Mini-Project: Network Simulation and Evaluation

Ruolan Wang
Yuqi Yi
Ziyue Feng

May 20, 2024

1 Measurement Setup

The problem in this setup is the packet timing started at the output of the source queue. The traffic pattern applied is not the intended pattern and the latency computed does not account for time waiting to enter the network. Thus the packet latency is understated. To correct it, the "Input count & timing" should be placed between the packet source and the source queue.

2 Performance Evaluation

2.1 Explain the differences of open-loop and closed-loop measurements

The open-loop measurement configuration enables the traffic parameters to be controlled independently of the network itself. The goal is generally to evaluate the network on a specific traffic pattern for throughput, latency, and fault tolerance.

Closed-loop measurement systems, in which the network influences the traffic, are useful for measuring overall system performance. A more typical application of this simulation setup would be to test the sensitivity of the application run time to network parameters such as bandwidths, routing algorithms, and flow control.

2.2 Explain why, in the open-loop measurement, during the drain phase, we need to run the network long enough for all of the measurement packets to reach their destination?

As these packets arrive at the destination, their finish time is measured and they are logged. Latency measures are computed from the start and finish times of all measurement packets. It is important that the simulation be run long enough to measure the finish time of every measurement packet to capture the tail of the latency distribution.

2.3 Explain the throughput bounds in a general network as shown in the latency vs. offered traffic graph, specifically, why the Topology bound is larger than the routing bound, which is larger than the saturation throughput

Topology throughput bound is ideal, assuming random traffic with perfect flow control (no idle cycles for channels) and perfect routing(load balance, minimal path). So the topology bound is the largest.

Routing throughput bound counts possible load imbalance of a routing algorithm, so it is smaller than the topology bound.

The Saturation throughput counts idle channels due to resource dependencies (VC allocation dependency for deadlock freedom, credit availability, etc.) So it is the smallest and most accurate.

3 Simulator and Simulation

3.1 The experimental setup

3.1.1 Explain how iSLIP algorithm works

iSLIP is a separable allocation method that uses round-robin arbiters and updates the priority of each arbiter only when that arbiter generates a winning grant. iSLIP can be used either in a single pass, or as an iterative matching algorithm. By rotating the winning arbiters, iSLIP acts to stagger the priority of the input arbiters, resulting in fewer conflicts at the output stage. The update of a priority only occurs when an arbitration results in a grant and, as in a round-robin arbiter, priorities are updated so that a winning request has the lowest priority in the next round.

3.1.2 Explain the routing algorithms, namely, DOR, ROMM, VAL and MAD

DOR (Dimension-Order Routing) is a deterministic routing algorithm. Digits of destination addresses are used one at a time to direct the routing. The packet is routed along these directions { +x / -x, +y / -y } until it reaches its destination.

ROMM (Randomized Oblivious Minimal algorithm) enhances load balancing by introducing randomness in routing. It evaluates all possible paths with minimal hop count and randomly selects one as the route for the packet.

VAL (Valiant's randomized algorithm) is non-minimal and designed to balance network load and avoid congestion. Instead of sending a packet directly to its destination, VAL routes packers through a randomly chosen intermediate node before sending them to their final destination.

MAD (minimal-adaptive routing algorithm) chooses among the minimal routes from source s to destination d. At each hop, a routing function generates a productive output vector that identifies which output channels of the current node will move the packet closer to its destination. The network state (current network traffic and congestion condition) is then used to select one of these channels for the next hop.

3.2 Section 25.1, Routing experiments

This part of experiments is to investigate the impact of routing from both latency and throughput perspectives, taking 8-ary 2-mesh as example. Four routing algorithms: dimension-order routing(DOR), the randomized minimal algorithm(ROMM), minimal-adaptive routing algorithm(MAD) and Valiant's randomized algorithm(VAL) are compared in Booksim simulations.

Unless otherwise stated, the offered traffic is calculated by multiplication of injection rate (packets per cycle) and packet size in flits, divided by ideal throughput $\Theta = 4b/k$, where the b is 1 flit per cycle. All packets are 20 flits.

3.2.1 Validate the latency performance

In Figure 1, the network with DOR achieves above 90% of capacity, and when it with ROMM and MAD the performance reaches 75%. In the perspective of ideal capacity, since 100% of capacity is ideal for DOR, ROMM and MAD, they reach 90%, 75% and 75% of the ideal respectively. For VAL, 50% of capacity is the maximum, so it also saturates around 90% of its ideal.

In next step, the traffic pattern is changed to transpose. Different with uniform traffic pattern, DOR, ROMM, MAD, VAL algorithms reach 35%, 60%, 75%, 45% of their ideal respectively(Figure 2).

3.2.2 Validate the throughput performance

Throughput, which is the accepted traffic, is also related to the offered traffic. We simulate the throughput performance in the same topology with previous experiments. The simulation results are shown in Figure 3. We can see that throughput increases with the demand before it reaches the saturation, however, degrades after that. It may be because the network is not stable.

