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

M. Polverini, I. Germini, A. Cianfrani, F.G. Lavacca, M. Listanti

“Sapienza” University

RIPE86 - MAT-WG Workshop

Outline

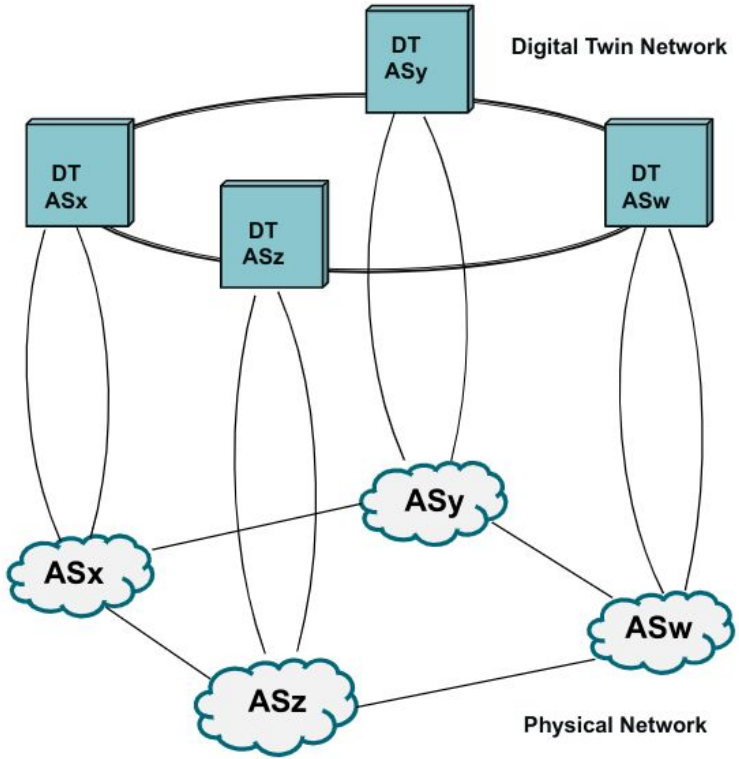
- Motivation
- Digital Twin
- Use Cases
- Framework overview
- Proof of Concept

The Tweak and Pray paradigm

- Common approach used to configure/optimize BGP configuration is based on the Tweak and Pray (T&P) paradigm
  - **perform the change and see if the situation has improved**
  - it might cause network instability, performance degradation, service disruption
- Wouldn't be great if the effect of a BGP configuration change could be known in advance?
- This is known as "What If" analysis
- In different engineering sectors, the execution of "what if" analysis is enabled by the Digital Twin (DT) concept
- We propose a framework to create/manage/operate the DT of a BGP ecosystem

Digital Twin

- A Digital Twin is a faithful replica of a Physical System
- The DT of an AS is represented by an **emulated network** replicating the **devices configuration**
- In [1] we present a framework for the creation of the DT of an Inter-AS ecosystem (e.g. the Internet)



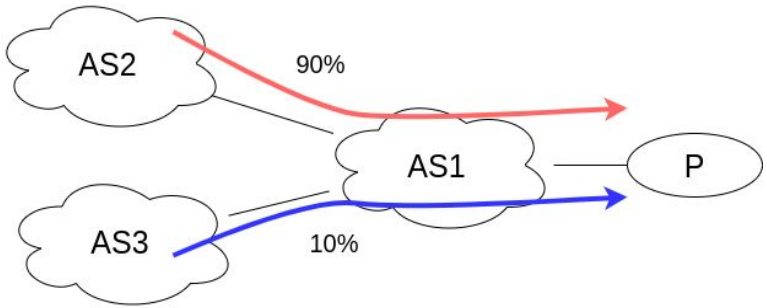
[1] M. Polverini, I. Germini, A. Cianfrani, F.G. Lavacca, M. Listanti, "A Digital Twin based Framework to Enable ``What-If'' Analysis in BGP Optimization," NOMS 2023 - 2023 IEEE/IFIP Network Operations and Management Symposium, Miami, USA, 2023

Use Cases

- Some concrete use cases where “what if” analysis can help
  - Balancing the Incoming Traffic
  - Create a new peering/transit connection
  - Avoid misconfiguration events

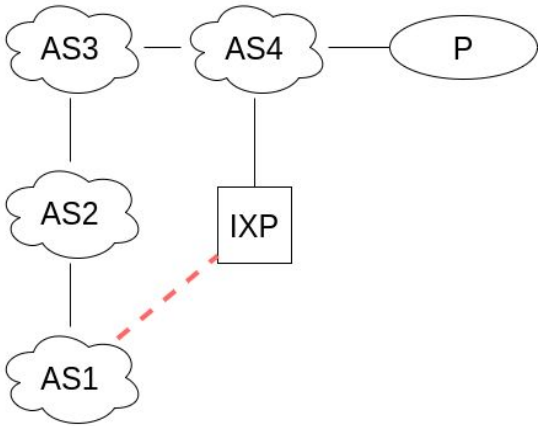
Use Cases

- Some concrete use cases where “what if” analysis can help
  - Balancing the Incoming Traffic
  - Create a new peering/transit connection
  - Avoid misconfiguration events



