IV, while Sec. V concludes the paper.

II. Related Work

The performance optimization of the inter domain routing requires to perform configuration changes involving the BGP protocol. A common approach that is used by the NOs to balance the incoming traffic is the AS Path Prepending, i.e. increasing the length of the AS Path attribute of a BGP Update by repeating the same AS Number multiple times. A tool based on this approach is described in [9], which introduce the *AutoPrepend* tool that is able to evaluate the effect of AS Path Prepending in an automated fashion. At the same time, a NO can change the *local preference* parameter in order to optimize the distribution of the traffic exiting the AS [2]. By properly tuning this parameter it is possible to find a suitable balance of the outgoing traffic over the different peering connections.

The previous techniques are all based on a trial and error approach, where a NO changes the configuration of its devices and evaluates the impact of such an action; configurations changes can cause system instability during the network convergence period. For this reason a branch of the literature has focused on the possibility to build models for predicting the effect of a configuration change in the BGP protocol. [10] is one of the first works aiming at enabling *what-if* analyses in the context of inter domain routing. Specifically, the authors propose a framework to predict routes in the Internet in case of specific events (e.g., removal of a peering connection). Differently than previous approaches, which assumed that an AS is atomic, [10] introduces the concept of multiple quasi-routers to capture route diversity within the ASs, thus allowing to discover multiple routes inside a single AS. In [7] a model for the prediction of the route selection process performed by each router of a single AS after a configuration change, is proposed. Unfortunately, both methods have some limitations, being based on inference (thus introducing possible errors in the output of the *what-if* analysis), or being limited to a single AS. In this paper, we aim to adopt the DT technology to overcome these limitations.

The DT technology is getting momentum in the context of communication networks. There are two different types of approaches to treat DTs in the networking field: i) the network for the DT, and ii) the DT for the network. In the first case [8], the DT is seen as a service provided to end users, thus the challenge (from the network perspective) is to guarantee resources for the creation of the DT (storage, CPU, link capacity, etc.) and its synchronization with the physical system. In the second case, the DT technology is seen as a novel tool to improve the network Operation and Management (OAM) solutions. The present work goes in the last direction, thus in the next we dive into the literature to provide an overview of the current use DT in the context of network OAM.

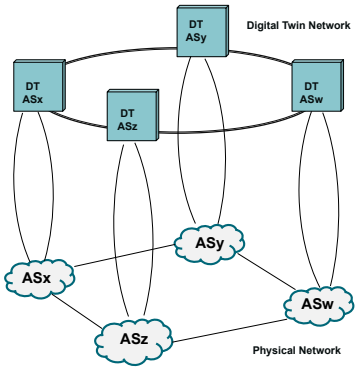

Figure 1. Example of DTN of an inter domain network scenario composed of four ASs.

A first interesting application of DTs in networking is introduced in [11], where the authors present a framework to coordinate concurrent applications (i.e., pursuing conflicting objectives) running on top of an SDN controller. The idea is that the effect of an action performed by an application is tested over a DT in order to evaluate if it degrades the KPI of other applications. Another interesting example is discussed in [12], where the DT technology is used to reduce the cost of applying a chaos engineering approach that evaluates the resiliency of a complex IT application realized over a hybrid cloud scenario. In [13] the DT of a network is realized using an approach based on Machine Learning techniques: in particular, a Graph Neural Network is trained in order to model the behavior of the real system in case of a change in the input state, thus allowing the execution of *what-if* analyses in real time. Two case studies are considered to validate the effectiveness of the approach, i.e., the prediction of the end to end delay and the routing optimization in order to meet QoS requirements. DTs are also applied in the context of network security. An example is reported in [14], where the DT technology is used to model the target system of an attack, allowing to learn dynamic security policy through a reinforcement learning approach, which exploits the DT to obtain a reward for the performed action.