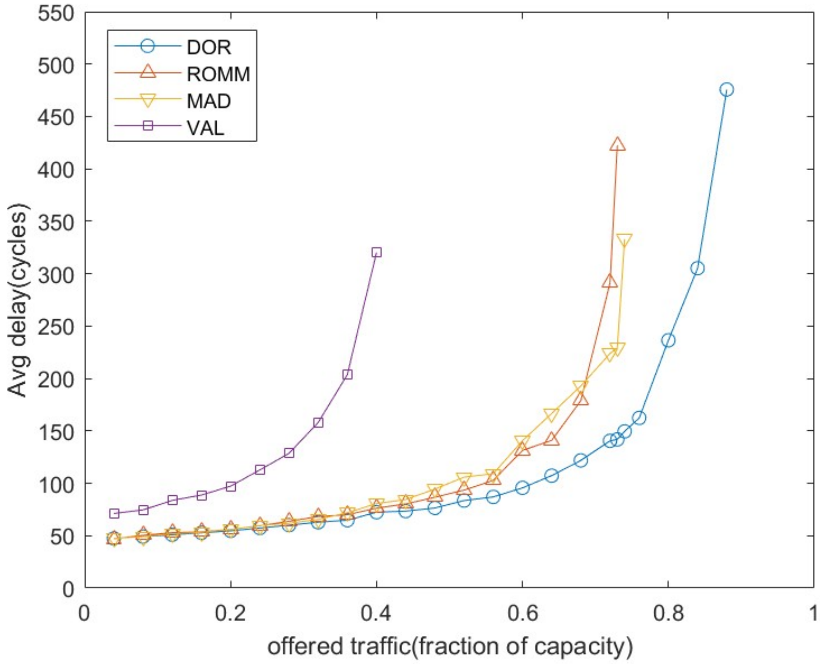

Figure 1: Performance of several routing algorithms on an 8-ary 2-mesh under uniform traffic(Reproduction of Figure 25.1).

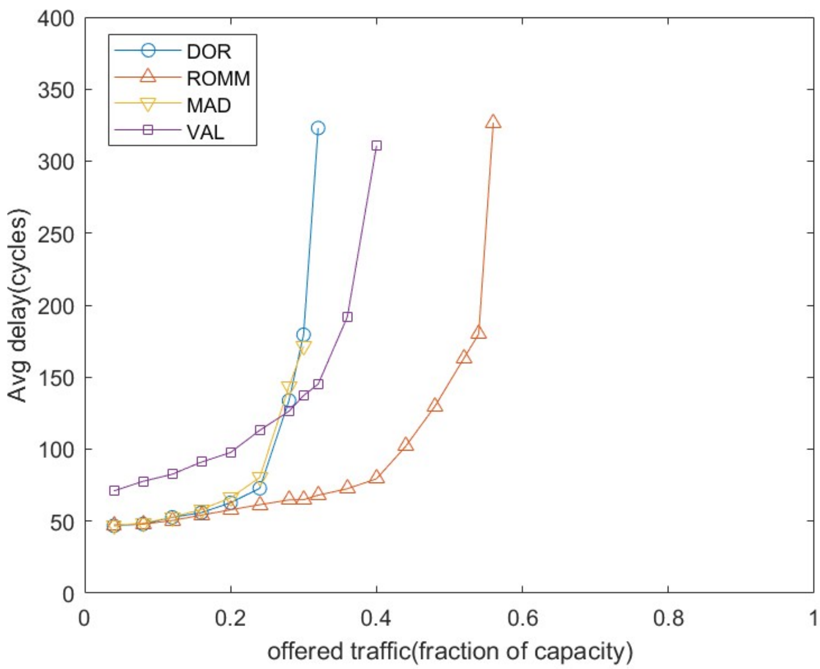

Figure 2: Performance of several routing algorithms on an 8-ary 2-mesh under transpose traffic(Reproduction of Figure 25.2).

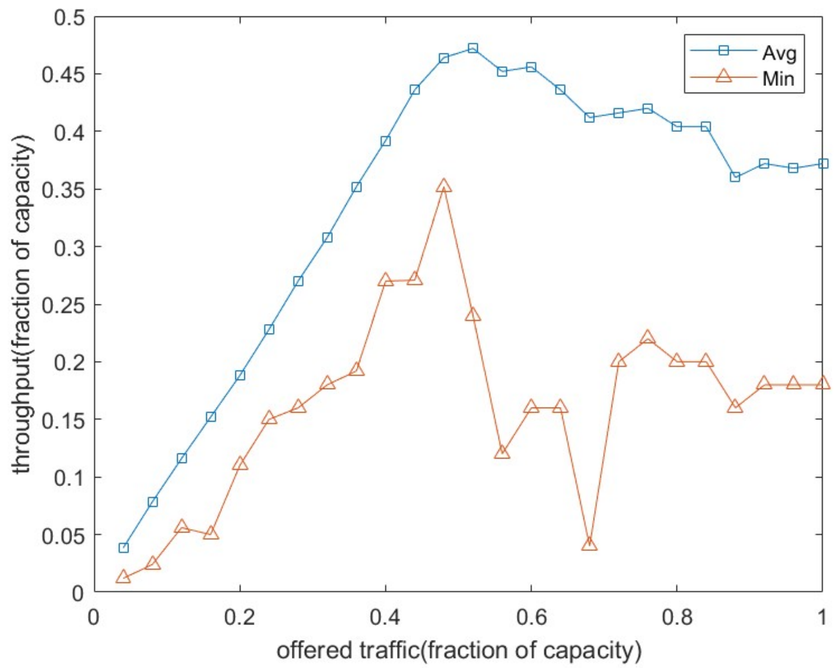

Figure 3: Throughput vs. offered traffic for an 8-ary 2-mesh under bit-complement traffic using dimension-order routing.(Reproduction of Figure 23.2).

