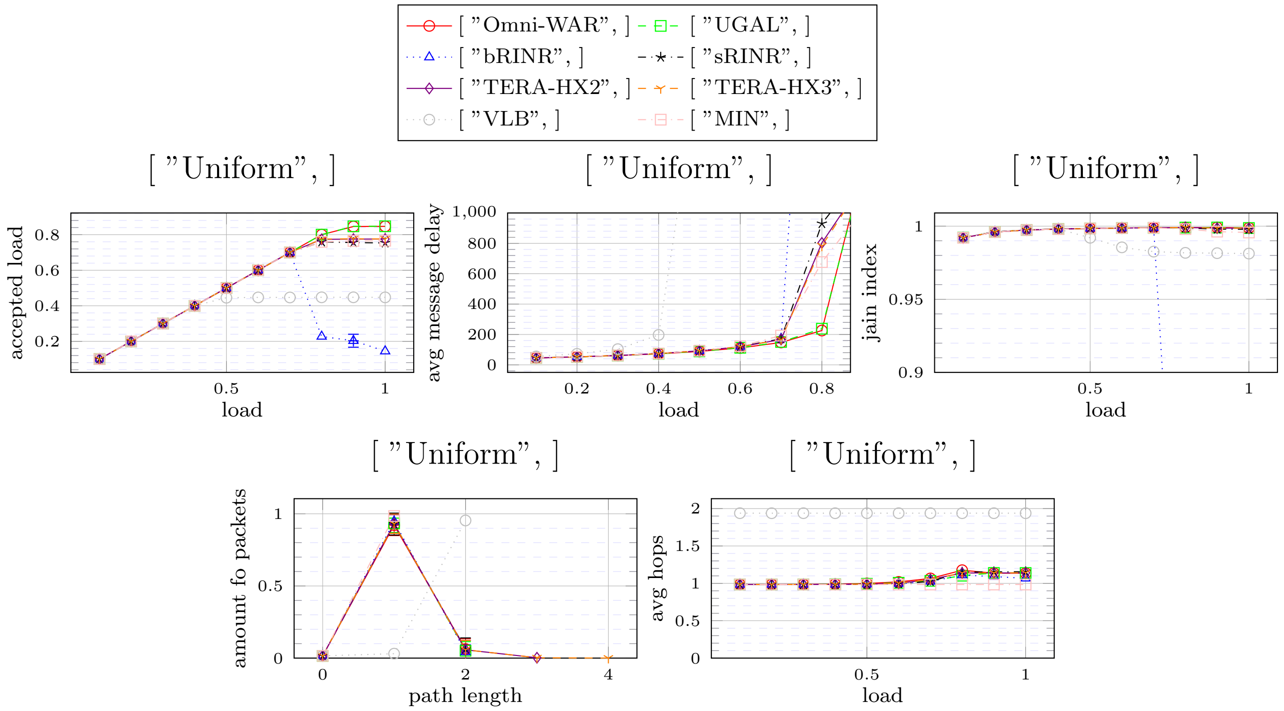

Figure 1: X: ["Uniform",]

