

AI+X: Report 2

Hanxi Lin

September 7, 2025

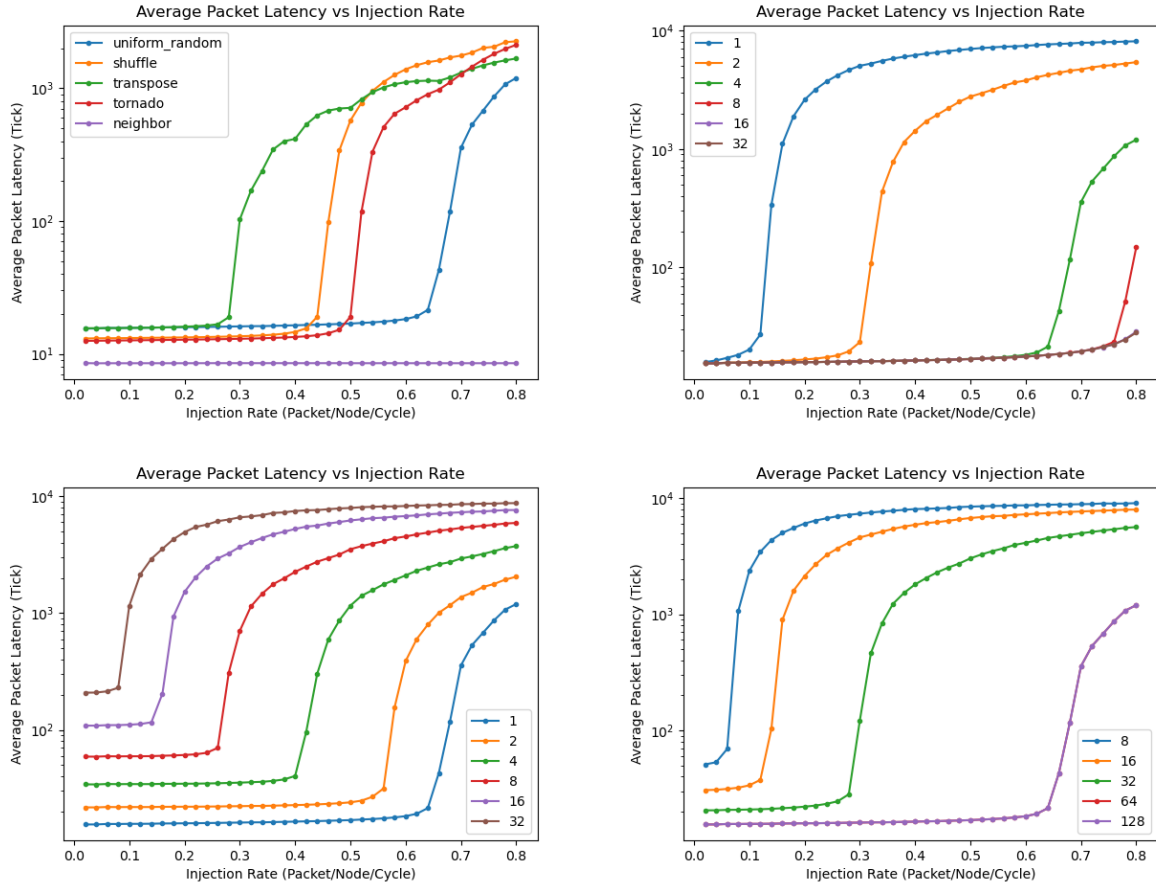


Figure 1: Average Packet Latency vs. Injection Rate under different configurations

Task 1&2

By running the shell script in appendix, we generated the four figures above. Here we generated the latency-injection rate curve instead of the latency-throughput curve, since we can still clearly observe the congestion

point, and we can better analyse the behavior after congestion.

- **SYNTHETIC TRAFFIC:** As figure 1 shows, when the injection rate is low, the average packet latency is dominated by the network latency, which is highly correlated to the average hops, where

$$H_{uniform} \approx 5.23, H_{transpose} \approx 5.27, H_{shuffle} \approx 4.01, H_{tornado} \approx 3.74, H_{neighbor} \approx 1.75$$

Whose order is identical as the order of average packet latency at low load. As the injection rate increases, the network becomes congested, and the average packet latency increases dramatically. The structure of transpose traffic makes the traffic concentrate on the diagonal and has the lowest congestion point, while uniform random traffic splits the traffic evenly and has the highest congestion point. After congestion, the average packet latency is dominated by the queuing latency, which is highly correlated to the injection rate. The order of average packet latency after congestion is identical as the order of injection rate at congestion point.