3.3 Section 25.2, Flow control experiments

3.3.1 Understand the impact of virtual channel partitionings on network performance by re-producing Figure 25.7

In figure 4, we obtain a similar result to Figure 25.7 in the book. The total amount of buffering (the product of the number of virtual channels times the individual virtual channel depth) is held constant across each of the configurations. So we can see the impact of different virtual-channel partitionings here. Under the same offered traffic, the network with 16 virtual channels and 2 buffers always has the largest latency while the network with 4 VCs and 8 buffers has the smallest. Increasing the number of virtual channels tends to increase latency below saturation.

3.3.2 Understand the impact of network size on network performance by re-producing Figure 25.8

In figure 5, we obtain a similar result to Figure 25.8 in the book. The size of a network can have a significant effect on the fraction of its ideal throughput it can achieve. The achieved capacity seems to be a function of the radix of the network. The radix-4 networks both begin to saturate at approximately 65%, while the radix-8 and radix-16 networks saturate near 80% and 83%, respectively.

3.3.3 Understand the impact of injection processes on network performance by re-producing Figure 25.9

In figure 6, we obtain a similar result to Figure 25.9 in the book. For the torus topology, ideal throughput $\Theta = 8b/k$. Here we can see the impact of packet size, i.e. the burstiness of the underlying traffic, on the average delay. The dominant trend in the figure is both the increasing latency and decreasing throughput that comes from larger packet size. The outlier from the overall trend is the case in which the packet size is one (PS = 1), which is the result of the conservative approach for reallocating virtual channels.

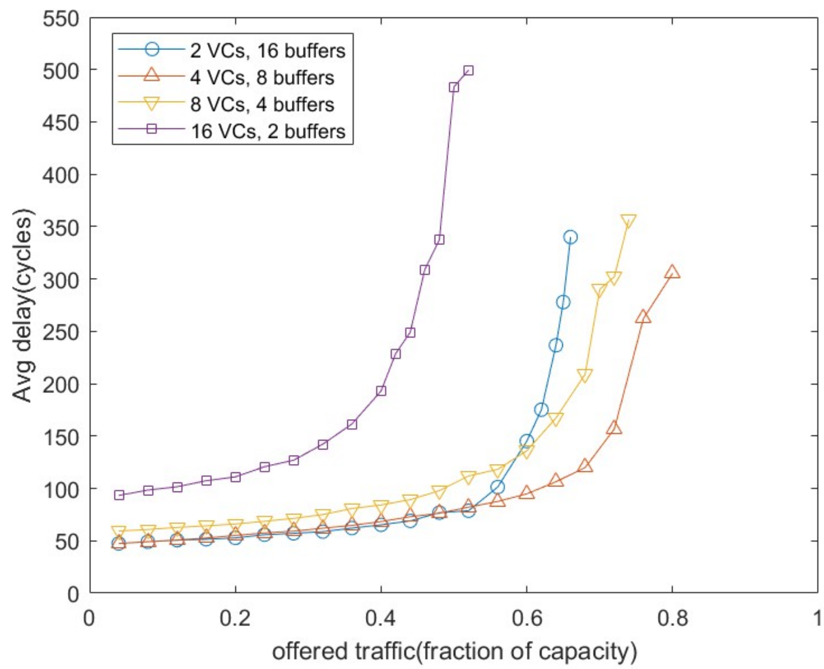

Figure 4: Latency vs. offered traffic for an 8-ary 2-mesh with various virtual channel partitionings under uniform traffic(Reproduction of Figure 25.7).

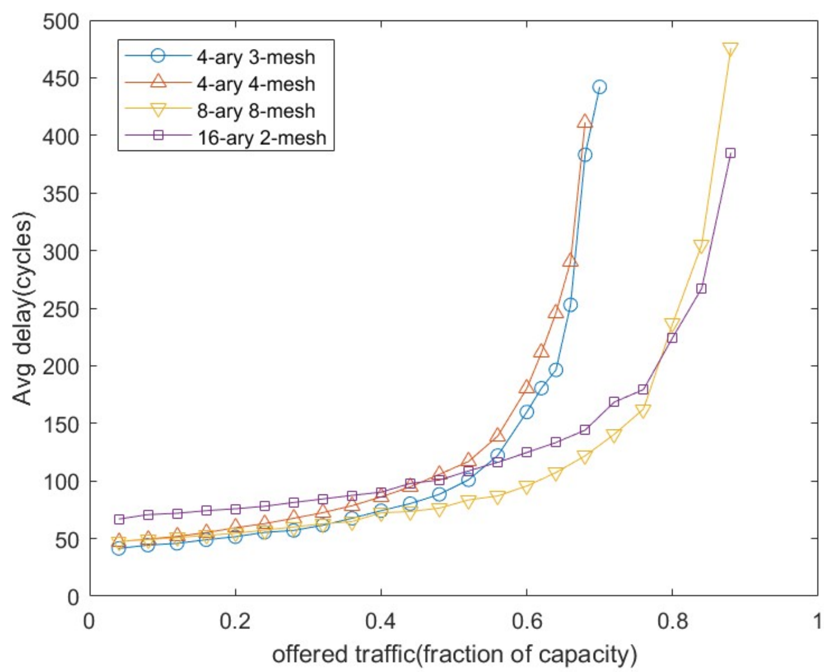

Figure 5: Latency vs. offered traffic for several mesh networks under uniform traffic with dimension-ordered routing. The injection process is Bernoulli(Reproduction of Figure 25.8).

