**Booksim 2.0**

**Why Booksim?**

BookSim network simulator is a detailed, cycle-accurate simulator for Network-on-Chips (NoCs) that can also be used to model interconnection networks for a variety of designs.

The simulator is designed for use along with the book written by William J. Dally, which makes it optimum for our use. Furthermore, it supports Torus topology, Virtual Channels, flexibility in choosing arbitration and allocation methods. The simulator makes it possible to simulate larger topologies which often takes a lot of time in RTL simulations and thus offer a wide range of experimentation with parameters.

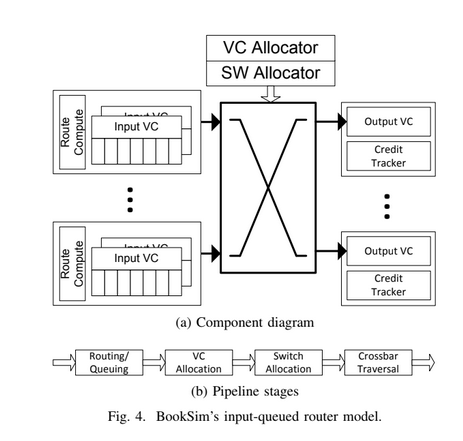
**Major differences between the Software simulator and RTL implementation:**

* The simulator uses credit-based back pressure, while the RTL implementation for REDEFINE NoC uses handshaking signals for back pressure. We came to the conclusion that as long as the latency in terms of clock cycles (for credits) is the same, it shouldn’t make a difference in the final results in terms of throughput and latency. The difference arises when implementing on hardware. For example, the credit based approach will give a smaller combinational delay but higher power consumption. The handshaking signals approach will give a higher combinational delay but a less power consumption. This wouldn't be reflected in the booksim simulation. The difference will only arise when comparing the power consumption and the critical path delay.
* There are two stages of Arbitration(two are locking type and one is Round-Robin) status of which is provided to the upstream routers, which makes single cycle router implementation possible in REDEFINE NoC)(same as Look Ahead Carry stages in adder)
* The aim is to get a single cycle delay for the router. So that, Both the implementations have a 1 cycle latency from inFlit to OutFlit.

In Single cycle router implementation, flit enters the router in the nth cycle and produces at the output at n+1th cycle.

* *Comparison of Latency :* In the RTL implementation it takes 6 cycles from packet generating in nth router to (n+1)th router. This includes three components: PacketToFlit and FlitToPacket delay, single cycle router delay(for 2 adjacent routers, 2 cycles), and Network Interface to router delay. While in the booksim simulation takes no such delays in such conversions. We can easily get rid of these extra cycles by simply subtracting 4 from the average flit latency in the booksim simulation. This makes the comparison between booksim and RTL simulation apt.

**Microarchitecture of the input-queued router:**



1. Pipeline

We can vary the delay associated with each of the pipeline stages by setting the various parameters provided by the simulator(such as routing delay, vc\_alloc\_delay, sw\_alloc\_delay and st\_final\_delay )

We can implement look ahead routing by setting the routing\_delay to 0. However, this may not give accurate results for complex routing methods such as adaptive routing. But setting this parameter to 0 will work fine for the RTL implementation, because a simple dimension-order routing is implemented.

The VC allocation and the switch allocation is performed independently and parallelly by setting speculative to 1.In that case max of vc\_allocation\_delay and sw\_allocation\_delay is considered. Else, without speculation sum of two stages is considered in latency

The switch traversal latency can be eliminated by setting st\_delay to 0. Performing above-mentioned optimization steps, we can get a single cycle implementation of the IQ router.

1. Allocators

We will be using the iSLIP allocator, which sets the priority of the last allotted request to the least priority. (Round-robin arbitration)

1. Buffer Management

We use static buffer management (a single buffer cannot be shared between 2 VC’s). We do this by setting buffer\_policy to “private”.

