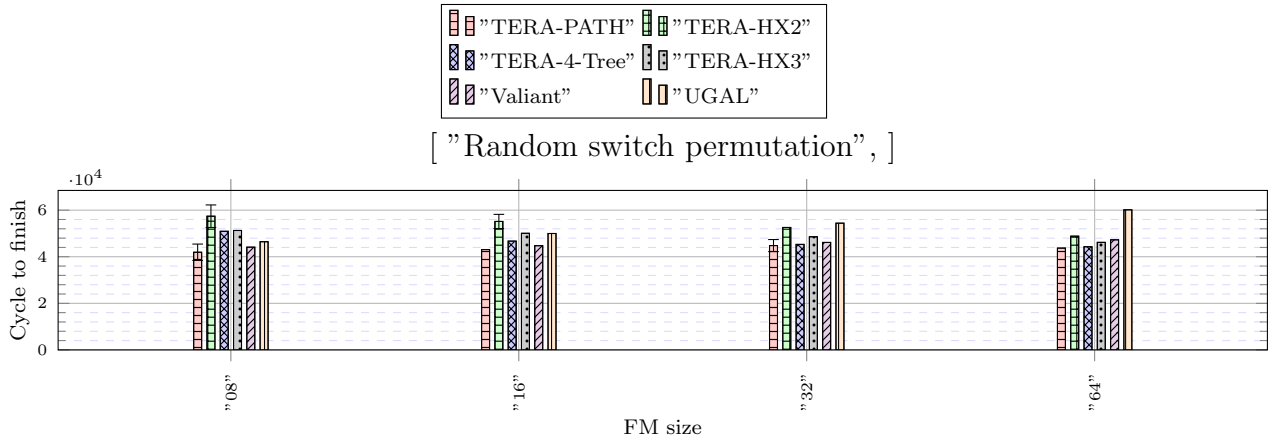

Figure 1: X: ["Random switch permutation",]

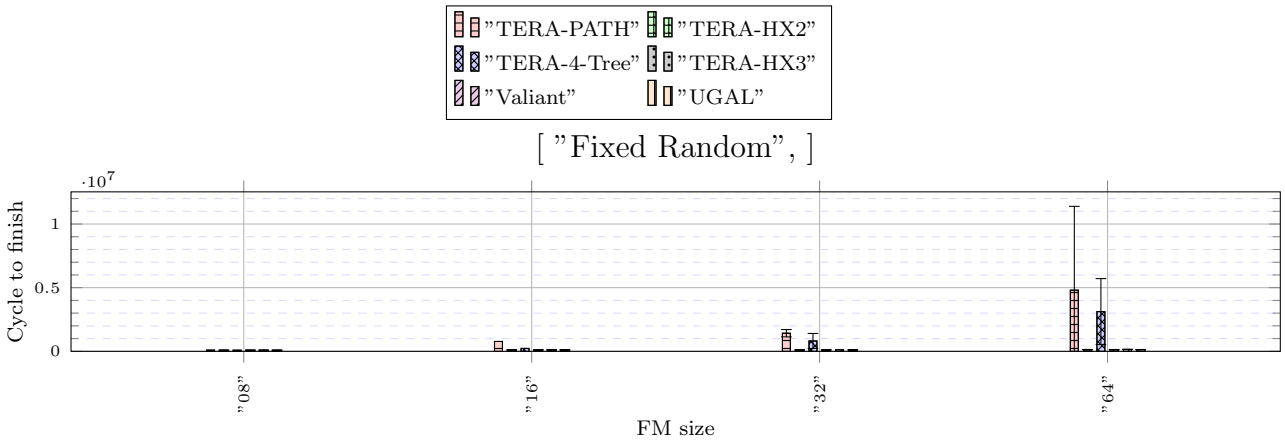

Figure 2: X: ["Fixed Random",]

The following versions used in the simulations.

- heads/alex-stable-8c73a52438392b8f3e3de5efc550d870ec3a0a1d(0.6.3)
- heads/alex-stable-release-TERA-1bbc361649a6952f6df662d4ea983c6a89d582f7(0.6.3)
- heads/alex-stable-release-TERA-bc545f9093be36b0a173d3eb574ed35e54e2c29c(0.6.3)

```
Configuration{
  random_seed: ![ 1, 2, 3 ],
  warmup: 999999999940000,
  measured: 9999999999940000,
  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: topologies![
    Hamming{
      servers_per_router: 8,
      sides: [ 8 ],
      legend_name: "08"},
    Hamming{
      servers_per_router: 16,
      sides: [ 16 ],
      legend_name: "16"},
    Hamming{
      servers_per_router: 32,
      sides: [ 32 ],
      legend_name: "32"},
    Hamming{
      servers_per_router: 64,
      sides: [ 64 ],
      legend_name: "64"}],
  traffic: Burst{
    pattern: ![
      Product{
        block_size: topologies![ 8, 16, 32, 64 ],
        global_pattern: RandomPermutation,
        block_pattern: Identity,
        legend_name: "Random switch permutation"},
      FixedRandom{ legend_name: "Fixed Random" }],
    tasks: topologies![ 64, 256, 1024, 4096 ],
    message_size: 20000,
    messages_per_task: 1},
  router: InputOutput{
    virtual_channels: mecanismo![ 1, 1, 1, 1, 2, 2 ],
    virtual_channel_policies: mecanismo![
      [
        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],
      [
        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],
      [
        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],
      [
        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],
      [
        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],
      [
        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]],
    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![
    SubTopologyRouting{
      logical_topology: Mesh{
        sides: topologies![
          [ 8 ],
          [ 16 ],
          [ 32 ],
          [ 64 ]],
        servers_per_router: 1},
      map: Identity,
      logical_routing: DOR{
        order: [ 0 ]},
      opportunistic_hops: true,
      livelock_avoidance: true,
      legend_name: "TERA-PATH"},
    SubTopologyRouting{
      logical_topology: Hamming{
        servers_per_router: 2,
        sides: topologies![
          [ 2, 4 ],
          [ 4, 4 ],
          [ 4, 8 ],
          [ 8, 8 ]]},
      map: Identity,
      logical_routing: DOR{
        order: [ 0, 1 ]},
      opportunistic_hops: true,
      legend_name: "TERA-HX2"},
    SubTopologyRouting{
      logical_topology: Tree{
        degree: 4,
        num_routers: topologies![ 8, 16, 32, 64 ],
        servers_per_router: 1},
      map: Identity,
      logical_routing: Shortest,
      opportunistic_hops: true,
      livelock_avoidance: true,
      legend_name: "TERA-4-Tree"},
    SubTopologyRouting{
      logical_topology: Hamming{
        servers_per_router: 2,
        sides: topologies![
          [ 2, 2, 2 ],
          [ 2, 2, 4 ],
          [ 2, 4, 4 ],
          [ 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: "Valiant"},
    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"}],
  link_classes: [
    LinkClass{ delay: 2 },
    LinkClass{ delay: 2 },
    LinkClass{ delay: 2 },
    LinkClass{ delay: 2 }],
  launch_configurations: [
    Slurm{
      job_pack_size: 1,
      sbatch_args: [ "--exclude=node82,node69,node70,node123" ],
      time: "2-10:00:00"}]}
```