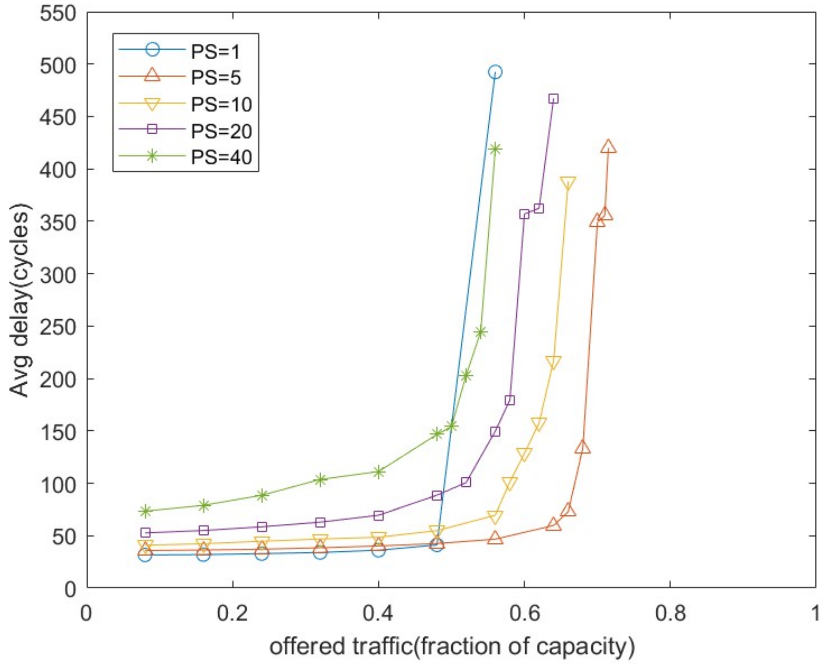

Figure 6: Latency vs. offered traffic in an 8-ary 2-cube under uniform traffic with dimension-ordered routing using different packet sizes(Reproduction of Figure 25.9).

4 Simulation with BookSim

4.1 simulate a 6x6 mesh network, a 6x6 torus network. Draw average latency vs. offered traffic graphs, one with queuing latency, and the other without queuing latency.

It can be seen from the figure 7 and figure 8 that when offered traffic latency increases, the average delay also increases. For the network that count in the queuing latency, the queues on the router will get congested when the traffic on the network increases, so the average latency increases rapidly after the offered traffic reaches a certain level. However for the network without queuing latency, the packets can the delivered immediately without waiting in a queue, so the average latency only increase a little bit with the offered traffic.

Regarding that the ideal throughput of a torus is 2 times that of a mesh, it takes half of the offered traffic to reach the same latency for a torus.

4.2 Draw per-node throughput as bar diagram (one bar for one node) for the 6x6 mesh and 6x6 torus networks in A).

The two sets of experiment are conducted under the same offered traffic. As it shows in figure 10, the per-node throughput in a torus remains basically the same because all the points in the same column or a same row are connected by a ring, so there's no difference between the nodes. However for a mesh, the throughput of the nodes on two edges of a row are significantly higher. The edge points can only deliver packets in one direction(+x or -x) instead of two(+x and -x) and thus the throughput are higher .