III. The Architecture to Support the Creation of the Inter-AS DTN

In order to enable the execution of a *what-if* analysis to evaluate the impact of configuration changes in the BGP protocol, we aim to realize a DTN of an Inter Domain Network (IDN) scenario, i.e., a network realized by interconnecting different ASs. An illustrative example of such a DTN is depicted in Fig. 1, where an IDN and its DTN, i.e. its digital replica, are shown. More in detail, the DTN is composed by the interconnection of the DTs of every single AS. The DT of an AS is a digital representation of its control plane, i.e., the network

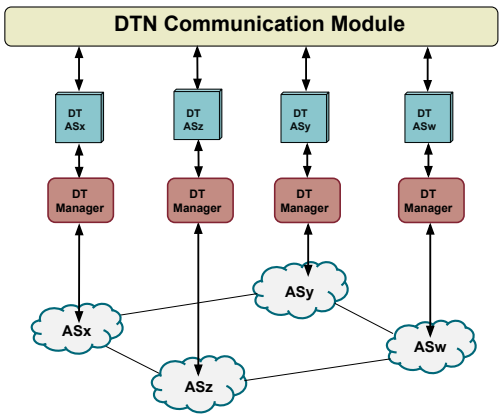

Figure 2. The architecture of the proposed framework.

topology, the devices configuration, the routing tables, etc. In particular, we exploit the Kathara framework [15] to create the digital replica of an AS. Kathara is an emulation platform based on Docker containers and Quagga software routers [16], which allows to easily reproduce a network topology in a single host machine. Thus, the DT of an AS is realized as a Kathara laboratory, where the topology of the AS is faithfully replicated and the configuration of each device (IP addresses, IGP and EGP routing protocol, etc.) is synchronized with the physical counterpart. Since all these details constitute a sensitive information of the AS that cannot be disclosed to other parties, in the proposed framework each AS is in charge of creating and managing its own DT.

When a Network Operator (NO) of an AS needs to perform a *what-if* analysis to evaluate the effect of a configuration change at the control plane level, it can interact with its own DT. If the targeted configuration change only concerns intra-AS routing, no other DTs are involved in the analysis. On the contrary, if the evaluation is related to a change in the inter-AS routing protocol, then the NO must involve also the DTs of the other ASs to obtain a feedback. For this reason, we propose an architectural framework to create a logical interconnection among the different DTs, to form a DTN.

Fig. 2 shows the logical entities of the proposed architecture:

- the *physical system*, which is composed of the different ASs and their interconnections. With reference to a single AS, the relevant information is represented by the topology and the routers configuration[1];
- the *Digital Twin Manager* (DTM), which is an entity controlled by the NO of an AS performing different tasks, such as the management of the life cycle of the DT and the exposition of an interface to perform the *what-if* analysis. Life cycle management includes the creation of a DT, its synchronization with the physical twin, and the termination of a DT after it has been used to perform a *what-if* analysis. The DTM also offers to the NO an interface to perform tests on the DTN, e.g, by changing the configuration of one (or more) device or by injecting messages to stimulate the creation of a feedback (e.g., inject an echo request toward a destination address);
- the *Digital Twin* (DT), consisting on an emulated network environment, that replicates the topology and the configuration of all the protocols and services in each device. For each AS there can be one or more DTs, in order to enable the execution of multiple *what-if* analyses. A DT can be in one of the following two states: i) *sync*, meaning that it is faithfully replicating its physical counterpart, and ii) *corrupted*, when the DT configuration deviates from its physical twin due to some changes done to perform the *what-if* analysis. A corrupted DT must be erased once the test is completed;
- the *DTN Communication module* (DCm), that handles the communication among all the different DTs, thus allowing the creation of the DTN. To perform a *what-if* analysis, the exchange of packets among DTs is needed and, since each DT is isolated from the others, the DCm extracts packets from a DT and injects them in different one/s;

In the following subsections we detail the processes of creating a DT and handling the inter DT communication.

A. Setting up a new DT of an AS

The logical entity in charge of executing the setup of a new DT of an AS is the DTM. The DTM obtains from the physical AS its routing-level information, such as the network topology, the configuration (specifically of the routing protocols) of every router of the AS, and the routing tables. Once all these inputs are available, according to the Kathara logic, the DTM generates a *lab.conf* and a set *device.startup* files. The first one is a description of the network topology, in terms of routers and links, while the latter contains the configuration commands (e.g., IP addresses, OSPF and BGP configuration, etc.) to be executed at the device boot phase. To realize a faithful replica of the physical AS, the routing tables of routers in the DT must be synchronized with the ones of real routers of the AS; in other words, the same routing convergence procedure must take place at DT and physical levels. In the case of IGP routing, the convergence is only a matter of time since the routers exchanging control messages are all part of the same AS.