- **VCS PER VNET:** As figure 2 shows, when the injection rate is low, the average packet latency is almost identical under different VCs per VNet, since the network latency dominates. As the injection rate increases, the network becomes congested, and the average packet latency increases dramatically. More VCs per VNet can alleviate the congestion and increase the congestion point. After congestion, the average packet latency is dominated by the queuing latency, which is highly correlated to the injection rate. The order of average packet latency after congestion is identical as the order of injection rate at congestion point.
- **ROUTER LATENCY:** As figure 3 shows, when the injection rate is low, the average packet latency is dominated by the network latency, which is highly correlated to the router latency.
- **LINK WIDTH BITS:** As figure 4 shows, link width bits decides the number of cycles necessary to transmit a flit. So the network latency and congestion point is almost linearly correlated to the link width bits. When the link width bits is large enough, a flit only need one hop to transmit, the latency no longer keep decreasing.

Task 3

1. Transmission, propagation, processing, and queuing delay. In NoC we needn't consider the propagation delay since the distance is very short. The processing delay is also very small compared to the transmission and queuing delay. When the network is not congested, the transmission delay dominates the total delay. When the network is congested, the queuing delay dominates the total delay.
2. 4 flits and virtual channel flow control.

Appendix

```

1  #! /bin/bash
2
3  NUM_CPUS=64
4  SIM_CYCLES=10000
5
6  echo > network_stats.txt
7
8  for SYNTH in uniform_random shuffle transpose tornado neighbor
9  do
10     echo "SYNTHETIC TRAFFIC: $SYNTH" >> network_stats.txt
11     for INJ_RATE in 0.02 0.04 0.06 0.08 0.10 0.12 0.14 0.16 0.18 0.20 0.22 0.24 0.26
12         0.28 0.30 0.32 0.34 0.36 0.38 0.40 0.42 0.44 0.46 0.48 0.50 0.52 0.54 0.56 0.58
13         0.60 0.62 0.64 0.66 0.68 0.70 0.72 0.74 0.76 0.78 0.80
14     do
15         ./build/NULL/gem5.opt \
16         configs/example/garnet_synth_traffic.py \
17         --network=garnet --num-cpus=$NUM_CPUS --num-dirs=64 \
18         --topology=Mesh_XY --mesh-rows=8 \
19         --inj-vnet=0 --synthetic=$SYNTH \
20         --sim-cycles=$SIM_CYCLES --injectionrate=$INJ_RATE
21         INJ_TOT=$(grep -Eo "packets_injected::total\s*[0-9.]*" m5out/stats.txt |
22             grep -Eo "[0-9.]*")
23         RECV_TOT=$(grep -Eo "packets_received::total\s*[0-9.]*" m5out/stats.txt |
24             grep -Eo "[0-9.]*")
25         RECV_RATE=$(echo "scale=6;$RECV_TOT/$NUM_CPUS/$SIM_CYCLES" | bc)
26         AVG_PKT_QUEUE_LATENCY=$(grep -Eo "average_packet_queueing_latency\s*[0-9.]*"
27             m5out/stats.txt | grep -Eo "[0-9.]*")
28         AVG_PKT_NETWK_LATENCY=$(grep -Eo "average_packet_network_latency\s*[0-9.]*"
29             m5out/stats.txt | grep -Eo "[0-9.]*")
30         AVG_PKT_LATENCY=$(grep -Eo "average_packet_latency\s*[0-9.]*" m5out/stats.
31             txt | grep -Eo "[0-9.]*")
32         AVG_HOPS=$(grep -Eo "average_hops\s*[0-9.]*" m5out/stats.txt | grep -Eo "
33             [0-9.]*")
34         echo "[$INJ_RATE, $INJ_TOT, $RECV_TOT, $RECV_RATE, $AVG_PKT_QUEUE_LATENCY,
35             $AVG_PKT_NETWK_LATENCY, $AVG_PKT_LATENCY, $AVG_HOPS]" >> network_stats.
36             txt
37     done
38     echo >> network_stats.txt
39 done
40
41 python3 plot.py
42
43 echo > network_stats.txt
44
45 for VCS_PER_VNET in 1 2 4 8 16 32
46 do
47     echo "VCS PER VNET: $VCS_PER_VNET" >> network_stats.txt
48     for INJ_RATE in 0.02 0.04 0.06 0.08 0.10 0.12 0.14 0.16 0.18 0.20 0.22 0.24 0.26
49         0.28 0.30 0.32 0.34 0.36 0.38 0.40 0.42 0.44 0.46 0.48 0.50 0.52 0.54 0.56 0.58
50         0.60 0.62 0.64 0.66 0.68 0.70 0.72 0.74 0.76 0.78 0.80
51     do
52         ./build/NULL/gem5.opt \