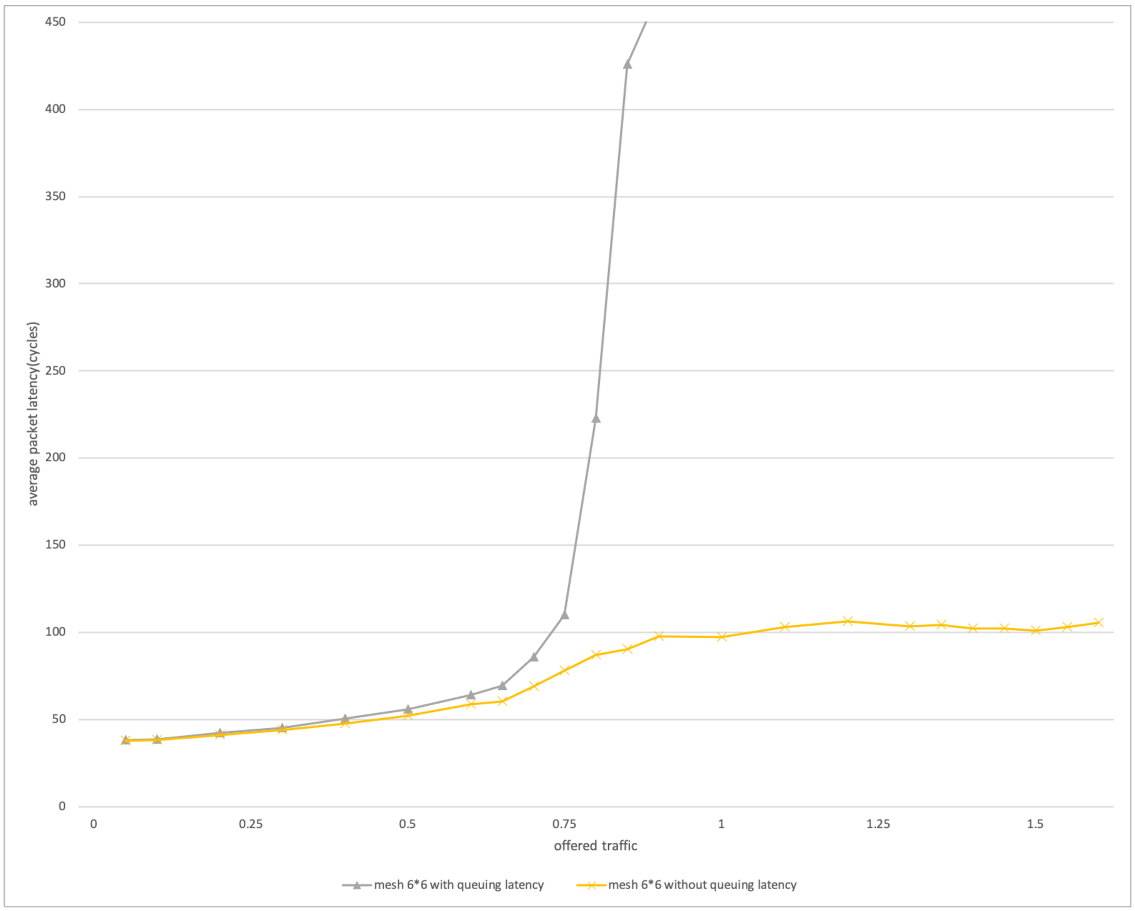

Figure 7: Latency vs. offered traffic in an 6*6 mesh network under uniform traffic with dimension-ordered routing with or without latency.

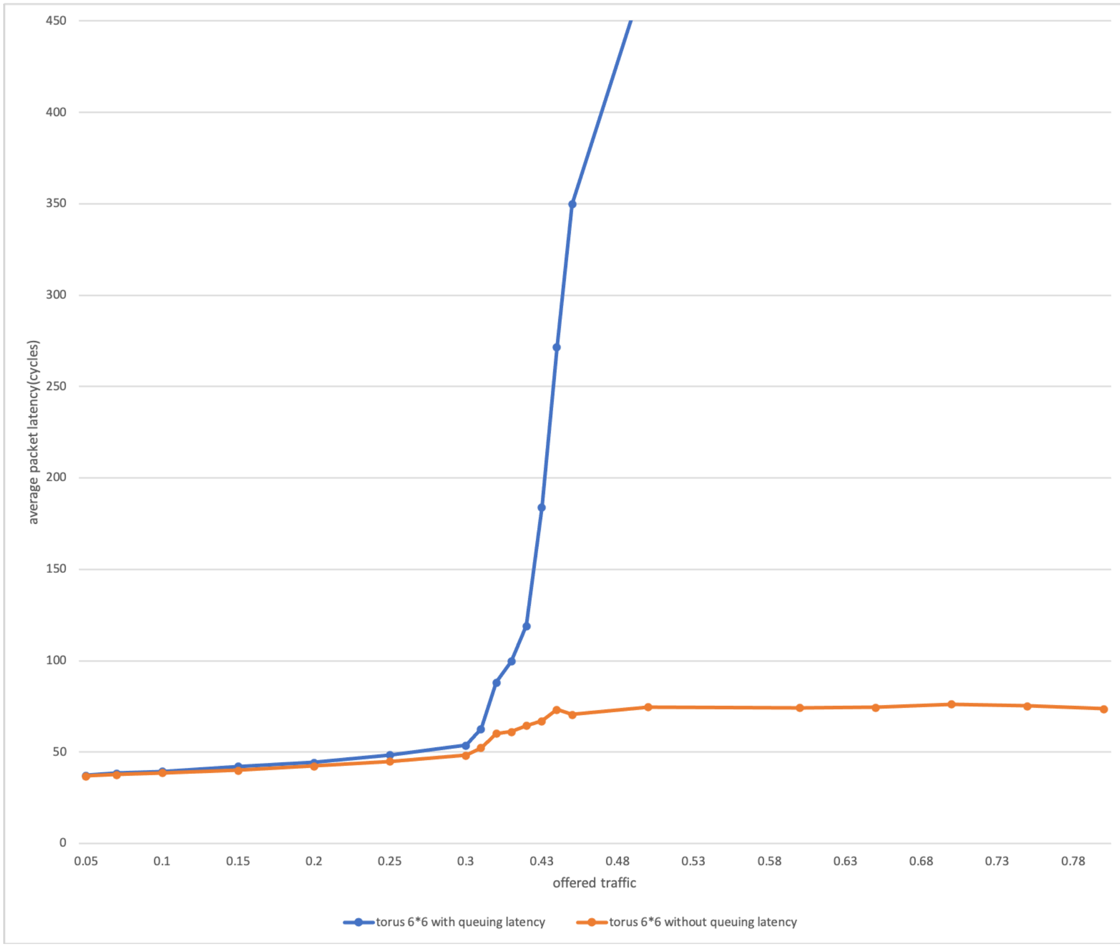

Figure 8: Latency vs. offered traffic in an 6*6 torus network under uniform traffic with dimension-ordered routing with or without latency.