In the case of BGP routing, the border routers of different ASs must communicate each other (e.g., BGP update exchange between peers) to converge. Thus the challenge is to let the DTM instantiate a new AS without cooperating with the neighbouring ASs. To this extent, the DTM continuously sniffs BGP update messages that are injected in the targeted AS and store them in a set of pcap files (one for each BGP peer of the AS). The BGP message sniffing phase is shown in Fig. 3. In particular, some BGP updates related to prefixes $P1$, $P2$, $P3$ and $P4$ are received by the border routers of the AS_y. These are captured by the DTM and stored inside different pcap files. When a new DT is created, the DTM merges the pcap files in a single one, trying to preserve the logical order of the

[1]In this work we focus our attention only on the Control Plane. In future works also Data Plane parameters (delays, traffic, queue, etc.) will be included.

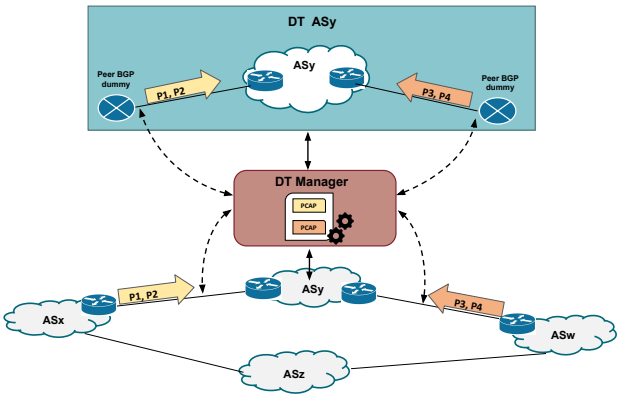

Figure 3. Example of BGP update injection in the DT performed by the DTM.

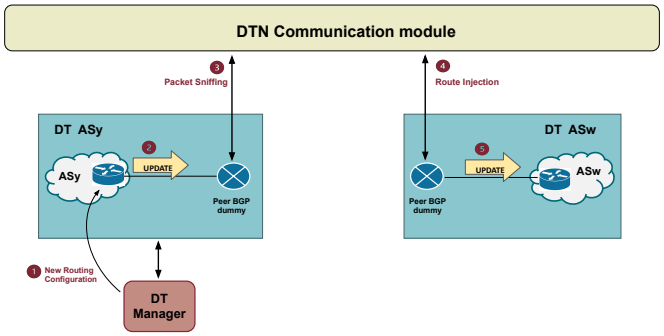

Figure 4. Example message exchange between two DTs performed by the DCm.

captured messages[2]. After that, the DTM sequentially injects the BGP updates in the DT, in order to let the routers learning the route for the advertised remote prefixes. In order to do this, at the Kathara lab creation step, the DTM includes in the AS topology some dummy nodes, with the aim of emulating external BGP peers. For instance, in the example shown in Fig. 3, two dummy BGP peers are inserted in the DT of the AS_y, in order to emulate the BGP sessions existing between the physical AS_y and its neighbours. This approach allows the routers in the DT of the AS_y to be aware of the routes for external prefixes, without the need to know the configuration of the routers of other ASs. To perform the message injection, in our prototype implementation, we have used the yaBGP tool [17].

B. The Inter DT Communication

The communication among different DTs is the key element toward the enabling of the *what-if* analysis in the context of the inter domain routing optimization. To explain how it is handled by the DCm, we refer to a use case, which is depicted in Fig. 4. The considered example is related to a BGP configuration change (point 1 in Fig. 4) performed by the AS_y on one of its routers (e.g., the advertisement of a new prefix). Such a configuration change induces the sending of a new BGP update

[2]This step is important, since the reception order is sometime used as a tie-breaker in the selection of the best route.

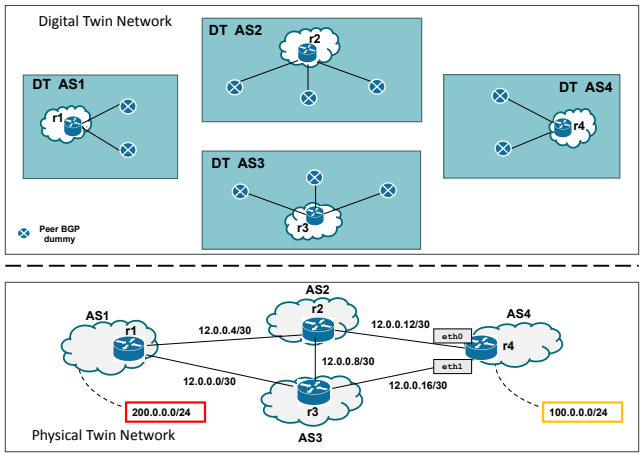

Figure 5. The reference scenario considered for the PoC.