```

```

41         configs/example/garnet_synth_traffic.py \
42         --network=garnet --num-cpus=$NUM_CPUS --num-dirs=64 \
43         --topology=Mesh_XY --mesh-rows=8 --vcs-per-vnet=$VCS_PER_VNET\
44         --inj-vnet=0 --synthetic=uniform_random \
45         --sim-cycles=$SIM_CYCLES --injectionrate=$INJ_RATE
46     INJ_TOT=$(grep -Eo "packets_injected::total\s*[0-9.]*" m5out/stats.txt |
47         grep -Eo "[0-9.]*")
48     RECV_TOT=$(grep -Eo "packets_received::total\s*[0-9.]*" m5out/stats.txt |
49         grep -Eo "[0-9.]*")
50     RECV_RATE=$(echo "scale=6;$RECV_TOT/$NUM_CPUS/$SIM_CYCLES" | bc)
51     AVG_PKT_QUEUE_LATENCY=$(grep -Eo "average_packet_queueing_latency\s*[0-9.]*"
52         m5out/stats.txt | grep -Eo "[0-9.]*")
53     AVG_PKT_NETWK_LATENCY=$(grep -Eo "average_packet_network_latency\s*[0-9.]*"
54         m5out/stats.txt | grep -Eo "[0-9.]*")
55     AVG_PKT_LATENCY=$(grep -Eo "average_packet_latency\s*[0-9.]*" m5out/stats.
56         txt | grep -Eo "[0-9.]*")
57     AVG_HOPS=$(grep -Eo "average_hops\s*[0-9.]*" m5out/stats.txt | grep -Eo "
58         [0-9.]*")
59     echo "[$INJ_RATE, $INJ_TOT, $RECV_TOT, $RECV_RATE, $AVG_PKT_QUEUE_LATENCY,
60         $AVG_PKT_NETWK_LATENCY, $AVG_PKT_LATENCY, $AVG_HOPS]" >> network_stats.
61         txt
62 done
63 echo >> network_stats.txt
64 done
65
66 python3 plot.py
67
68 echo > network_stats.txt
69
70 for ROUTER_LATENCY in 1 2 4 8 16 32
71 do
72     echo "ROUTER_LATENCY: $ROUTER_LATENCY" >> network_stats.txt
73     for INJ_RATE in 0.02 0.04 0.06 0.08 0.10 0.12 0.14 0.16 0.18 0.20 0.22 0.24 0.26
74         0.28 0.30 0.32 0.34 0.36 0.38 0.40 0.42 0.44 0.46 0.48 0.50 0.52 0.54 0.56 0.58
75         0.60 0.62 0.64 0.66 0.68 0.70 0.72 0.74 0.76 0.78 0.80
76     do
77         ./build/NULL/gem5.opt \
78         configs/example/garnet_synth_traffic.py \
79         --network=garnet --num-cpus=$NUM_CPUS --num-dirs=64 \
80         --topology=Mesh_XY --mesh-rows=8 --router-latency=$ROUTER_LATENCY\
81         --inj-vnet=0 --synthetic=uniform_random \
82         --sim-cycles=$SIM_CYCLES --injectionrate=$INJ_RATE
83     INJ_TOT=$(grep -Eo "packets_injected::total\s*[0-9.]*" m5out/stats.txt |
84         grep -Eo "[0-9.]*")
85     RECV_TOT=$(grep -Eo "packets_received::total\s*[0-9.]*" m5out/stats.txt |
86         grep -Eo "[0-9.]*")
87     RECV_RATE=$(echo "scale=6;$RECV_TOT/$NUM_CPUS/$SIM_CYCLES" | bc)
88     AVG_PKT_QUEUE_LATENCY=$(grep -Eo "average_packet_queueing_latency\s*[0-9.]*"
89         m5out/stats.txt | grep -Eo "[0-9.]*")
90     AVG_PKT_NETWK_LATENCY=$(grep -Eo "average_packet_network_latency\s*[0-9.]*"
91         m5out/stats.txt | grep -Eo "[0-9.]*")
92     AVG_PKT_LATENCY=$(grep -Eo "average_packet_latency\s*[0-9.]*" m5out/stats.