1. Other flow control parameters

a. Setting wait\_for\_tail\_credit to 1 allows for atomic VC allocation. This provides us with the functionality of lanes( we did this to be on safer side, though it increase the latency)

b. Setting vc\_busy\_when\_full as 1 marks VCs in use when they have no credit available. Setting this to 0, may drop flits when downstream VC is full

c. Setting hold\_switch\_for\_packet to 1 will perform the locking mechanism RTL implementation of REDEFINE NoC

**Simulation**

* Injection rate can be set between 0 and 1(0 corresponds to 0% and 1 corresponds to 100% packet injection rate).
* To check the validity of the simulation, we kept a very low injection rate (10^-5), classes = 1 and injection\_process as “on\_off”. The reasoning behind the above configuration is that, this would describe the latency of 1 flit traversing independently through the network. We got a latency of 4, which is what we expect from RTL simulation without the packet to flit and flit to packet conversion.
* For running the simulation, we had to remove assertion in switch\_update stage(GetSimTime()==time. ), Otherwise setting the switch traversal to 0 was aborting the simulation

The following parameters have been modified to understand different cases using the simulation:

1.No. of classes ( Parameter : classes)

2.Atomic VC Allocation ( Parameter : Wait for tail credit)

3.Fabric size ( Parameter : k)

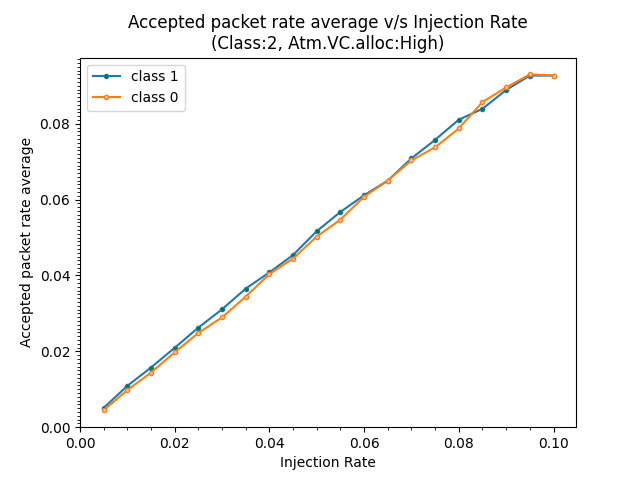
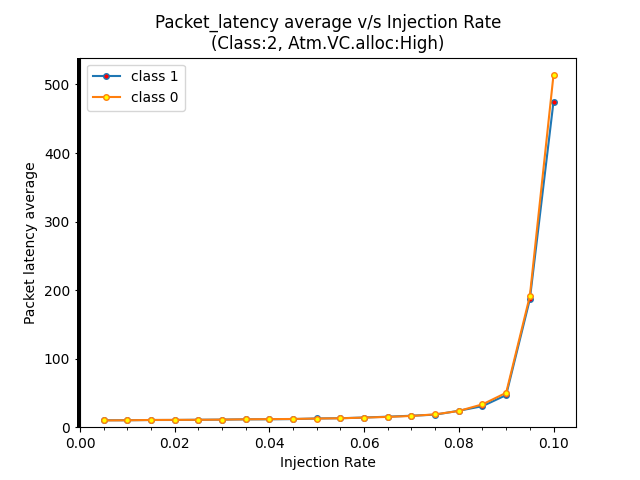
Note:

1. The Latency threshold has been set to 600 from 500 for better Saturation region visibility
2. Packet latency average and Accepted Packet Average which represent the throughput were plotted w.r.t a range of injection rates
3. A graph is plotted with the Packet Latency average w.r.t the Fabric size keeping the injection rate constant at 0.075(This is the highest injection rate value at which the simulation with all fabric sizes ranging from 2X2 to 5X5 run without an error)

