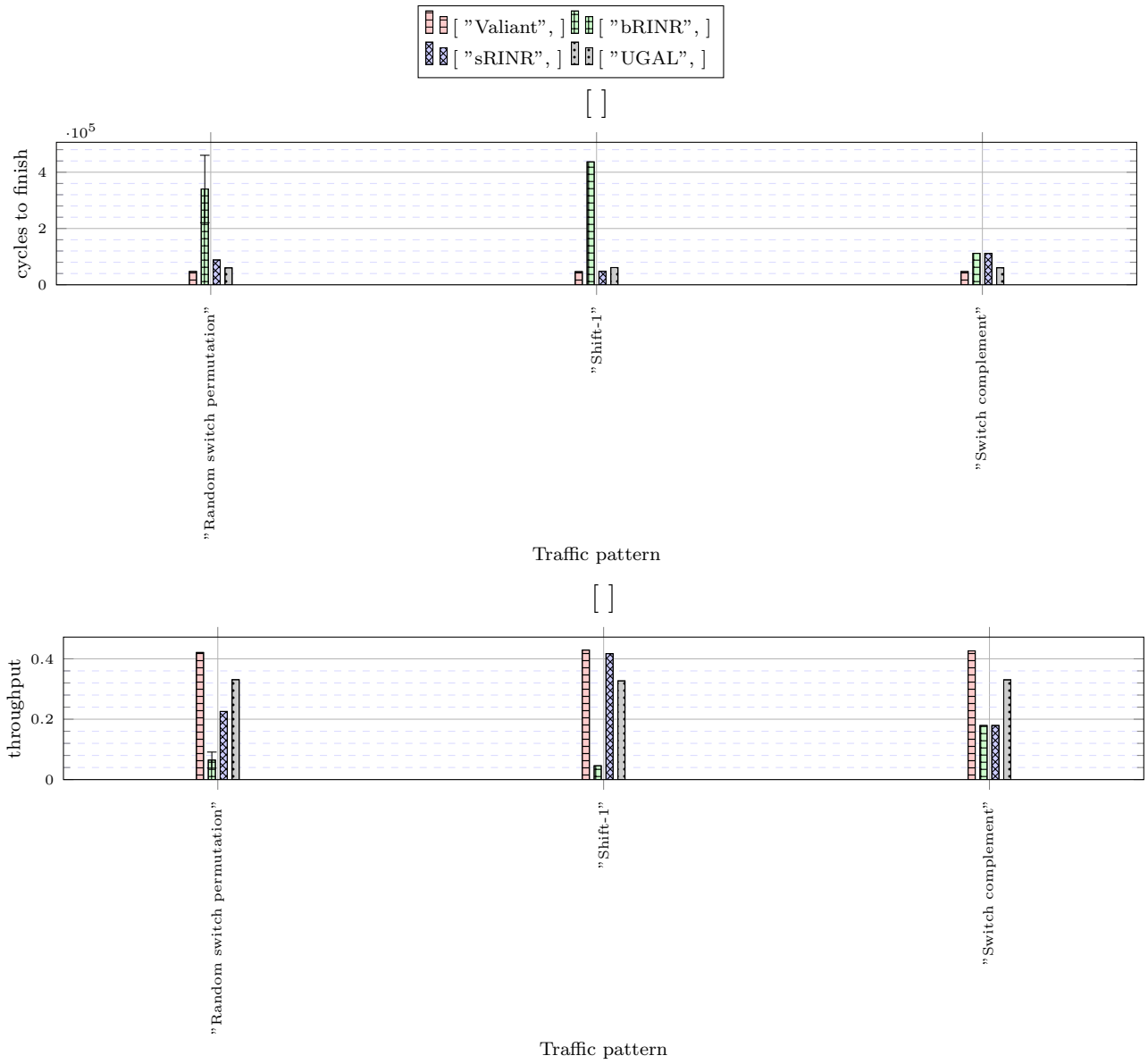
["Valiant",]
["bRINR",]
["sRINR",]
["UGAL",]
[ ]
$\cdot 10^5$
cycles to finish
0
2
4
"Random switch permutation"
"Shift-1"
"Switch complement"
Traffic pattern
[ ]
throughput
0
0.2
0.4
"Random switch permutation"
"Shift-1"
"Switch complement"
Traffic pattern

Figure 1: X: [ ]

The following versions used in the simulations.

- 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: 9990000,
  measured: 9999990000,
  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: Burst{
    pattern: ![
      CartesianTransform{
        sides: [ 64, 64 ],
        shift: [ 0, 1 ],
        legend_name: "Shift-1"},
      Composition{
        patterns: [
          LinearTransform{
            source_size: [ 64, 64 ],
            matrix: [
              [ 1, 0 ],
              [ 0, -1 ]],
            target_size: [ 64, 64 ]},
          CartesianTransform{
            sides: [ 64, 64 ],
            shift: [ 0, 63 ]}],
        legend_name: "Switch complement"},
      Product{ block_size: 64, global_pattern: RandomPermutation, block_pattern: Identity, legend_name: "Random switch permutation" }],
    tasks: 4096,
    messages_per_task: 1,
    message_size: 20000},
  router: InputOutput{
    virtual_channels: mecanismo![ 2, 1, 1, 2 ],
    virtual_channel_policies: mecanismo![
      [ 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, 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![
    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: CGLabel{ 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: CGLabel{
        balance_algorithm: sRINR{ a: 0, b: 0 },
        weight_repetition: true},
      first_allowed_virtual_channels: [ 0 ],
      second_allowed_virtual_channels: [ 0 ],
      second_extra_label: 1,
      legend_name: "sRINR"},
```

```
    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"}]}
```