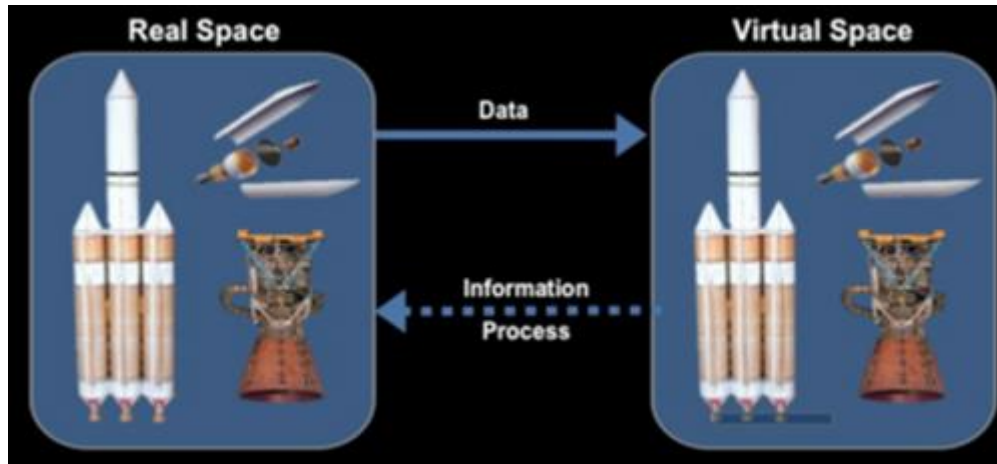# Digital Twin and Automation

A network digital twin models certain aspects of a real network in the cyberspace. One main value for that model is to run what-if scenarios without an impact on the real world.


Depending on which aspect of the network we want to observe, a model must be built based on the most suited technology. We are currently experimenting with Machine Reasoning, Machine Learning and AI to extract information and predict outcomes.
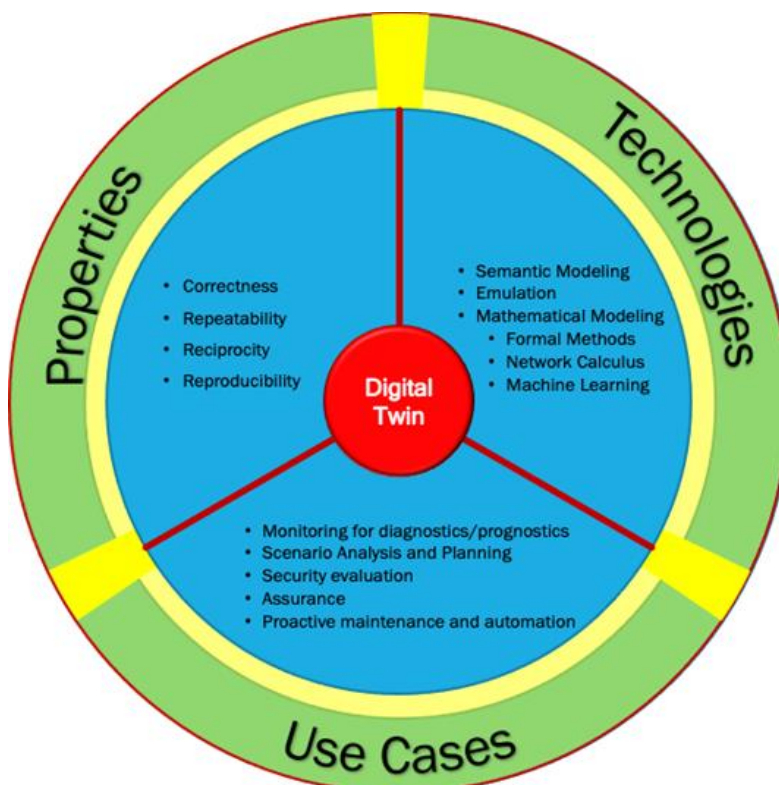
In a network model, it is typically possible to reroute flows, add access control and traffic filter rules, power down devices, and query all sorts of metrics that are available to the model. Typical metrics would be the load on link, interfaces, and devices, and typical what-if scenarios would incur changing flow paths and volumes and observe the impact on the network.

