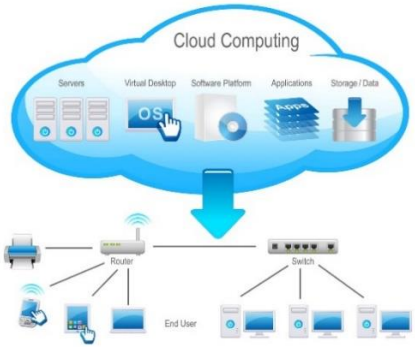
Skydive Noisy-Neighbor Detection

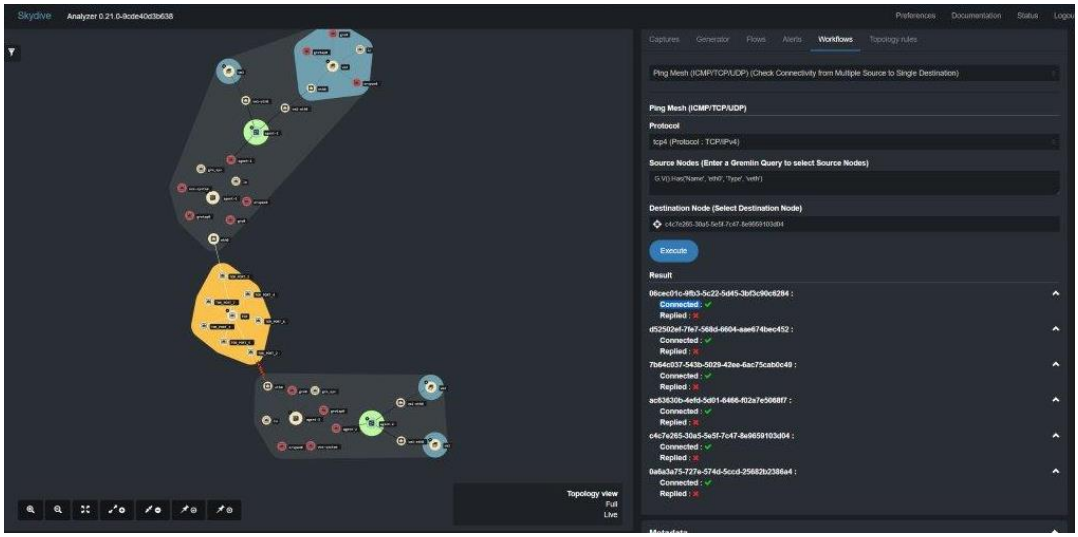
Abstract:

In the past, businesses and firms spent large capital investments on locally hosting of software and Information Technology (IT) infrastructure. The emerging Cloud computing technology is a big shift from this traditional way of thinking about IT resources.

Cloud computing enables standard access to shared pools of configurable system resources and higher-level services that can be rapidly provisioned with minimal management effort, often over the Internet. It relies on sharing of resources to achieve coherence and economies of scale, similarly to a public utility.



The main enabling technology for cloud computing is virtualization. Virtualization software separates a physical resource into one or more "virtual" resources, each of which can be easily used and managed to perform computing tasks. Cloud network is one such resource and is virtualized through multiple logical constructs and SW layers. It is essential to monitor cloud network to ensure transparency, correct and efficient operations, both from the customer and from the provider perspectives. Skydive is an open source real-time network topology and protocols analyzer. It aims to provide a comprehensive way of understanding what is happening in the network infrastructure.



Milestones

1. Installation and learning
    a. Get access to the server in CS cloud with IBM Cloud Private (ICP) ICP and SkyDive
    b. Review last semester project in https://gitlab.cs.technion.ac.il/lccn/w2018-ibm-skydive-noisy-neighbor
    c. Install on the server: ICP community edition, Skydive, kubectl

    Expected outcome: demonstrate capturing traffic with Skydive command line

2. Noisy neighbor generation
    a. Understand "noisy neighbor" use case
    b. Create traffic generator of "noisy neighbor" - TBD
        i. Optional: use iperf – inside each container for different tenant (user)
        ii. Optional: suggested topology with UDP/TCP traffic for unfairness

    Expected outcome: demonstrate noisy neighbor in Skydive GUI

3. Noisy neighbor detection
    a. Generate control traffic for telemetry gathering - Skydive native APIs (python) or Learn Skydive workflows (javascript)
    b. Build Skydive workflow for detecting noisy neighbor
        i. Ping-mesh – detect outstanding high latency. This will require pings between each source-destination pair
        ii. Outlier detection – Find the root cause, i.e - who is the noisy neighbor
    c. Output Ping-mesh - either print table or create html

    Expected outcome: demonstrate ping-mesh for noisy neighbor and outlier detection

4. Summary
    a. Demo
    b. Documentation/blog
    c. Skydive github contribution

Requirements:

Introduction to Networking (236334), Internet Networking (236341)

Pyhton or Golang

Guided by:

Eran Raichstein – Cloud Architect

Dr. Anna Levin – Research Staff Member