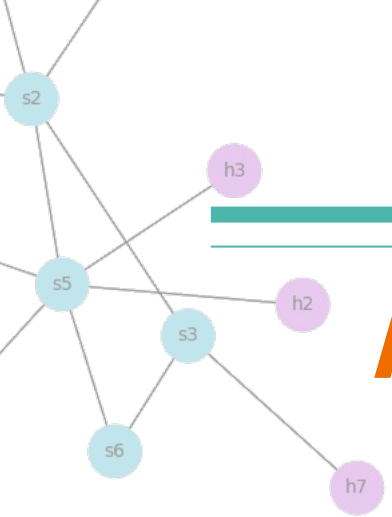
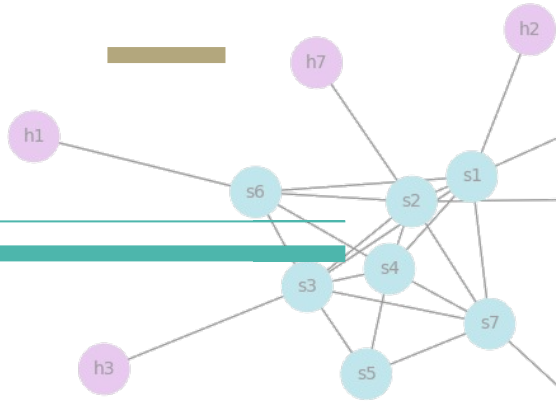
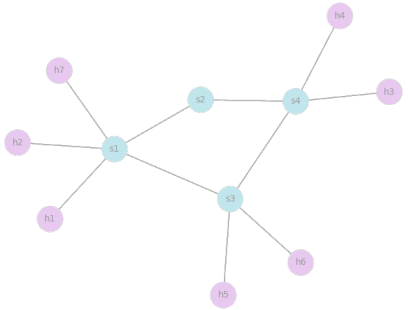


Automatic Deployment of ““Containers””

Project by Berardo C., Castagnaro T., Consolaro J.



Introduction



Objective

Develop an **SDN-based network** using Mininet with multiple switches and automate the deployment of services.

Key Features

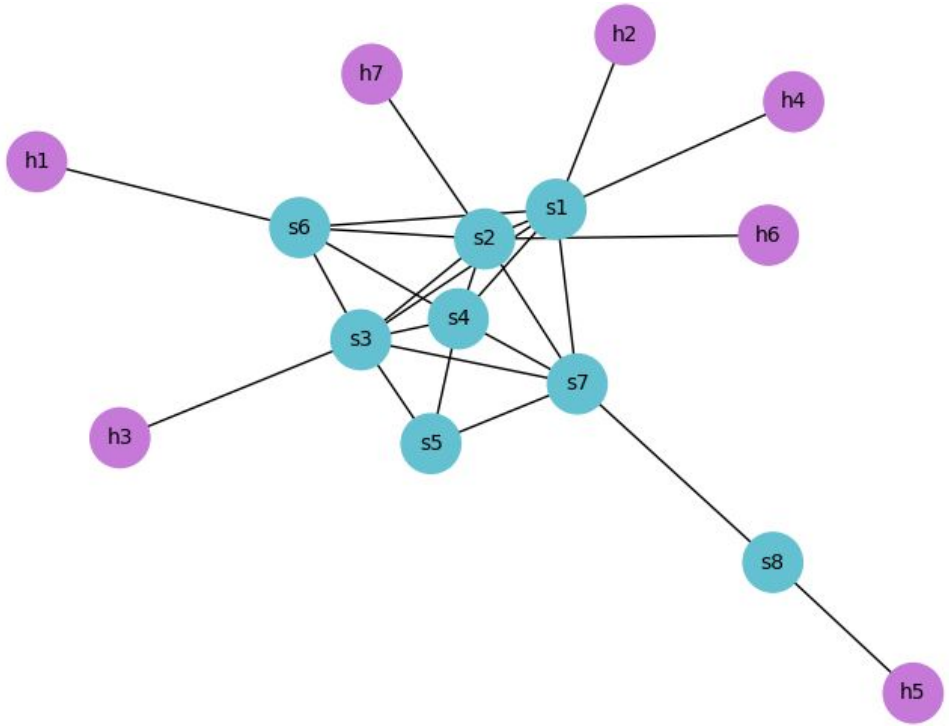
- Topology creation
- Graphical User Interface (GUI)
- Flows Creation and Deletion

Outcome

A flexible and automated SDN environment capable of managing application deployment and communication dynamically.