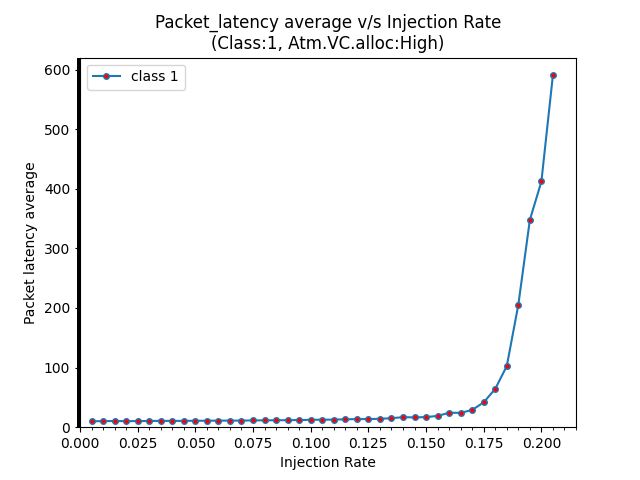
The plot was done to the default case of [Classes=2]and [Atomic VC allocation =

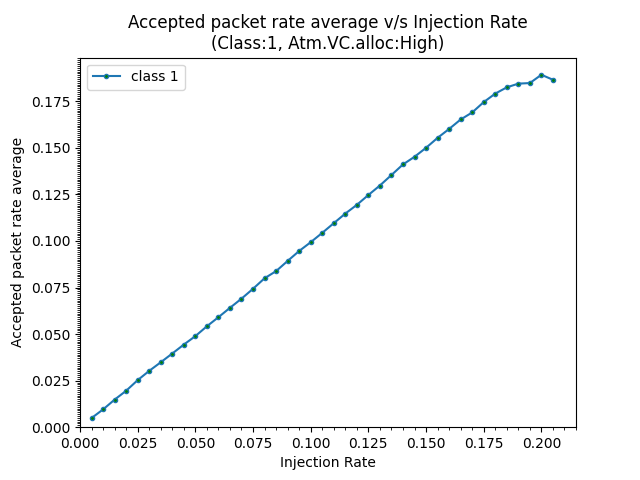
High]

1. All the graphs are plotted till the maximum injection rate possible(after which simulation fails or aborts)

**I. Classes = 2 , Atomic VC Allocation = High(Worst Case in terms of Latency)**

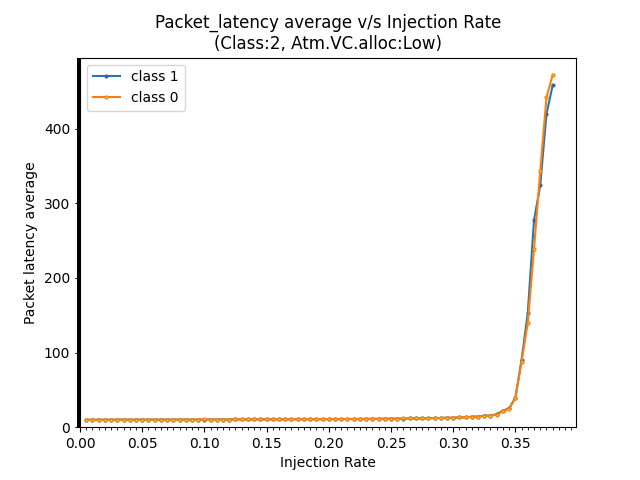
**II. Classes = 1, Atomic VC allocation = High**

