Zostaly zaimplementowane 2 algorytmy routingu:
- ShortPathRouter, czyli opcja z distance vectorem, u mnie jest to slownik (router_id : (dystans, czas, router_posredniczacy)), ktory aktualizuje w ramach przychodzacych distance vectorow od sasiadow. Usuwanie krawedzi powoduje ustawienie dystansu jako -dystans (w sensie distance = -distance jesli distance < 0). Informacje z distance vectorow sasiadow sa przetwarzane tylko jezeli byly conajmniej tak samo aktualne jak obecna chwila. W przypadku dostania informacji o usunietej krawedzi, zmieniamy nasz distance vector w przypadku gdy krawedz uczestniczyla w najkrotszej sciezce.
- the_whole_graph, czyli opcja w przechowywaniem calego grafu w kazdym routerze, jest to przechowywane jako lista list i w czasie usuwania krawedzi jest zmienany pojedynczy bool w odpowiedniej liscie.
Przesylanie pakietu odbywa sie za pomoca bfs-a, ktory wyszukuje kolejny wierzcholek, stojacy na najkrotszej sciezce do packet.dst, czyli naszego celu.
Wykorzystuje on set used_links, czyli zbior linkow, ktore już wykorzystaliśmy w danej turze i dict predecessors, czyli slownik optymalnych poprzednikow, zeby wiedziec w ktora strone poslac dany pakiet (o ile wiemy ze mozna go poslac)