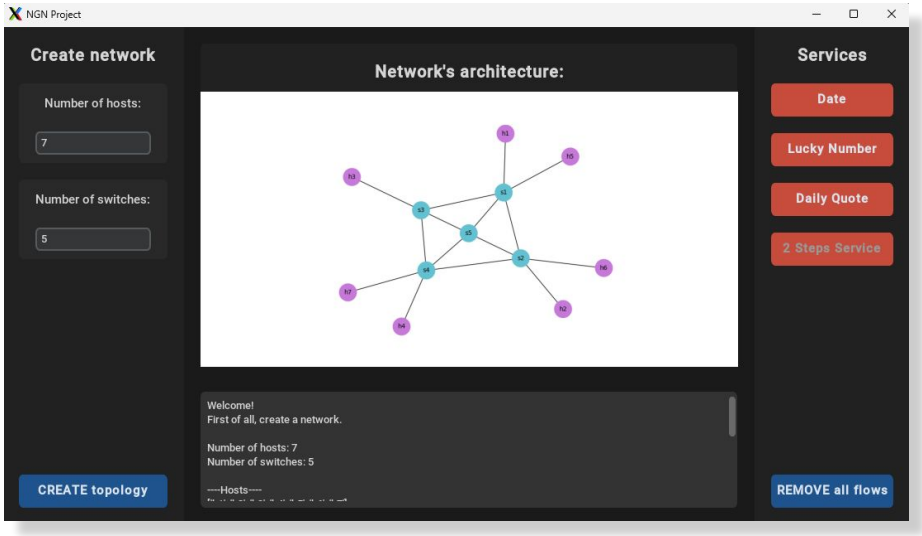
Key-Features: Topology Creation

- **Tools and libraries**:
  - Mininet
  - NetworkX
  - Matplotlib
- **Design Choices**:
  - N. of Hosts (at least 7)
  - N. of Switches (at least 1)
  - Loops
  - Level of randomization
- **topology_generator.py features**:
  - Topology generation
  - Run the Network
  - Services allocation
- **Not-used additional features**:
  - *get_shortest_path*(...) with Dijkstra's algorithm.

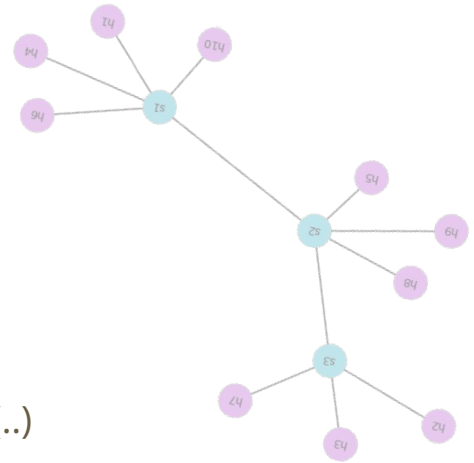


Key-Features: Graphical User Interface

- **Main library: CustomTkinter**
  - Python 3.7 is required!
  - Modern looking and customizable widgets
- **Sections**
  - Sidebar on the left for network creation
  - Sidebar on the right for services activation
    Status Box that shows services' outputs and progresses during the running process
  - An image on the center displays the structure of the topology
- **Interactions with other files**
  - topology_generator.py
- **Dynamic scaling**
  - The application can change its size keeping a coherent distribution of the elements

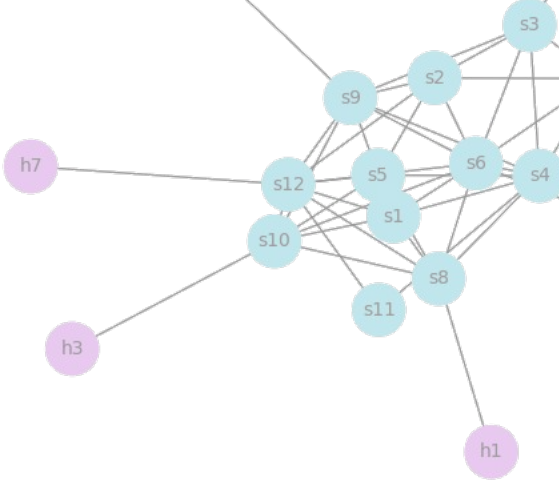


Key-Features: Flows Creation and Deletion



- **Description**: we have used the simple switch_13_stp extend simple_switch_13 and add one function to delete the flows
- **Design choices**: using HTTP Post to get the request of *delete_flood_flows*(..)
- **Tools and libraries used**: Response
- **Adding flows (__packet_in_handler)**:
  - 00:00:00:00:00:01→ host client
  - 00:00:00:00:00:03 → Date
  - 00:00:00:00:00:04 → Lucky number
  - 00:00:00:00:00:06 → Daily quote
  - 00:00:00:00:00:07 → Date + Time (2 step service)

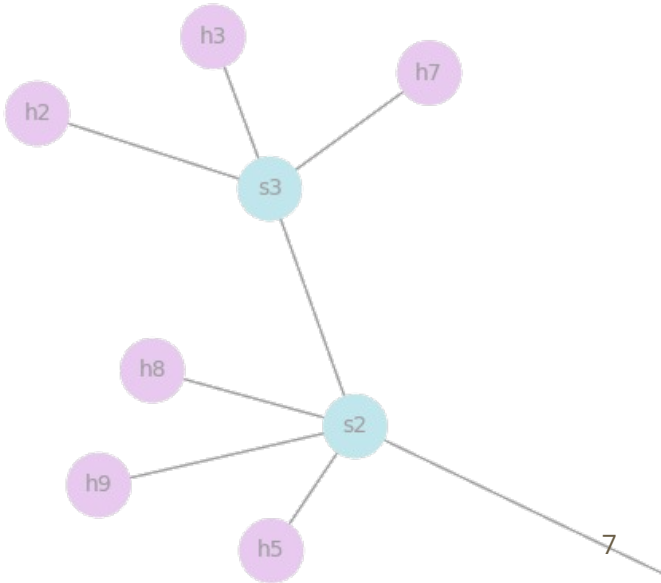
Challenges



- Creation, deletion of the Flows and dynamic updating of Flow Tables
- Instantiation of the services
- Creation of sockets to enable communication between entities
- Python compatibility with different libraries (Python3.7 and Python3.6)
- Dynamic update of the application
- Degree of randomness when creating the topology
- Developing for two OSs (Windows 11 & MacOS)

Future implementations

- Integrate shell into the GUI application
- Automatic interaction with Mininet from GUI



Thanks You For Your Attention!