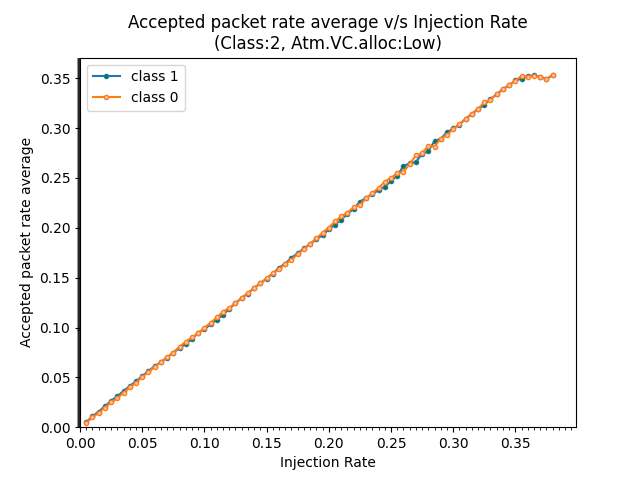
**(We could plot more injection rates upto 0.2 because of decreased latency and increased saturation throughput due to decreased classes)**



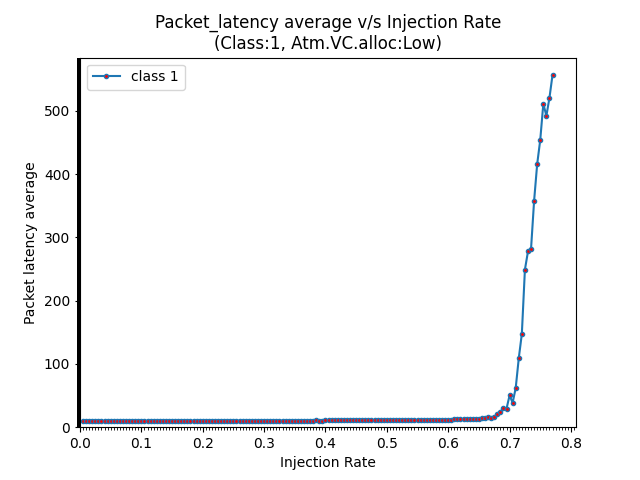
**III. Classes = 2 , Atomic VC allocation = Low**

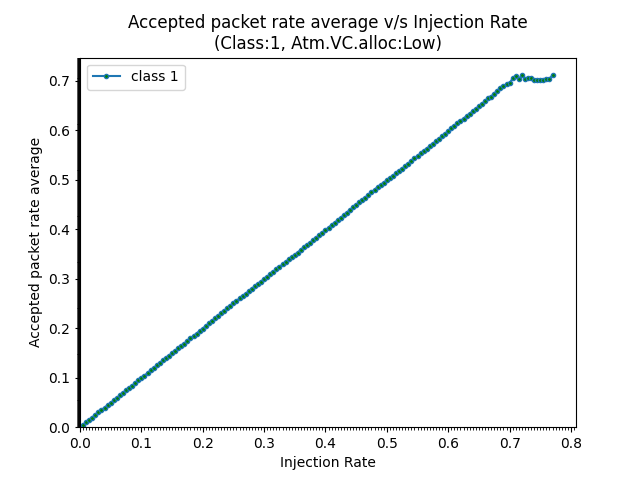
**(Slightly better than First case, could plot upto 0.35 , Also it shows that setting the “Atomic VC allocation” as low has increased the injection rate range better than reducing the number of classes)**

****

****

**IV. Classes = 1 , Atomic VC allocation = Low(Best Case, could plot till 0.8)**

****

****

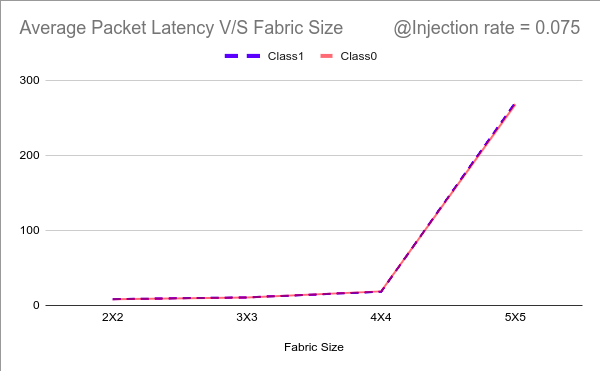
**Observation**

It can be observed that Latency has increased by increasing the no. of classes.(Classes may correspond to lanes flit in REDEFINE NoC) that is because the data from different lanes have to ultimately travel through a single path to reach the next router node. This requires arbitration to accommodate the flow of data from all the lanes , which takes time.