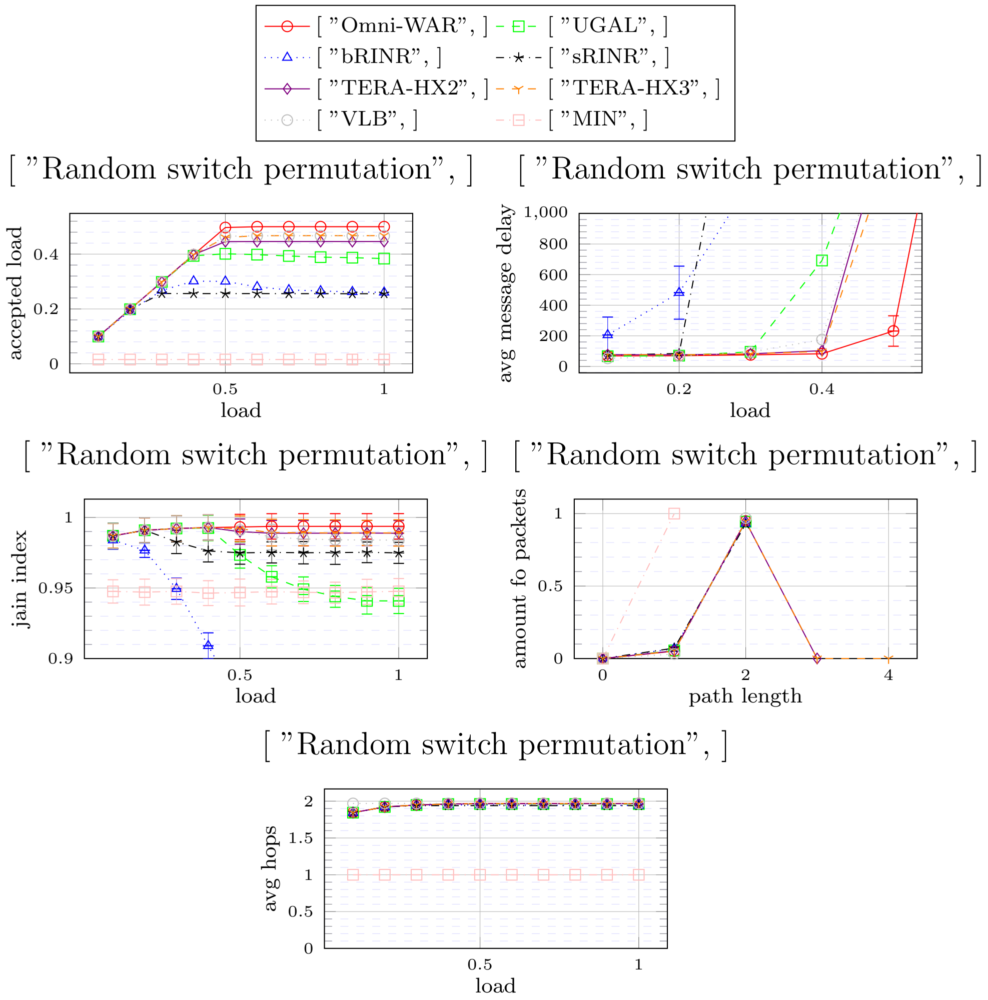

Figure 2: X: ["Random switch permutation",]

The following versions used in the simulations.