toward its peer/s (point 2 in Fig. 4). In the context of the DT of the AS_y, the aforementioned update is sent toward the dummy BGP peer, thus it would not lead to the creation of any relevant feedback. Then, in order to let the update to be received by the DT of the peer AS_w, the DCm continuously sniffs the packets exchanged over the peering connection with the dummy BGP peer (point 3 in Fig. 4): every time it sees a packet being sent on that interface, it injects it in the next DT by means of the related dummy BGP peer (point 4 in Fig. 4).

IV. Proof of Concept

In this section we describe the PoC that has been realized to validate the effectiveness of the proposed framework. The reference scenario is the one reported in Fig. 5. The physical network is represented by a kathara lab and is composed of 4 ASs, each of which contains a single router. BGP peering relationships are established among neighbouring ASs. Furthermore, AS4 advertises the existence of the local prefix **P** $100.0.0.0/24$ (shown in yellow in the Fig. 5) by sending BGP updates to its peers. At the same time, 4 DTs are realized by the related DTM, and interconnected to form the DTN by means of the DCm (not shown in the figure).

Currently the traffic originated by the AS1 and destined to the prefix **P** enters the AS4 through the interface *eth0* of the router *r4*, as shown in the capture reported in Fig. 6, where it can be seen that packets originating by the address $200.0.0.1$ and destined to $100.0.0.1$ pass through the peering link connecting AS4 with AS2.

Then, in order to load balance the incoming traffic over the two peering links, the NO of the AS4 adopts the AS Path Prepending technique with respect to the BGP update sent for the advertisement of the prefix **P**. In particular, increasing the length of the AS Path advertised toward the router *r2*, could lead to a change in the best route adopted by the router *r1*, which could then prefer the route advertised by the router *r3*. Nonetheless, since AS4 does not know the BGP policies adopted by the other ASs, it cannot know in advance if this configuration change will lead to any change in the incoming

Capture on Interface eth0 of router r4 BEFORE applying the re-configuration

```
08:42:45.877362 IP 200.0.0.1 > 100.0.0.1: ICMP echo reply, id 46, seq 1, length 64
08:42:46.885400 IP 200.0.0.1 > 100.0.0.1: ICMP echo reply, id 46, seq 2, length 64
08:42:47.909452 IP 200.0.0.1 > 100.0.0.1: ICMP echo reply, id 46, seq 3, length 64
08:42:48.933430 IP 200.0.0.1 > 100.0.0.1: ICMP echo reply, id 46, seq 4, length 64
08:42:49.957425 IP 200.0.0.1 > 100.0.0.1: ICMP echo reply, id 46, seq 5, length 64
08:42:50.981446 IP 200.0.0.1 > 100.0.0.1: ICMP echo reply, id 46, seq 6, length 64
08:42:52.005424 IP 200.0.0.1 > 100.0.0.1: ICMP echo reply, id 46, seq 7, length 64
08:42:53.029401 IP 200.0.0.1 > 100.0.0.1: ICMP echo reply, id 46, seq 8, length 64
08:42:54.053419 IP 200.0.0.1 > 100.0.0.1: ICMP echo reply, id 46, seq 9, length 64
08:42:55.077308 IP 200.0.0.1 > 100.0.0.1: ICMP echo reply, id 46, seq 10, length 64
08:42:56.101460 IP 200.0.0.1 > 100.0.0.1: ICMP echo reply, id 46, seq 11, length 64
08:42:57.125363 IP 200.0.0.1 > 100.0.0.1: ICMP echo reply, id 46, seq 12, length 64
08:42:58.149431 IP 200.0.0.1 > 100.0.0.1: ICMP echo reply, id 46, seq 13, length 64
```