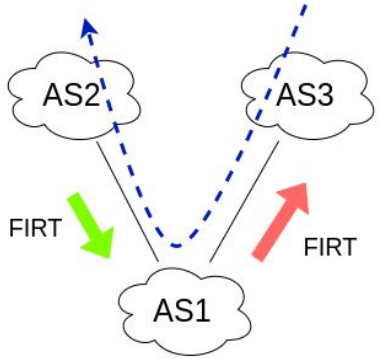
Use Cases

- Some concrete use cases where “what if” analysis can help
  - Balancing the Incoming Traffic
  - Create a new peering/transit connection
  - Avoid misconfiguration events



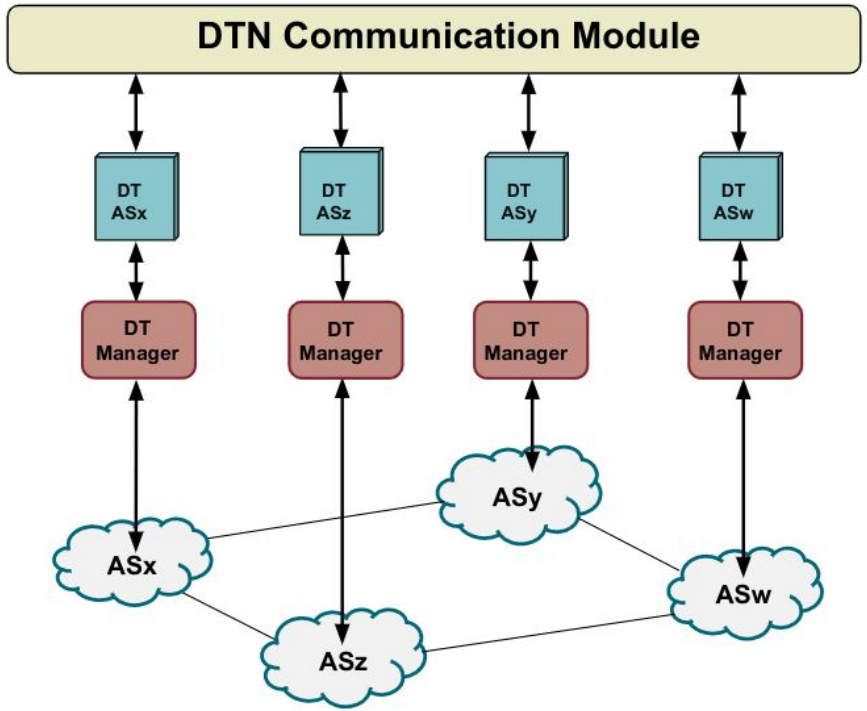
Use Cases

- Some concrete use cases where “what if” analysis can help
  - Balancing the Incoming Traffic
  - Create a new peering/transit connection
  - Avoid misconfiguration events



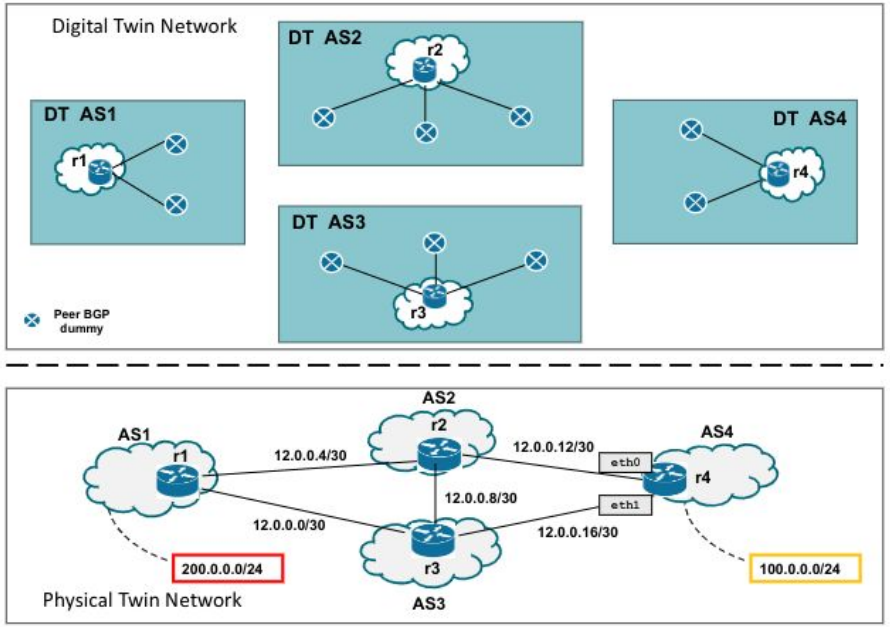
Framework Architecture

- The proposed framework is based on the following components:
  - **Digital Twin Manager**
    - responsible for lifecycle management of a DT
  - **Digital Twin**
    - an emulated network that replicate the control plane configuration of the physical AS
  - **DTN Communication module**
    - needed to interconnect the different DTs during the execution of a “what if” analysis



Proof of Concept

- Kathara network emulator is used to realize either the PTs and the DTs
  - The entire scenario runs in a single host machine
- **AS4 uses the AS Path Prepending technique to (try) to change the path used by AS1 to reach the prefix 100.0.0.0/24**
- What If analysis is executed in the DTN



Proof of Concept

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

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
```