**Average packet latency vs Fabric size**

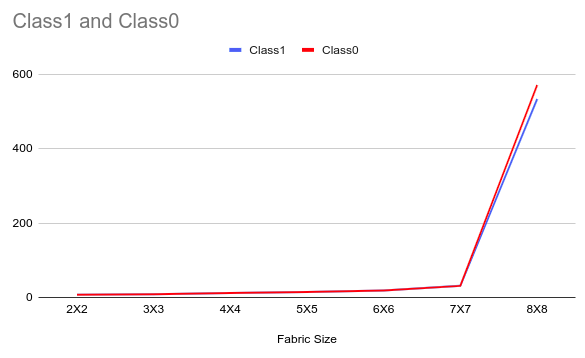
**I.** Average Packet Latency V/S Fabric Size @Injection rate = 0.075

**CLASS = 2**



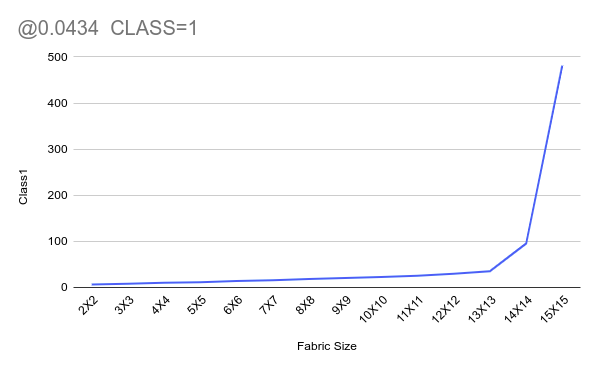
**II.** Average Packet Latency V/S Fabric Size @Injection rate = 0.0434

**CLASS = 2**



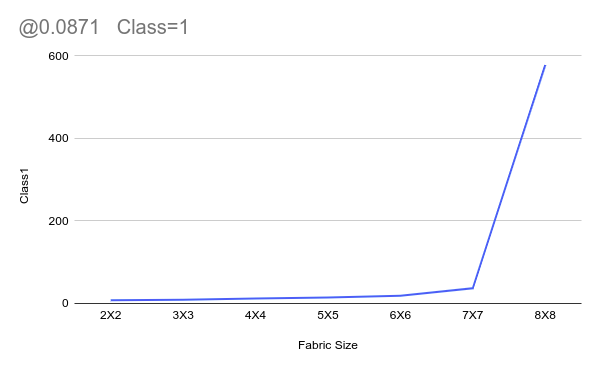
**III.** Average Packet Latency V/S Fabric Size @Injection rate = 0.0434

**CLASS = 1**



**IV.** Average Packet Latency V/S Fabric Size @Injection rate = 0.0871

**CLASS = 1**



**Observation**

1. It can be observed that the knee(Sharp rise in Average Packet Latency) has been shifted towards higher fabric size , with the decrease of injection rate.

@injection rate 0.075 , the knee is found at 4X4

@injection rate 0.0434 , the knee has moved to 7X7

1. It should be noted that the chosen injection rate is the maximum rate which can be set to achieve an errorless simulation for the target fabric size(which was 5X5 in case I and 8X8 in case II)
2. At the same injection rate 0.0434, we have set the number of classes to 1, which further extends the knee to fabric size 14X14 with 15X15 being the limit for errorless simulation.
3. With the target fabric size of 8X8 and keeping the number of classes to 1, we have found the maximum injection rate to be 0.0871

**Future Work**

1. Comparison with RTL Simulation
2. Checking if BookSim can be modified for Error Correction and Detection codes