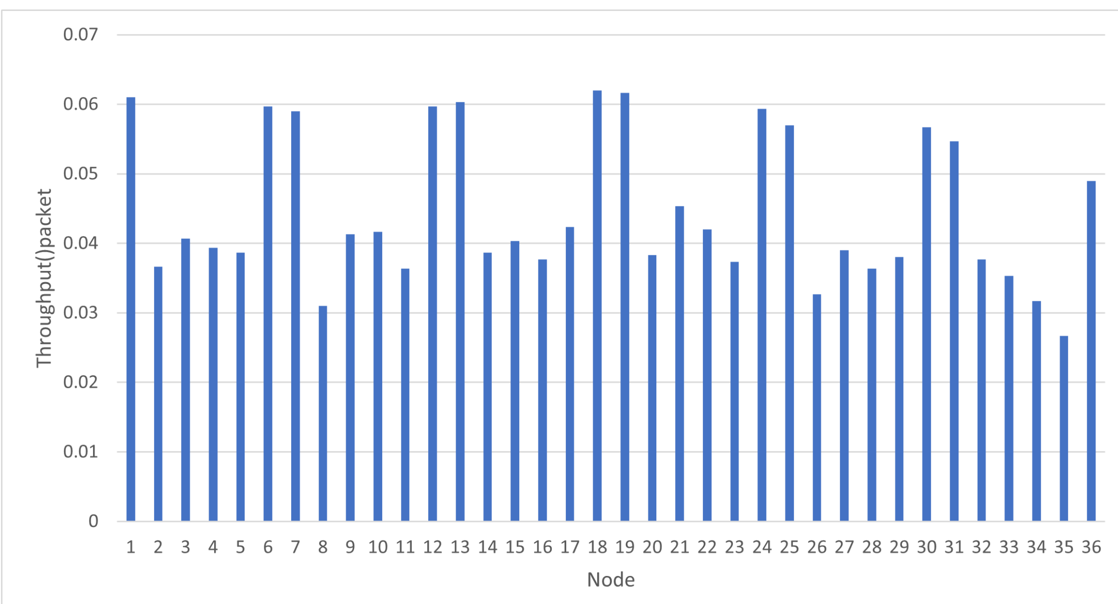

Figure 9: Per-node throughput for the 6*6 mesh.

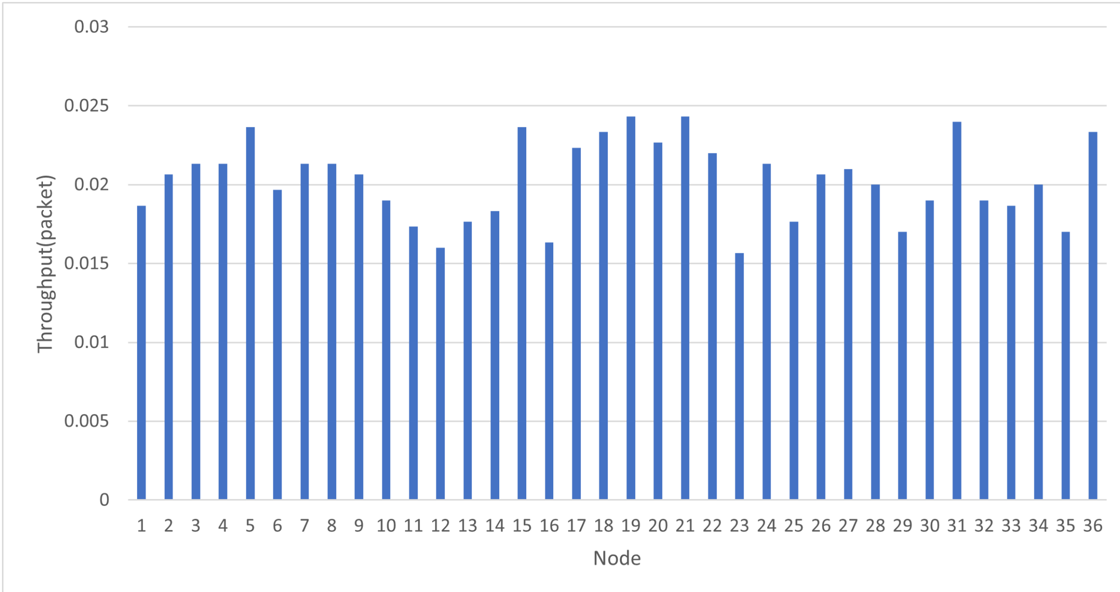

Figure 10: Per-node throughput for the 6*6 torus.

4.3 Simulate mesh and torus networks with the following sizes: 2x2, 4x4, 6x6, 8x8, 10x10. Design your own figures to illustrate the performance differences (average latency, maximum latency, saturation throughput) of the two types of networks. Which network is more scalable? Why?

Torus are more scalable. In smaller size of network, the performance are basically the same for mesh and torus. When the network size gets bigger, the latency of mesh increases more rapidly than a torus and the throughput is decreases steeper than a torus. This may come from that when the size gets bigger, it becomes more difficult for the edge nodes of a mesh to deliver a packet. However, all the points are on a ring for a torus, there's no such edge point dilemma.

