

## Static Predefined Routing

- Ryu controller doesn't have information about the topology ( such as hop counts, Link information ,bandwidth, delay, information of neighboring nodes, etc)
- Since the topology is small and fixed, the programmer knows the hop counts in the topology. So, based on the minimum hop counts in the topology, the predefined primary routes are written in the script files of the edges and the alternative routes used after the link fails are written at northbound interface of the Ryu controller. If the topology is large and not fixed, it is so much overloaded for the programmer to predefine all the alternative routes based on the hop counts. – *one of the weaknesses of static predefined routing*  
Based on the above two facts, we call it “**static predefined routing**”.
- Ryu controller can only detect the failures of the links, not the information of the link costs. After one link is failed, Ryu controller assigns the flowrules to the respective edges based on the predefined alternative routes no matter the selected alternative route has a lot of delay or less bandwidth ( because the Ryu controller doesn't have the information of link costs). – *main weakness of static predefined routing*

### How the Ryu controller detects the link failure?

Ryu controller sends the Openflow flowstats request messages to the edges. Then, the edges reply the Openflow flowstats reply messages to the Ryu controller. Ryu controller now gets the information about the packet counts and byte counts in each respective flow from the Openflow flowstats reply messages. Ryu controller does that procedure in each defined time interval. Then, Ryu controller knows that each link is failed or not based on number of the packet counts and byte counts in each flow. If the packet count and byte count are still the same in the previous time period and in the current time period in the flow even though the packets are still generating, then, the Ryu controller knows that that link is failed.

### How to verify that our static predefined routing program works?

We need to simulate traffic and simulate link failure cases by ourselves to verify the functionalities of our static predefined routing program. The reason why we need to simulate traffic and simulate link failures is that we are testing only in the small room in our lab where there are no link blocks that can happen link failures and we are not using the real traffic yet (later we use IoT data).

To generate traffic, we use Iperf3.

To simulate link failures, we manually shutdown the wireless interfaces of the edges so that there will be no traffic receiving in the respective flow and Ryu controller sees it as it is link failure.

After link failures are happened, Ryu controller assigns the flowrules to the respective edges to establish the predefined alternative routes.

### **What kind of results we expect from static predefined routing?**

- 1> Predefined alternative routes work. ( can be verified just by Ping results)
- 2> The selected alternative route is not the optimum route every time because Ryu controller just chooses the predefined alternative route written by the programmer based on the hop counts in the topology. ( can be proved by using Ping result's RTT values, we just simulate more traffic on the selected alternative route and shows that this alternative route cannot be the best route every time.)

*It is not like Ryu controller knows all the link information of the topology dynamically and periodically and decides the optimum route based on the learnt link information –thus why, later, we use what we call “dynamic routing”.*