Capture on Interface eth1 of router r4 AFTER applying the re-configuration

```
08:42:59.173482 IP 200.0.0.1 > 100.0.0.1: ICMP echo reply, id 46, seq 14, length 64
08:43:00.197414 IP 200.0.0.1 > 100.0.0.1: ICMP echo reply, id 46, seq 15, length 64
08:43:01.221428 IP 200.0.0.1 > 100.0.0.1: ICMP echo reply, id 46, seq 16, length 64
08:43:02.245430 IP 200.0.0.1 > 100.0.0.1: ICMP echo reply, id 46, seq 17, length 64
08:43:03.269426 IP 200.0.0.1 > 100.0.0.1: ICMP echo reply, id 46, seq 18, length 64
08:43:04.293425 IP 200.0.0.1 > 100.0.0.1: ICMP echo reply, id 46, seq 19, length 64
08:43:05.317426 IP 200.0.0.1 > 100.0.0.1: ICMP echo reply, id 46, seq 20, length 64
08:43:06.341443 IP 200.0.0.1 > 100.0.0.1: ICMP echo reply, id 46, seq 21, length 64
08:43:07.365449 IP 200.0.0.1 > 100.0.0.1: ICMP echo reply, id 46, seq 22, length 64
08:43:08.389433 IP 200.0.0.1 > 100.0.0.1: ICMP echo reply, id 46, seq 23, length 64
```

Figure 6. Packet sniffing performed on the interfaces of router *r4* in the DT.

routes. In fact, router *r3* could have a higher local preference for the BGP updates received from AS2 than that related to AS4, or AS2 could filter updates advertised toward AS1, since it does not want to serve as a transit.

Thus, in the following we show the result of the *what-if* analysis performed by the NO of AS4, to obtain a feedback about an AS Path Prepending operation performed on BGP updates sent to AS2 and related to the prefix **P**. The analysis is triggered by a configuration change performed by the NO of the AS4 on the router *r4* of the related DT. The new configuration commands are the following ones:

```
neighbor 12.0.0.14 route-map PREPEND out
route-map PREPEND permit 10
   set as-path prepend 4 4 4
```

In particular, a new *route-map* is defined with the aim to insert three times the AS4 on the AS Path attribute of BGP Updates. This operation is performed only in case of updates sent to the router *r2* in AS2. Once the configuration change is performed, the router *r4* in the DT is forced to generate new BGP updates related to the prefix **P** toward all its neighbors, by issuing the following command:

```
clear ip bgp * soft out
```

This command triggers then the actual execution of the *what-if* analysis in the DTN. In particular, in this phase the ASs exchange routing information until the process converge and the new best routers to reach the prefix **P** are computed. This step is governed by the DCm module, which continuously transfer BGP updates among the DTs. Once the new best routes are available, the last step is to evaluate the effect of the configuration change. To this extent, the NO of the AS4 injects in the DTN through the DTM an ICMP echo request directed to a destination address belonging to AS1, i.e., 200.0.0.1. In the meantime, it also starts sniffing traffic on interfaces *eth0* and *eth1* of the router *r4* in the DT, in order to understand from which of these two interfaces is received the traffic originated by the AS1. The result of this test is reported in Fig. 6, where it is shown that the related ICMP echo reply are received through the interface *eth1*. Thus, the NO of the AS4 knows that the prepending operation is effective in changing the best path of the AS1 for the prefix **P**.

A. Discussion

In this subsection we discuss about the role of the DT technology in the context of a *what-if* analysis for the optimization of the inter domain routing. The first point to be discussed is the difference between a classical emulation tool and the proposed DTN based framework. Clearly, emulation plays a crucial role in this regards; we used Kathara to create the DT of a single AS. Nonetheless, the simple emulation would not be enough to perform the *what-if* analysis discussed in the presented PoC. This is because a single NO does not have the configuration details of the other ASs, thus it could not setup an emulation environment by only relying on the information it knows. In this perspective, the DTN is then the interconnection of different emulators that replicate the single ASs, each one realized and managed by a different NO: each AS owner exposes its DT to the others as if it is a sort of interface.

This last point raises another concern, which is related to the possibility for different NOs to collaborate in order to realize the DTN. We believe that the opportunities provided by the possibility to realize *what-if* analyses could push the NOs in federating to create the DTN. Thanks to the architectural choice of keeping each DTM owned and managed by a single AS, the internal configuration of the ASs is kept confidential. This point is crucial for its widespread adoption from the different NOs. At the same time, similarly as it happens in case of Internet Exchange Points (IXPs), it is also needed the introduction of a third party that is in charge of managing the DCm.