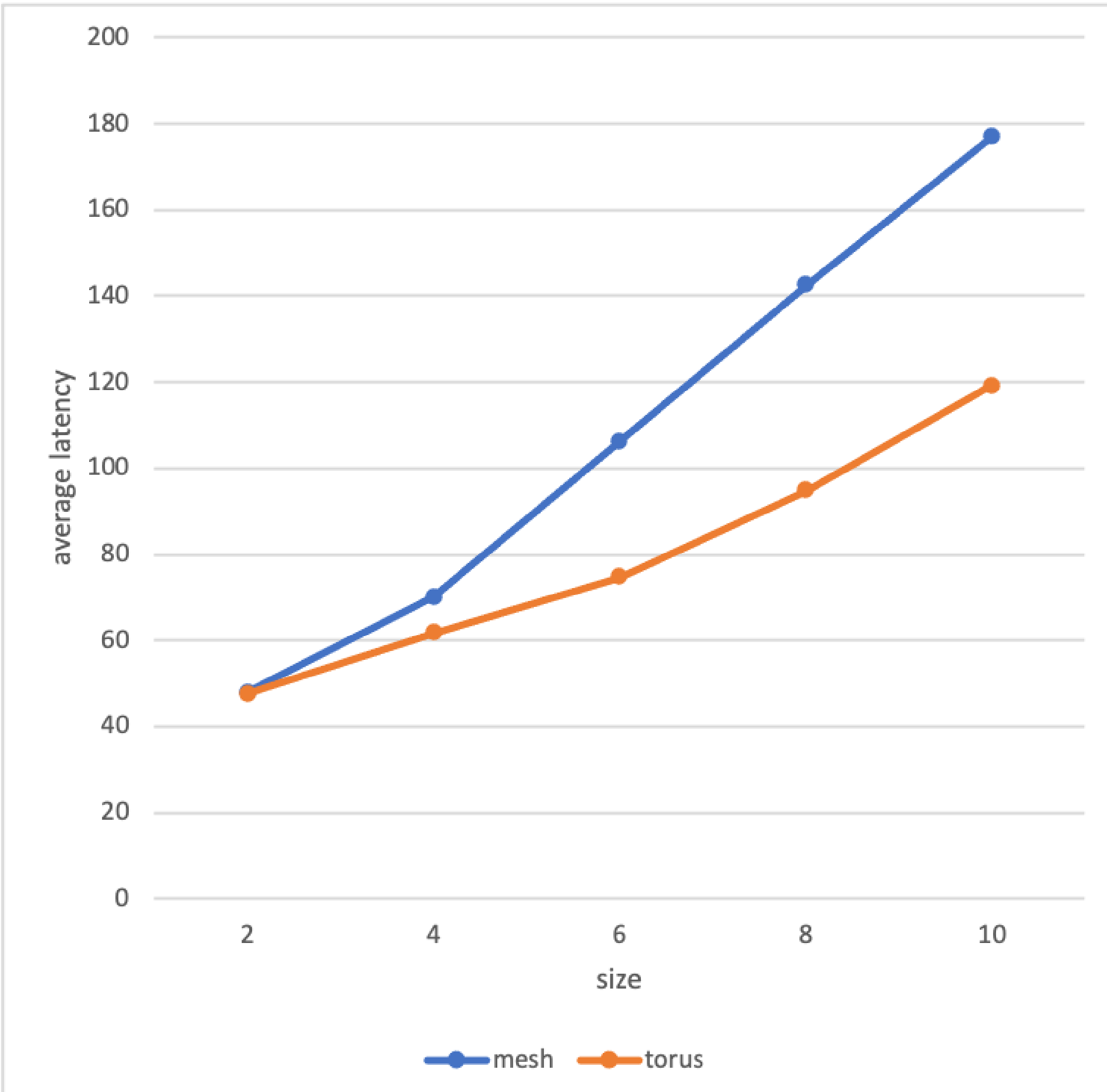

Figure 11: Average latency for mesh and torus in different sizes.