Metrics can be modeled on a per interface or per device level and can be aggregated along a path. For instance, if a flow is capable of congestion control, the model may find the most constrained pipe along the path of the flow and compute the throughput as what that pipe lets through.

Now consider the power of the digital twin applied to sustainability. One may model the hardware and flows in such a manner that instead of aggregating volumes of data, the model would compute the energy spent in a hardware component for a flow, and then report the total cost of energy of that flow in a line card or in the whole router aggregating all components involved in processing that flow, and then aggregate along the path of the flow adding up the cost in all routers on the actual path of the flow. The twin would also produce the total energy consumption of a router by aggregating all flows and the total energy consumption of the whole network if desired.

In a sustainability-oriented network digital twin, the extra cost of rerouting flows in the additional hops could be pondered with the saving of powering down a network device, based on the current load, an ML-based automation could control if / when that happens, or at least make recommendations in an open loop.
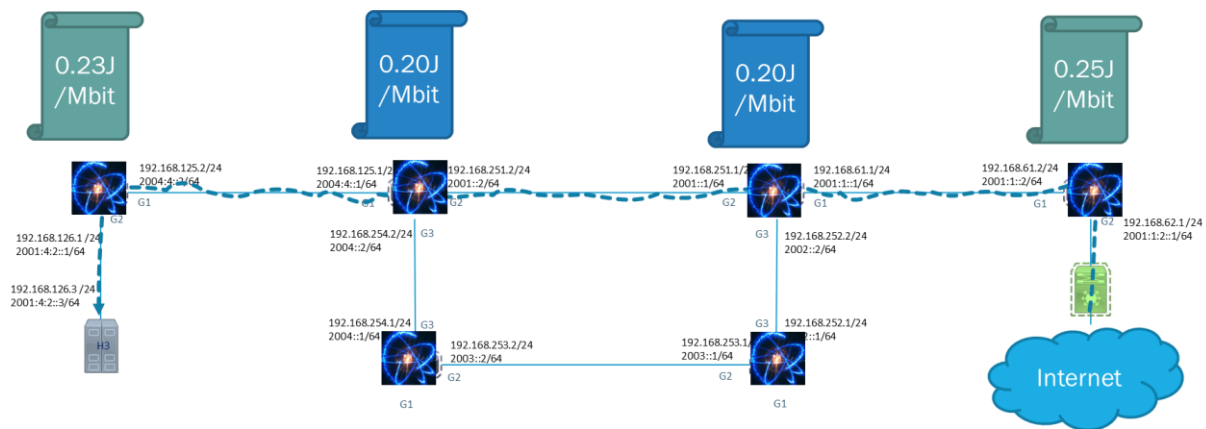
Clearly, the result of a computation is only as good as the models that sustain it. The digital twin provides a framework where the model for energy consumption could be estimated by aggregated modeled consumption atoms, and the energy aspect that models those atoms could leverage a technology that is best suited for that aspect, even if different from the that of the framework, e.g., machine learning.

Also clearly, congestion collapse yields not only wasted bandwidth but also wasted energy. A packet dropped in congestion loss is energy wasted from the server to the drop point. Another aspect of the twin could model the transport layer and the congestion control operation, and help understanding the impact of congestion control on energy consumed per delivered packet. Combining those aspects could help an operator optimize his throughput for energy, and make decisions such as shape and/or reroute early in the network for an optimal result.

To enable such functionality across platforms and vendors, there is a need to provide comparable models for energy consumption and provide the values for a device that can be compared with the model. Provided that a standard exists to produce real world measurements in telemetry attribute compatible with the model, those computed values could be compared with the estimation in the twin, to update and correct the twin till it is a correct mimic of the real world.