```

```

79         txt | grep -Eo "[0-9.]*")
      AVG_HOPS=$(grep -Eo "average_hops\s*[0-9.]*" m5out/stats.txt | grep -Eo "[0-9.]*")
80      echo "[$INJ_RATE, $INJ_TOT, $RECV_TOT, $RECV_RATE, $AVG_PKT_QUEUE_LATENCY,
          $AVG_PKT_NETWK_LATENCY, $AVG_PKT_LATENCY, $AVG_HOPS]" >> network_stats.
          txt
81      done
82      echo >> network_stats.txt
83  done
84
85  python3 plot.py
86
87  echo > network_stats.txt
88
89  for LINK_WIDTH_BITS in 8 16 32 64 128
90  do
91      echo "LINK WIDTH BITS: $LINK_WIDTH_BITS" >> network_stats.txt
92      for INJ_RATE in 0.02 0.04 0.06 0.08 0.10 0.12 0.14 0.16 0.18 0.20 0.22 0.24 0.26
          0.28 0.30 0.32 0.34 0.36 0.38 0.40 0.42 0.44 0.46 0.48 0.50 0.52 0.54 0.56 0.58
          0.60 0.62 0.64 0.66 0.68 0.70 0.72 0.74 0.76 0.78 0.80
93      do
94          ./build/NULL/gem5.opt \
95          configs/example/garnet_synth_traffic.py \
96          --network=garnet --num-cpus=$NUM_CPUS --num-dirs=64 \
97          --topology=Mesh_XY --mesh-rows=8 --link-width-bits=$LINK_WIDTH_BITS\
98          --inj-vnet=0 --synthetic=uniform_random \
99          --sim-cycles=$SIM_CYCLES --injectionrate=$INJ_RATE
100         INJ_TOT=$(grep -Eo "packets_injected::total\s*[0-9.]*" m5out/stats.txt |
            grep -Eo "[0-9.]*")
101         RECV_TOT=$(grep -Eo "packets_received::total\s*[0-9.]*" m5out/stats.txt |
            grep -Eo "[0-9.]*")
102         RECV_RATE=$(echo "scale=6;$RECV_TOT/$NUM_CPUS/$SIM_CYCLES" | bc)
103         AVG_PKT_QUEUE_LATENCY=$(grep -Eo "average_packet_queueing_latency\s*[0-9.]*"
            m5out/stats.txt | grep -Eo "[0-9.]*")
104         AVG_PKT_NETWK_LATENCY=$(grep -Eo "average_packet_network_latency\s*[0-9.]*"
            m5out/stats.txt | grep -Eo "[0-9.]*")
105         AVG_PKT_LATENCY=$(grep -Eo "average_packet_latency\s*[0-9.]*" m5out/stats.
            txt | grep -Eo "[0-9.]*")
106         AVG_HOPS=$(grep -Eo "average_hops\s*[0-9.]*" m5out/stats.txt | grep -Eo "[0-9.]*")
107         echo "[$INJ_RATE, $INJ_TOT, $RECV_TOT, $RECV_RATE, $AVG_PKT_QUEUE_LATENCY,
            $AVG_PKT_NETWK_LATENCY, $AVG_PKT_LATENCY, $AVG_HOPS]" >> network_stats.
            txt
108     done
109     echo >> network_stats.txt
110 done
111
112 python3 plot.py

```