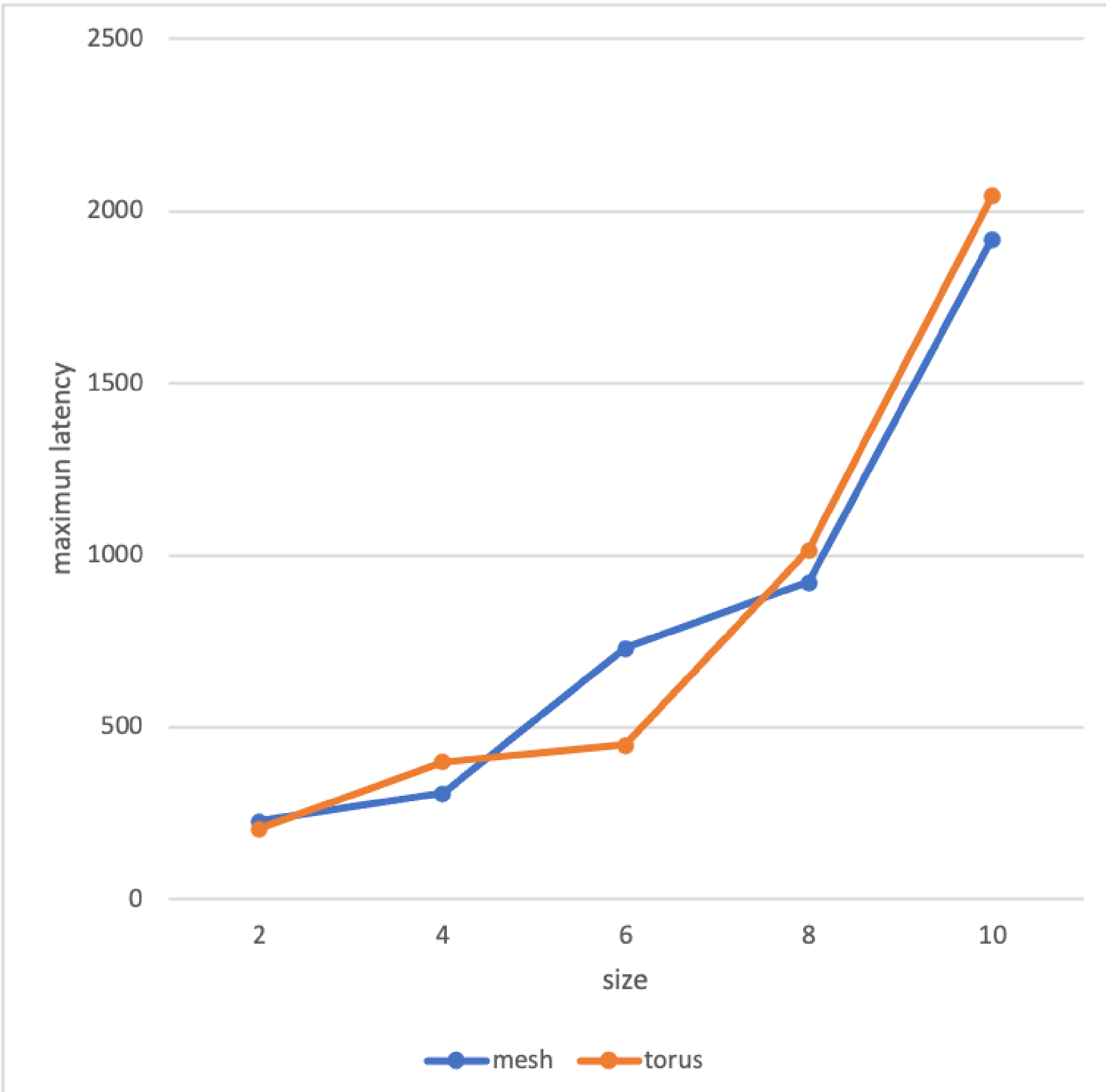

Figure 12: Maximum latency for mesh and torus in different sizes.

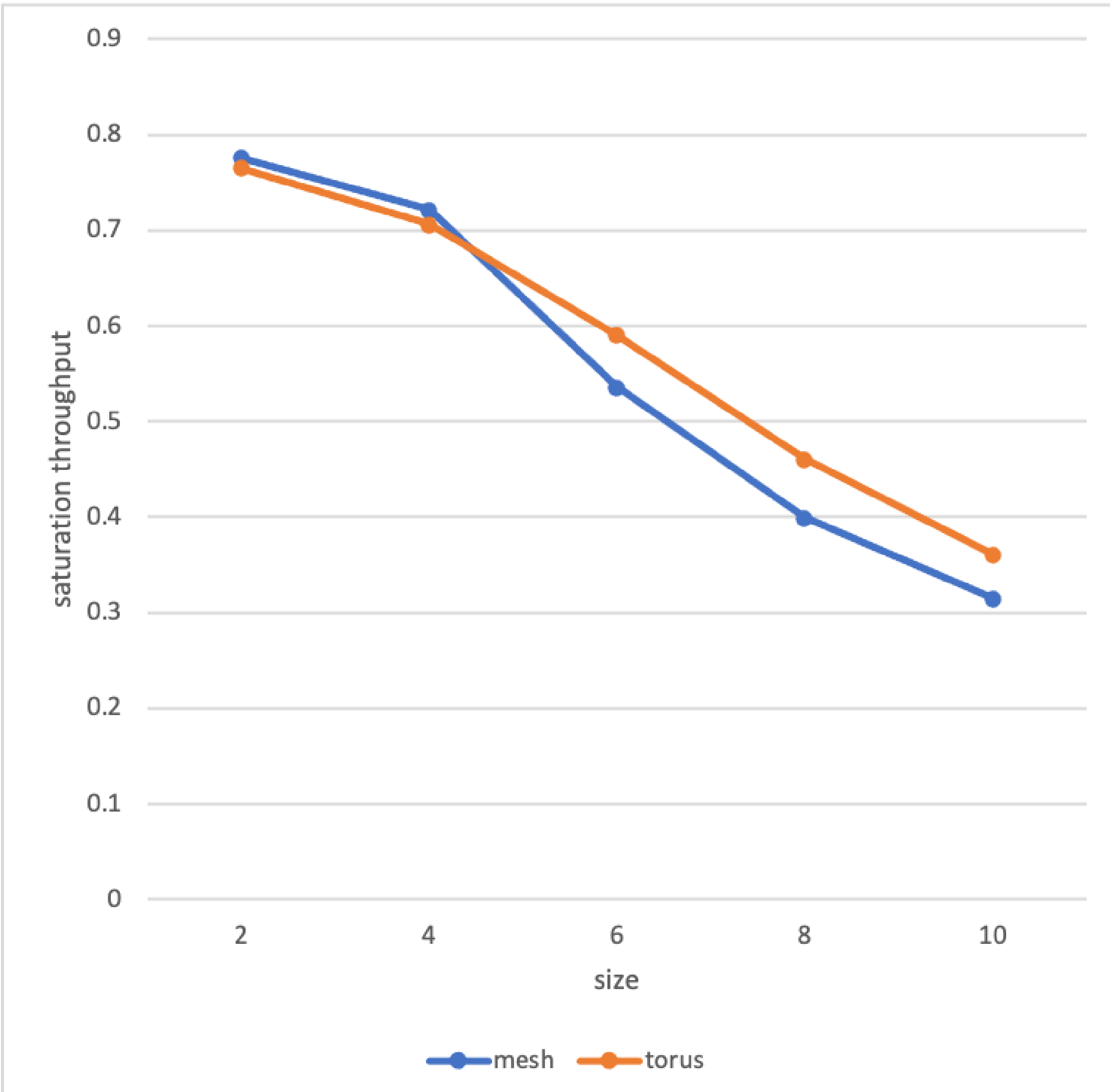

Figure 13: Saturation throughput for mesh and torus in different sizes.