V. Conclusion and Future Work

In this paper we have presented a framework to realize a Digital Twin Network with the aim of enabling the execution of *what-if* analises in the context of the inter domain routing optimization. The proposed framework is based on two main building blocks: the DTM and the DCm. A proof of concept, related to the prediction of the effects of a AS Path Prepending strategy used to load balance the incoming traffic over multiple peering connections, has been realized to show the effectiveness of the proposed approach. As next steps, we aim to introduce also the realization of the DT of the data plane of each AS. This will allow to get further feedback, such the estimation of the RTT and the amount of traffic that is moved by a configuration change.

References

[1] Y. Rekhter, S. Hares, and T. Li, "A Border Gateway Protocol 4 (BGP-4)," RFC 4271, Jan. 2006. [Online]. Available: https://www.rfc-editor.org/info/rfc4271

[2] B. Quoitin, C. Pelsser, L. Swinnen, O. Bonaventure, and S. Uhlig, "Inter-domain traffic engineering with bgp," *IEEE Communications magazine*, vol. 41, no. 5, pp. 122–128, 2003.

[3] P. Marcos, L. Prehn, L. Leal, A. Dainotti, A. Feldmann, and M. Barcellos, "As-path prepending: there is no rose without a thorn," in *Proceedings of the ACM Internet Measurement Conference*, 2020, pp. 506–520.

[4] S. Janardhan, "More details about the october 4 outage," Oct 2021. [Online]. Available: https://engineering.fb.com/2021/10/05/networking-traffic/outage-details/

[5] K. Sriram, D. Montgomery, D. R. McPherson, E. Osterweil, and B. Dickson, "Problem Definition and Classification of BGP Route Leaks," RFC 7908, Jun. 2016. [Online]. Available: https://www.rfc-editor.org/info/rfc7908

[6] "Massive route leak causes internet slowdown." [Online]. Available: https://www.bgpmon.net/massive-route-leak-cause-internet-slowdown/

[7] N. Feamster, J. Winick, and J. Rexford, "A model of bgp routing for network engineering," *ACM SIGMETRICS Performance Evaluation Review*, vol. 32, no. 1, pp. 331–342, 2004.

[8] Y. Wu, K. Zhang, and Y. Zhang, "Digital twin networks: A survey," *IEEE Internet of Things Journal*, vol. 8, no. 18, pp. 13 789–13 804, 2021.

[9] R. K. Chang and M. Lo, "Inbound traffic engineering for multihomed ass using as path prepending," *IEEE network*, vol. 19, no. 2, pp. 18–25, 2005.

[10] W. Mühlbauer, A. Feldmann, O. Maennel, M. Roughan, and S. Uhlig, "Building an as-topology model that captures route diversity," *ACM SIGCOMM Computer Communication Review*, vol. 36, no. 4, pp. 195–206, 2006.

[11] M. Polverini, F. G. Lavacca, J. Galán-Jiménez, D. Aureli, A. Cianfrani, and M. Listanti, "Digital twin manager: A novel framework to handle conflicting network applications," in *2022 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*, 2022, pp. 85–88.

[12] F. Poltronieri, M. Tortonesi, and C. Stefanelli, "A chaos engineering approach for improving the resiliency of it services configurations," in *NOMS 2022-2022 IEEE/IFIP Network Operations and Management Symposium*, 2022, pp. 1–6.

[13] P. Almasan, M. Ferriol-Galmés, J. Paillisse, J. Suárez-Varela, D. Perino, D. López, A. A. P. Perales, P. Harvey, L. Ciavaglia, L. Wong, V. Ram, S. Xiao, X. Shi, X. Cheng, A. Cabellos-Aparicio, and P. Barlet-Ros, "Network digital twin: Context, enabling technologies, and opportunities," *IEEE Communications Magazine*, vol. 60, no. 11, pp. 22–27, 2022.

[14] K. Hammar and R. Stadler, "An online framework for adapting security policies in dynamic it environments," in *2022 18th International Conference on Network and Service Management (CNSM)*. IEEE, 2022, pp. 359–363.

[15] M. Scazzariello, L. Ariemma, and T. Caiazzi, "Kathará: A lightweight network emulation system," in *NOMS 2020 - 2020 IEEE/IFIP Network Operations and Management Symposium*, 2020, pp. 1–2.

[16] P. Jakma and D. Lamparter, "Introduction to the quagga routing suite," *IEEE Network*, vol. 28, no. 2, pp. 42–48, 2014.

[17] "Yet another bgp python implementation." [Online]. Available: https://github.com/smartbgp/yabgp