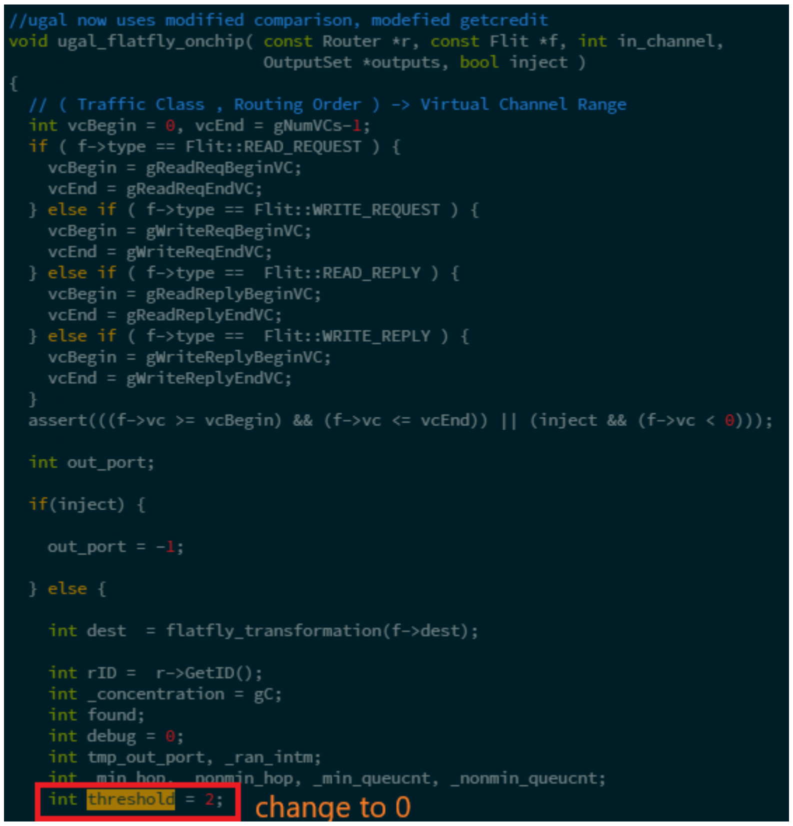
Task 0: Traffic and Routing Basics

For this task, we will explore the performance of routing algorithms on different traffics.

For the basics of routing, please refer to chapter 9 & 10 of Dally's textbook. Please be familiar with the difference between oblivious (minimal, non-minimal / Valiant) and adaptive routing.

For the basic of traffic, please refer to chapter 3.

Before starting, in the networks/flatfly_onchip.cpp code, please set the threshold value to 0, as shown below.



```
//ugal now uses modified comparison, modefied getcredit
void ugal_flatfly_onchip( const Router *r, const Flit *f, int in_channel,
                          OutputSet *outputs, bool inject )
{
  // ( Traffic Class , Routing Order ) -> Virtual Channel Range
  int vcBegin = 0, vcEnd = gNumVCs-1;
  if ( f->type == Flit::READ_REQUEST ) {
    vcBegin = gReadReqBeginVC;
    vcEnd = gReadReqEndVC;
  } else if ( f->type == Flit::WRITE_REQUEST ) {
    vcBegin = gWriteReqBeginVC;
    vcEnd = gWriteReqEndVC;
  } else if ( f->type ==  Flit::READ_REPLY ) {
    vcBegin = gReadReplyBeginVC;
    vcEnd = gReadReplyEndVC;
  } else if ( f->type ==  Flit::WRITE_REPLY ) {
    vcBegin = gWriteReplyBeginVC;
    vcEnd = gWriteReplyEndVC;
  }
  assert(((f->vc >= vcBegin) && (f->vc <= vcEnd)) || (inject && (f->vc < 0)));

  int out_port;

  if(inject) {

    out_port = -1;

  } else {

    int dest  = flatfly_transformation(f->dest);

    int rID =  r->GetID();
    int _concentration = gC;
    int found;
    int debug = 0;
    int tmp_out_port, _ran_intm;
    int _min_hop, _nonmin_hop, _min_queucnt, _nonmin_queucnt;
    int threshold = 2;
```

Don't forget to run make.

For all the simulations in this tutorial, we will use 1 cycle channel latency, 2 VCs and 16-flit buffer depth / VC.

Questions

1. Minimal routing is the ideal routing for uniform random traffic. Why?
2. Non-minimal / Valiant routing is the ideal routing for adversarial traffic. Why?
3. What is the ideal throughput for uniform random traffic and adversarial traffic?
4. In non-minimal routing, the intermediate node is chosen randomly. Why?
5. What is the adversarial traffic for a 1D Flattened Butterfly topology? Define as an equation as shown in Table 3.1 in Dally's textbook. *(Hint: an adversarial traffic is a traffic that causes the largest bottleneck, hence reducing the ideal throughput. Refer to Section 3.3 of Dally's textbook)*.

Task 1: Adding Adversarial Traffic

Since the adversarial traffic is not provided by BookSim, we need to define the traffic by ourselves. To do so, we need to modify `traffic.cpp` and `traffic.hpp`.

Instead of providing step-by-step solution, please look into how the adversarial traffic in Dragonfly is implemented in those 2 files mentioned above. The traffic is defined as class `BadPermDFlyTrafficPattern`. Please understand how the code works and apply it to define the adversarial traffic for 1D Flattened Butterfly. The only changes required should only be how to define the `dest` function, since the adversarial traffic for Dragonfly and 1D Flattened Butterfly is slightly different (in Dragonfly, the adversarial traffic congests the inter-group / global channel but in 1D Flattened Butterfly, the congestion occurs at inter-router channel).

Questions

1. Simulate a 16-router, 256-nodes 1D Flattened Butterfly node (please adjust the value of `k`, `c`, `n`, `x`, `y`, `xr`, and `yr` accordingly). Run minimal routing on adversarial traffic. What is the saturation throughput (*Hint: plot the latency-throughput graph)*? Is it far from the ideal throughput?
2. Similar with question 1, we will run minimal routing on adversarial traffic but on a 4-router, 16-nodes 1D Flattened Butterfly. Is the saturation throughput higher/lower than the configuration in Question 1? Why?
3. Similar with question 1 and 2 but try with a 32-router, 1024-node 1D Flattened Butterfly. How does the saturation throughput compare (higher/lower) with question 1 and 2? Why?

Task 2: Adaptive Routing

For this task, we will try to run a widely-used adaptive routing called UGAL. To change the routing function, please change the routing function as follows.

```
routing_function = ugal;
```

Questions

1. How does UGAL work? How does it decide when to route minimally or non-minimally?
2. Run the simulation on a 16-router, 256-nodes 1D Flattened Butterfly. What is the saturation throughput achieved by UGAL for minimal and adversarial traffic? Does it approach the ideal saturation throughput for both traffics?
3. Do you think the performance of UGAL will be affected if it is run on different network sizes (16-nodes and 1024-nodes 1D Flattened Butterfly).