- heads/alex-stable-release-TERA-35e794829433712d102999df79a051a0256a860a(0.6.3)

```
Configuration{
  random_seed: ![ 1, 2, 3 ],
  warmup: 40000,
  measured: 40000,
  statistics_server_percentiles: [ 0, 5, 25, 50, 75, 95, 100 ],
  statistics_packet_percentiles: [ 0, 5, 25, 50, 75, 95, 100 ],
  general_frequency_divisor: 2,
  statistics_temporal_step: 1000,
  topology: Hamming{
    servers_per_router: 64,
    sides: [ 64 ]},
  traffic: HomogeneousTraffic{
    pattern: ![
      Uniform{ legend_name: "Uniform" },
      Product{ block_size: 64, global_pattern: RandomPermutation, block_pattern: Identity, legend_name: "Random switch permutation" }],
    tasks: 4096,
    load: ![ 0.05, 0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45, 0.5 ],
    message_size: 16},
  router: InputOutput{
    virtual_channels: mecanismo![ 2, 2, 1, 1, 1, 1, 2, 1 ],
    virtual_channel_policies: mecanismo![
      [
        WideHops{ width: 1 },
        VecLabel{
          label_vector: [ 0, 64 ]},
        OccupancyFunction{ label_coefficient: 1, occupancy_coefficient: 1, product_coefficient: 0, constant_coefficient: 0, use_internal_space: true,
use_neighbour_space: true, aggregate: true },
        OccupancyFunction{ label_coefficient: 1, occupancy_coefficient: 1, product_coefficient: 0, constant_coefficient: 0, use_internal_space: true,
use_neighbour_space: true, aggregate: false },
        LowestLabel,
        EnforceFlowControl,
        Random],
      [
        VecLabel{
          label_vector: [ 0, 64 ]},
        OccupancyFunction{ label_coefficient: 1, occupancy_coefficient: 1, product_coefficient: 0, constant_coefficient: 0, use_internal_space: true,
use_neighbour_space: true, aggregate: true },
        OccupancyFunction{ label_coefficient: 1, occupancy_coefficient: 1, product_coefficient: 0, constant_coefficient: 0, use_internal_space: true,
use_neighbour_space: true, aggregate: false },
        LowestLabel,
        EnforceFlowControl,
        Random],
      [
        VecLabel{
          label_vector: [ 0, 56 ]},
        OccupancyFunction{ label_coefficient: 1, occupancy_coefficient: 1, product_coefficient: 0, constant_coefficient: 0, use_internal_space: true,
use_neighbour_space: true, aggregate: true },
        LowestLabel,
        EnforceFlowControl,
        Random],
      [
        VecLabel{
          label_vector: [ 0, 56 ]},
        OccupancyFunction{ label_coefficient: 1, occupancy_coefficient: 1, product_coefficient: 0, constant_coefficient: 0, use_internal_space: true,
use_neighbour_space: true, aggregate: true },
        LowestLabel,
        EnforceFlowControl,
        Random],
      [
        VecLabel{
          label_vector: [ 0, 56 ]},
        OccupancyFunction{ label_coefficient: 1, occupancy_coefficient: 1, product_coefficient: 0, constant_coefficient: 0, use_internal_space: true,
use_neighbour_space: true, aggregate: true },
        LowestLabel,
        EnforceFlowControl,
        Random],
      [
        VecLabel{
          label_vector: [ 0, 56, 56 ]},
        OccupancyFunction{ label_coefficient: 1, occupancy_coefficient: 1, product_coefficient: 0, constant_coefficient: 0, use_internal_space: true,
use_neighbour_space: true, aggregate: true },
        LowestLabel,
        EnforceFlowControl,
        Random],
      [ LowestLabel, EnforceFlowControl, Random ],
      [ LowestLabel, EnforceFlowControl, Random ]],
    allocator: Random,
    buffer_size: 160,
    bubble: false,
    flit_size: 16,
    intransit_priority: false,
    allow_request_busy_port: true,
    output_buffer_size: 80,
    crossbar_frequency_divisor: 1,
    crossbar_delay: 2},
  maximum_packet_size: 16,
  routing: mecanismo![
    OmniDimensionalDeroute{ allowed_deroutes: 1, include_labels: true, legend_name: "Omni-WAR" },
    Sum{
      policy: TryBoth,
      first_routing: Shortest,
      second_routing: Valiant{
        first: Shortest,
        second: Shortest,
        first_reserved_virtual_channels: [ 0 ],
        second_reserved_virtual_channels: [ 1 ]},
      first_allowed_virtual_channels: [ 0 ],
      second_allowed_virtual_channels: [ 0, 1 ],
      second_extra_label: 1,
      legend_name: "UGAL"},
    Sum{
      policy: TryBoth,
      first_routing: Shortest,
      second_routing: FMLabel{ balance_algorithm: bRINR },
      first_allowed_virtual_channels: [ 0 ],
      second_allowed_virtual_channels: [ 0 ],
      second_extra_label: 1,
      legend_name: "bRINR"},
    Sum{
      policy: TryBoth,
      first_routing: Shortest,
      second_routing: FMLabel{
        balance_algorithm: Alex{ a: 0, b: 0 },
        weight_repetition: true},
      first_allowed_virtual_channels: [ 0 ],
      second_allowed_virtual_channels: [ 0 ],
      second_extra_label: 1,
      legend_name: "sRINR"},
    SubTopologyRouting{
      logical_topology: Hamming{
        servers_per_router: 2,
        sides: [ 8, 8 ]},
      map: Identity,
      logical_routing: DOR{
        order: [ 0, 1 ]},
      opportunistic_hops: true,
      legend_name: "TERA-HX2"},
    SubTopologyRouting{
      logical_topology: Hamming{
        servers_per_router: 2,
        sides: [ 4, 4, 4 ]},
      map: Identity,
      logical_routing: DOR{
        order: [ 0, 1, 2 ]},
      opportunistic_hops: true,
      livelock_avoidance: true,
      legend_name: "TERA-HX3"},
    Valiant{
      first: Shortest,
      second: Shortest,
      first_reserved_virtual_channels: [ 0 ],
      second_reserved_virtual_channels: [ 1 ],
      legend_name: "VLB"},
    Shortest{ legend_name: "MIN" }],
  link_classes: [
    LinkClass{ delay: 2 },
    LinkClass{ delay: 2 }],
  launch_configurations: [
    Slurm{ job_pack_size: 1, time: "2-10:00:00" }]}
```